Panel data econometrics in R:

the plm package

Yves Croissant

Giovanni Millo

Abstract

This introduction to the `plm` package is a modified and extended version of Croissant and Millo (2008), published in the *Journal of Statistical Software*.

Panel data econometrics is obviously one of the main fields in the statistics profession, but most of the models used are difficult to estimate with only plain `R`. `plm` is a package for `R` which intends to make the estimation of linear panel models straightforward. `plm` provides functions to estimate a wide variety of models and to make (robust) inference.

Introduction

Panel data econometrics is a continuously developing field. The increasing availability of data observed on cross-sections of units (like households, firms, countries etc.) *and* over time has given rise to a number of estimation approaches exploiting this double dimensionality to cope with some of the typical problems associated with economic data, first of all that of unobserved heterogeneity.

Timewise observation of data from different observational units has long been common in other fields of statistics (where they are often termed *longitudinal* data). In the panel data field as well as in others, the econometric approach is nevertheless peculiar with respect to experimental contexts, as it is emphasizing model specification and testing and tackling a number of issues arising from the particular statistical problems associated with economic data.

Thus, while a very comprehensive software framework for (among many other features) maximum likelihood estimation of linear regression models for longitudinal data, packages `nlme` (J. Pinheiro et al. 2007) and `lme4` (Bates 2007), is available in the `R` (Development Core Team (2008)) environment and can be used, e.g., for estimation of random effects panel models, its use is not intuitive for a practicing econometrician, and maximum likelihood estimation is only one of the possible approaches to panel data econometrics. Moreover, economic panel data sets often happen to be *unbalanced* (i.e., they have a different number of observations between groups), which case needs some adaptation to the methods and is not compatible with those in `nlme`. Hence the need for a package doing panel data "from the econometrician's viewpoint" and featuring at a minimum the basic techniques econometricians are used to: random and fixed effects estimation of static linear panel data models, variable coefficients models, generalized method of moments estimation of dynamic models; and the basic toolbox of specification and misspecification diagnostics.

Furthermore, we felt there was a need for automation of some basic data management tasks such as lagging, summing and, more in general, `apply`ing (in the `R` sense) functions to the data, which, although conceptually simple, become cumbersome and error-prone on two-dimensional data, especially in the case of unbalanced panels.

This paper is organized as follows: Section linear panel model presents a very short overview of the typical model taxonomy[1]. Section software approach discusses the software approach used in the package. The next three sections present the functionalities of the package in more detail: data management (Section managing data and formulae), estimation (Section model estimation) and testing (Section tests), giving a short description and illustrating them with examples. Section plm vs nlme and lme4 compares the approach in `plm` to that of `nlme` and `lme4`, highlighting the features of the latter two that an econometrician might find most useful. Section conclusion concludes the paper.

The linear panel model

The basic linear panel models used in econometrics can be described through suitable restrictions of the following general model:

$$y_{it} = \alpha_{it} + \beta_{it}^{\top} x_{it} + u_{it}$$

where $i = 1, \ldots, n$ is the individual (group, country ...) index, $t = 1, \ldots, T$ is the time index and u_{it} a random disturbance term of mean 0.

Of course u_{it} is not estimable with $N = n \times T$ data points. A number of assumptions are usually made about the parameters, the errors and the exogeneity of the regressors, giving rise to a taxonomy of feasible models for panel data.

The most common one is parameter homogeneity, which means that $\alpha_{it} = \alpha$ for all i, t and $\beta_{it} = \beta$ for all i, t. The resulting model

$$y_{it} = \alpha + \beta^{\top} x_{it} + u_{it}$$

is a standard linear model pooling all the data across i and t.

To model individual heterogeneity, one often assumes that the error term has two separate components, one of which is specific to the individual and doesn't change over time[2]. This is called the unobserved effects model:

$$(\#eq : errcomp) y_{it} = \alpha + \beta^{\top} x_{it} + \mu_i + \epsilon_{it}$$

The appropriate estimation method for this model depends on the properties of the two error components. The idiosyncratic error ϵ_{it} is usually assumed well-behaved and independent of both the regressors x_{it} and the individual error component μ_i. The individual component may be in turn either independent of the regressors or correlated.

If it is correlated, the ordinary least squares (OLS) estimator of β would be inconsistent, so it is customary to treat the μ_i as a further set of n parameters to be estimated, as if in the general model $\alpha_{it} = \alpha_i$ for all t. This is called the fixed effects (a.k.a. *within* or *least squares dummy variables*) model, usually estimated by OLS on transformed data, and gives consistent estimates for β.

If the individual-specific component μ_i is uncorrelated with the regressors, a situation which is usually termed *random effects*, the overall error u_{it} also is, so the OLS estimator is consistent. Nevertheless, the common error component over individuals induces correlation across the composite error terms, making OLS estimation inefficient, so one has to resort to some form of feasible generalized least squares (GLS) estimators. This is based on the estimation of the variance of the two error components, for which there are a number of different procedures available.

If the individual component is missing altogether, pooled OLS is the most efficient estimator for β. This set of assumptions is usually labelled *pooling* model, although this actually refers to the errors'

properties and the appropriate estimation method rather than the model itself. If one relaxes the usual hypotheses of well-behaved, white noise errors and allows for the idiosyncratic error ϵ_{it} to be arbitrarily heteroskedastic and serially correlated over time, a more general kind of feasible GLS is needed, called the *unrestricted* or *general* GLS. This specification can also be augmented with individual-specific error components possibly correlated with the regressors, in which case it is termed *fixed effects* GLS.

Another way of estimating unobserved effects models through removing time-invariant individual components is by first-differencing the data: lagging the model and subtracting, the time-invariant components (the intercept and the individual error component) are eliminated, and the model

$$\Delta y_{it} = \beta^\top \Delta x_{it} + \Delta u_{it}$$

(where $\Delta y_{it} = y_{it} - y_{i,t-1}$, $\Delta x_{it} = x_{it} - x_{i,t-1}$ and, from @ref(eq:errcomp), $\Delta u_{it} = u_{it} - u_{i,t-1} = \Delta \epsilon_{it}$ for $t = 2, \ldots, T$) can be consistently estimated by pooled OLS. This is called the *first-difference* or FD estimator. Its relative efficiency, and so reasons for choosing it against other consistent alternatives, depends on the properties of the error term. The FD estimator is usually preferred if the errors u_{it} are strongly persistent in time, because then the Δu_{it} will tend to be serially uncorrelated.

Lastly, the *between* model, which is computed on time (group) averages of the data, discards all the information due to intragroup variability but is consistent in some settings (e.g., non-stationarity) where the others are not, and is often preferred to estimate long-run relationships.

Variable coefficients models relax the assumption that $\beta_{it} = \beta$ for all i, t. Fixed coefficients models allow the coefficients to vary along one dimension, like $\beta_{it} = \beta_i$ for all t. Random coefficients models instead assume that coefficients vary randomly around a common average, as $\beta_{it} = \beta + \eta_i$ for all t, where η_i is a group– (time–) specific effect with mean zero.

The hypotheses on parameters and error terms (and hence the choice of the most appropriate estimator) are usually tested by means of:

- *pooling* tests to check poolability, i.e., the hypothesis that the same coefficients apply across all individuals,
- if the homogeneity assumption over the coefficients is established, the next step is to establish the presence of unobserved effects, comparing the null of spherical residuals with the alternative of group (time) specific effects in the error term,
- the choice between fixed and random effects specifications is based on Hausman-type tests, comparing the two estimators under the null of no significant difference: if this is not rejected, the more efficient random effects estimator is chosen,
- even after this step, departures of the error structure from sphericity can further affect inference, so that either screening tests or robust diagnostics are needed.

Dynamic models and in general lack of strict exogeneity of the regressors, pose further problems to estimation which are usually dealt with in the generalized method of moments (GMM) framework.

These were, in our opinion, the basic requirements of a panel data econometrics package for the `R` language and environment. Some, as often happens with `R`, were already fulfilled by packages developed for other branches of computational statistics, while others (like the fixed effects or the between estimators) were straightforward to compute after transforming the data, but in every case there were either language inconsistencies w.r.t. the standard econometric toolbox or subtleties to be dealt with (like, for example, appropriate computation of standard errors for the demeaned model, a common pitfall), so we felt there was need for an “all in one” econometrics-oriented package allowing to make specification searches, estimation and inference in a natural way.

Software approach

Data structure

Panel data have a special structure: each row of the data corresponds to a specific individual and time period. In `plm` the `data` argument may be an ordinary `data.frame` but, in this case, an argument called `index` has to be added to indicate the structure of the data. This can be:

- `NULL` (the default value), it is then assumed that the first two columns contain the individual and the time index and that observations are ordered by individual and by time period,
- a character string, which should be the name of the individual index,
- a character vector of length two containing the names of the individual and the time index,
- an integer which is the number of individuals (only in case of a balanced panel with observations ordered by individual).

The `pdata.frame` function is then called internally, which returns a `pdata.frame` which is a `data.frame` with an attribute called index. This attribute is a `data.frame` that contains the individual and the time indexes.

It is also possible to use directly the `pdata.frame` function and then to use the `pdata.frame` in the estimation functions.

Interface

Estimation interface

Package `plm` provides various functions for panel data estimation, among them:

- `plm`: estimation of the basic panel models and instrumental variable panel models, *i.e.*, between and first-difference models and within and random effect models. Models are estimated internally using the `lm` function on transformed data,
- `pvcm`: estimation of models with variable coefficients,
- `pgmm`: estimation of generalized method of moments models,
- `pggls`: estimation of general feasible generalized least squares models,
- `pmg`: estimators for mean groups (MG), demeaned MG (DMG) and common correlated effects MG (CCEMG) for heterogeneous panel models,
- `pcce`: estimators for common correlated effects mean groups (CCEMG) and pooled (CCEP) for panel data with common factors,
- `pldv`: panel estimators for limited dependent variables.

The interface of these functions is consistent with the `lm()` function. Namely, their first two arguments are `formula` and `data` (which should be a `data.frame` and is mandatory). Three additional arguments are common to these functions:

- `index`: this argument enables the estimation functions to identify the structure of the data, *i.e.*, the individual and the time period for each observation,
- `effect`: the kind of effects to include in the model, *i.e.*, individual effects, time effects or both[3],
- `model`: the kind of model to be estimated, most of the time a model with fixed effects or a model

with random effects.

The results of these four functions are stored in an object which class has the same name of the function. They all inherit from class `panelmodel`. A `panelmodel` object contains: `coefficients`, `residuals`, `fitted.values`, `vcov`, `df.residual` and `call` and functions that extract these elements are provided.

Testing interface

The diagnostic testing interface provides both `formula` and `panelmodel` methods for most functions, with some exceptions. The user may thus choose whether to employ results stored in a previously estimated `panelmodel` object or to re-estimate it for the sake of testing.

Although the first strategy is the most efficient one, diagnostic testing on panel models mostly employs OLS residuals from pooling model objects, whose estimation is computationally inexpensive. Therefore most examples in the following are based on `formula` methods, which are perhaps the cleanest for illustrative purposes.

Computational approach to estimation

The feasible GLS methods needed for efficient estimation of unobserved effects models have a simple closed-form solution: once the variance components have been estimated and hence the covariance matrix of errors $\hat{V}$, model parameters can be estimated as

$$(\#eq : naive)\hat{\beta} = (X^\top \hat{V}^{-1} X)^{-1}(X^\top \hat{V}^{-1} y)$$

Nevertheless, in practice plain computation of $\hat{\beta}$ has long been an intractable problem even for moderate-sized data sets because of the need to invert the $N \times N$ $\hat{V}$ matrix. With the advances in computer power, this is no more so, and it is possible to program the "naive" estimator @ref(eq:naive) in `R` with standard matrix algebra operators and have it working seamlessly for the standard "guinea pigs", e.g., the Grunfeld data. Estimation with a couple of thousands of data points also becomes feasible on a modern machine, although excruciatingly slow and definitely not suitable for everyday econometric practice. Memory limits would also be very near because of the storage needs related to the huge $\hat{V}$ matrix. An established solution exists for the random effects model which reduces the problem to an ordinary least squares computation.

The (quasi–)demeaning framework

The estimation methods for the basic models in panel data econometrics, the pooled OLS, random effects and fixed effects (or within) models, can all be described inside the OLS estimation framework. In fact, while pooled OLS simply pools data, the standard way of estimating fixed effects models with, say, group (time) effects entails transforming the data by subtracting the average over time (group) to every variable, which is usually termed *time-demeaning*. In the random effects case, the various feasible GLS estimators which have been put forth to tackle the issue of serial correlation induced by the group-invariant random effect have been proven to be equivalent (as far as estimation of βs is concerned) to OLS on *partially demeaned* data, where partial demeaning is defined as:

$$(\#eq : ldemmodel)y_{it} - \theta\bar{y}_i = (X_{it} - \theta\bar{X}_i)\beta + (u_{it} - \theta\bar{u}_i)$$

where $\theta = 1 - [\sigma_u^2/(\sigma_u^2 + T\sigma_e^2)]^{1/2}$, $\bar{y}$ and $\bar{X}$ denote time means of y and X, and the disturbance $v_{it} - \theta\bar{v}_i$ is homoskedastic and serially uncorrelated. Thus the feasible RE estimate for β may be obtained estimating $\hat{\theta}$ and running an OLS regression on the transformed data with `lm()`. The other estimators can be computed as special cases: for $\theta = 1$ one gets the fixed effects estimator, for $\theta = 0$ the pooled OLS one.

Moreover, instrumental variable estimators of all these models may also be obtained using several calls to `lm()`.

For this reason the three above estimators have been grouped inside the same function.

On the output side, a number of diagnostics and a very general coefficients' covariance matrix estimator also benefits from this framework, as they can be readily calculated applying the standard OLS formulas to the demeaned data, which are contained inside `plm` objects. This will be the subject of subsection inference in the panel model.

The object-oriented approach to general GLS computations

The covariance matrix of errors in general GLS models is too generic to fit the quasi-demeaning framework, so this method calls for a full-blown application of GLS as in @ref(eq:naive). On the other hand, this estimator relies heavily on n–asymptotics, making it theoretically most suitable for situations which forbid it computationally: e.g., "short" micropanels with thousands of individuals observed over few time periods.

`R` has general facilities for fast matrix computation based on object orientation: particular types of matrices (symmetric, sparse, dense etc.) are assigned the relevant class and the additional information on structure is used in the computations, sometimes with dramatic effects on performance (see Bates (2004)) and packages `Matrix` (see Bates and Maechler (2016)) and `SparseM` (see Koenker and Ng (2016)). Some optimized linear algebra routines are available in the `R` package `bdsmatrix` (see Therneau (2014)) which exploit the particular block-diagonal and symmetric structure of $\hat{V}$ making it possible to implement a fast and reliable full-matrix solution to problems of any practically relevant size.

The $\hat{V}$ matrix is constructed as an object of class `bdsmatrix`. The peculiar properties of this matrix class are used for efficiently storing the object in memory and then by ad-hoc versions of the `solve` and `crossprod` methods, dramatically reducing computing times and memory usage. The resulting matrix is then used "the naive way" as in @ref(eq:naive) to compute $\hat{\beta}$, resulting in speed comparable to that of the demeaning solution.

Inference in the panel model

General frameworks for restrictions and linear hypotheses testing are available in the `R` environment[4]. These are based on the Wald test, constructed as $\hat{\beta}^\top \hat{V}^{-1} \hat{\beta}$, where $\hat{\beta}$ and $\hat{V}$ are consistent estimates of β and $V(\beta)$, The Wald test may be used for zero-restriction (i.e., significance) testing and, more generally, for linear hypotheses in the form $(R\hat{\beta} - r)^\top [R\hat{V}R^\top]^{-1}(R\hat{\beta} - r)$[5]. To be applicable, the test functions require extractor methods for coefficients' and covariance matrix estimates to be defined for the model object to be tested. Model objects in `plm` all have `coef()` and `vcov()` methods and are therefore compatible with the above functions.

In the same framework, robust inference is accomplished substituting ("plugging in") a robust estimate

of the coefficient covariance matrix into the Wald statistic formula. In the panel context, the estimator of choice is the White system estimator. This called for a flexible method for computing robust coefficient covariance matrices *à la White* for `plm` objects.

A general White system estimator for panel data is:

$$\hat{V}_R(\beta) = (X^\top X)^{-1} \sum_{i=1}^{n} X_i^\top E_i X_i (X^\top X)^{-1}$$

where E_i is a function of the residuals $\hat{e}_{it}$, $t = 1, \dots T$ chosen according to the relevant heteroskedasticity and correlation structure. Moreover, it turns out that the White covariance matrix calculated on the demeaned model's regressors and residuals (both part of `plm` objects) is a consistent estimator of the relevant model's parameters' covariance matrix, thus the method is readily applicable to models estimated by random or fixed effects, first difference or pooled OLS methods. Different pre-weighting schemes taken from package `sandwich` (see Zeileis (2004); Lumley and Zeileis (2015)) are also implemented to improve small-sample performance. Robust estimators with any combination of covariance structures and weighting schemes can be passed on to the testing functions.

Managing data and formulae

The package is now illustrated by application to some well-known examples. It is loaded using

```
library("plm")
```

The four data sets used are `EmplUK` which was used by M. Arellano and Bond (1991), the `Grunfeld` data (Kleiber and Zeileis 2008) which is used in several econometric books, the `Produc` data used by Munnell (1990) and the `Wages` used by Cornwell and Rupert (1988).

```
data("EmplUK", package="plm")
data("Produc", package="plm")
data("Grunfeld", package="plm")
data("Wages", package="plm")
```

Data structure

As observed above, the current version of `plm` is capable of working with a regular `data.frame` without any further transformation, provided that the individual and time indexes are in the first two columns, as in all the example data sets but `Wages`. If this weren't the case, an `index` optional argument would have to be passed on to the estimating and testing functions.

```
head(Grunfeld)
```

```
##   firm year   inv  value capital
## 1    1 1935 317.6 3078.5     2.8
## 2    1 1936 391.8 4661.7    52.6
## 3    1 1937 410.6 5387.1   156.9
```

```
## 4    1 1938 257.7 2792.2   209.2
## 5    1 1939 330.8 4313.2   203.4
## 6    1 1940 461.2 4643.9   207.2
```

```
E <- pdata.frame(EmplUK, index=c("firm","year"), drop.index=TRUE, row.names=TRUE)
head(E)
```

```
##        sector   emp    wage capital   output
## 1-1977      7 5.041 13.1516  0.5894  95.7072
## 1-1978      7 5.600 12.3018  0.6318  97.3569
## 1-1979      7 5.015 12.8395  0.6771  99.6083
## 1-1980      7 4.715 13.8039  0.6171 100.5501
## 1-1981      7 4.093 14.2897  0.5076  99.5581
## 1-1982      7 3.166 14.8681  0.4229  98.6151
```

```
head(attr(E, "index"))
```

```
##   firm year
## 1    1 1977
## 2    1 1978
## 3    1 1979
## 4    1 1980
## 5    1 1981
## 6    1 1982
```

Two further arguments are logical: `drop.index = TRUE` drops the indexes from the `data.frame` and `row.names = TRUE` computes "fancy" row names by pasting the individual and the time indexes. While extracting a series from a `pdata.frame`, a `pseries` is created, which is the original series with the index attribute. This object has specific methods, like `summary` and `as.matrix`. The former indicates the total variation of the variable and the shares of this variation due to the individual and the time dimensions. The latter gives the matrix representation of the series, with, by default, individuals as rows and times as columns.

```
summary(E$emp)
```

```
## total sum of squares: 261539.4
##          id        time
## 0.980765381 0.009108488
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.104   1.181   2.287   7.892   7.020 108.562
```

```
head(as.matrix(E$emp))
```

```
##     1976   1977   1978   1979   1980   1981   1982   1983 1984
## 1     NA  5.041  5.600  5.015  4.715  4.093  3.166  2.936   NA
## 2     NA 71.319 70.643 70.918 72.031 73.689 72.419 68.518   NA
## 3     NA 19.156 19.440 19.900 20.240 19.570 18.125 16.850   NA
## 4     NA 26.160 26.740 27.280 27.830 27.169 24.504 22.562   NA
## 5 86.677 87.100 87.000 90.400 89.200 82.700 73.700     NA   NA
## 6  0.748  0.766  0.762  0.729  0.731  0.779  0.782     NA   NA
```

Data transformation

Panel data estimation requires to apply different transformations to raw series. If x is a series of length nT (where n is the number of individuals and T is the number of time periods), the transformed series $\tilde{x}$ is obtained as $\tilde{x} = Mx$ where M is a transformation matrix. Denoting j a vector of one of length T and I_n the identity matrix of dimension n, we get:

- the between transformation: $P = \frac{1}{T} I_n \otimes jj'$ returns a vector containing the individual means. The `Between` and `between` functions perform this operation, the first one returning a vector of length nT, the second one a vector of length n,
- the within transformation: $Q = I_{nT} - P$ returns a vector containing the values in deviation from the individual means. The `Within` function performs this operation.
- the first difference transformation $D = I_n \otimes d$ where

$$d = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix}$$

is of dimension $(T-1, T)$.

Note that R's `diff()` and `lag()` functions don't compute correctly these transformations for panel data because they are unable to identify when there is a change in individual in the data. Therefore, specific methods for `pseries` objects have been written in order to handle correctly panel data. Note that compared to the `lag()` method for `ts` objects, the order of lags are indicated by a positive integer. Moreover, 0 is a relevant value and a vector argument may be provided:

```
head(lag(E$emp, 0:2))
```

```
##            0     1     2
## 1-1977 5.041    NA    NA
## 1-1978 5.600 5.041    NA
## 1-1979 5.015 5.600 5.041
## 1-1980 4.715 5.015 5.600
## 1-1981 4.093 4.715 5.015
## 1-1982 3.166 4.093 4.715
```

Further functions called `Between`, `between` and `Within` are also provided to compute the between and

the within transformation. The `between` returns unique values, whereas `Between` duplicates the values and returns a vector which length is the number of observations.

```
head(diff(E$emp), 10)
```

```
##     1-1977     1-1978     1-1979     1-1980     1-1981     1-1982     1-1983
##         NA  0.5590000 -0.5850000 -0.2999997 -0.6220003 -0.9270000 -0.2299998
##     2-1977     2-1978     2-1979
##         NA -0.6760020  0.2750010
```

```
head(lag(E$emp, 2), 10)
```

```
## 1-1977 1-1978 1-1979 1-1980 1-1981 1-1982 1-1983 2-1977 2-1978 2-1979
##     NA     NA  5.041  5.600  5.015  4.715  4.093     NA     NA 71.319
```

```
head(Within(E$emp))
```

```
##     1-1977     1-1978     1-1979     1-1980     1-1981     1-1982
##  0.6744285  1.2334285  0.6484285  0.3484288 -0.2735715 -1.2005715
```

```
head(between(E$emp), 4)
```

```
##         1         2         3         4
##  4.366571 71.362428 19.040143 26.035000
```

```
head(Between(E$emp), 10)
```

```
##         1         1         1         1         1         1         1         2
##  4.366571  4.366571  4.366571  4.366571  4.366571  4.366571  4.366571 71.362428
##         2         2
## 71.362428 71.362428
```

Formulas

In some circumstances, standard `formula`s are not very useful to describe a model, notably while using instrumental variable like estimators: to deal with these situations, we use the `Formula` package.

The `Formula` package provides a class which enables to construct multi-part formula, each part being separated by a pipe sign (`|`).

The two formulas below are identical:

```
emp ~ wage + capital | lag(wage, 1) + capital
emp ~ wage + capital | . -wage + lag(wage, 1)
```

In the second case, the `.` means the previous parts which describes the covariates and this part is "updated". This is particularly interesting when there are a few external instruments.

Model estimation

Estimation of the basic models with plm

Several models can be estimated with `plm` by filling the `model` argument:

- the fixed effects model (`"within"`), the default,
- the pooling model (`"pooling"`),
- the first-difference model (`"fd"`),
- the between model (`"between"`),
- the error components model (`"random"`).

The basic use of `plm` is to indicate the model formula, the data and the model to be estimated. For example, the fixed effects model and the random effects model are estimated using:

```
grun.fe <- plm(inv~value+capital, data = Grunfeld, model = "within")
grun.re <- plm(inv~value+capital, data = Grunfeld, model = "random")
```

Methods to display a summary of the model estimation are available via `summary`. For example, for a `random` model, the `summary` method gives information about the variance of the components of the errors and some test statistics. Random effects of the estimated model can be extracted via `ranef`.

```
summary(grun.re)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = inv ~ value + capital, data = Grunfeld, model = "random")
##
## Balanced Panel: n = 10, T = 20, N = 200
##
## Effects:
##                   var std.dev share
## idiosyncratic 2784.46   52.77 0.282
## individual    7089.80   84.20 0.718
## theta: 0.8612
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -177.6063  -19.7350    4.6851   19.5105  252.8743
```

```
##
## Coefficients:
##               Estimate Std. Error z-value             Pr(>|z|)
## (Intercept) -57.834415  28.898935 -2.0013              0.04536 *
## value         0.109781   0.010493 10.4627 < 0.0000000000000002 ***
## capital       0.308113   0.017180 17.9339 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    2381400
## Residual Sum of Squares: 548900
## R-Squared:      0.7695
## Adj. R-Squared: 0.76716
## Chisq: 657.674 on 2 DF, p-value: < 0.000000000000000222
```

```
ranef(grun.re)
```

```
##           1           2           3           4           5           6
##  -9.5242955 157.8910235 -172.8958044   29.9119801  -54.6790089   34.3461316
##           7           8           9          10
##  -7.8977584   0.6726376  -28.1393497   50.3144442
```

The fixed effects of a fixed effects model may be extracted easily using `fixef`. An argument `type` indicates how fixed effects should be computed: in levels by `type = "level"` (the default), in deviations from the overall mean by `type = "dmean"` or in deviations from the first individual by `type = "dfirst"`.

```
fixef(grun.fe, type = "dmean")
```

```
##         1         2         3         4         5         6         7         8
##  -11.5528  160.6498 -176.8279   30.9346  -55.8729   35.5826   -7.8095    1.1983
##         9        10
##  -28.4783   52.1761
```

The `fixef` function returns an object of class `fixef`. A summary method is provided, which prints the effects (in deviation from the overall intercept), their standard errors and the test of equality to the overall intercept.

```
summary(fixef(grun.fe, type = "dmean"))
```

```
##     Estimate Std. Error t-value        Pr(>|t|)
## 1   -11.5528    49.7080 -0.2324       0.8164700
## 2   160.6498    24.9383  6.4419 0.0000000009627 ***
## 3  -176.8279    24.4316 -7.2377 0.0000000000113 ***
## 4    30.9346    14.0778  2.1974       0.0292129 *
## 5   -55.8729    14.1654 -3.9443       0.0001129 ***
```

```
## 6    35.5826    12.6687  2.8087       0.0054998 **
## 7    -7.8095    12.8430 -0.6081       0.5438694
## 8     1.1983    13.9931  0.0856       0.9318489
## 9   -28.4783    12.8919 -2.2090       0.0283821 *
## 10   52.1761    11.8269  4.4116 0.0000172511647 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In case of a two-ways fixed effect model, argument `effect` is relevant in function `fixef` to extract specific effect fixed effects with possible values `"individual"` for individual fixed effects (default for two-ways fixed effect models), `"time"` for time fixed effects, and `"twoways"` for the sum of individual and time fixed effects. Example to extract the time fixed effects from a two-ways model:

```
grun.twfe <- plm(inv~value+capital, data=Grunfeld, model="within", effect="twoways")
fixef(grun.twfe, effect = "time")
```

```
##    1935    1936    1937    1938    1939    1940    1941    1942    1943    1944
##  -86.90 -106.10 -127.59 -126.13 -156.37 -131.14 -105.70 -108.04 -129.88 -130.00
##    1945    1946    1947    1948    1949    1950    1951    1952    1953    1954
## -142.58 -118.07 -126.29 -130.62 -160.40 -162.80 -149.38 -151.53 -154.62 -180.43
```

More advanced use of plm

Random effects estimators

As observed above, the random effect model is obtained as a linear estimation on quasi-demeaned data. The parameter of this transformation is obtained using preliminary estimations.

Four estimators of this parameter are available, depending on the value of the argument `random.method`:

- `"swar"`: from Swamy and Arora (1972), the default value,
- `"walhus"`: from Wallace and Hussain (1969),
- `"amemiya"`: from T. Amemiya (1971),
- `"nerlove"`: from Nerlove (1971).
- `"ht"`: for Hausman-Taylor-type instrumental variable (IV) estimation, discussed later, see Section Instrumental variable estimator.

For example, to use the `amemiya` estimator:

```
grun.amem <- plm(inv~value+capital, data=Grunfeld,
                 model="random", random.method="amemiya")
```

The estimation of the variance of the error components are performed using the `ercomp` function, which has a `method` and an `effect` argument, and can be used by itself:

```
ercomp(inv~value+capital, data=Grunfeld, method = "amemiya", effect = "twoways")
```

```
##                   var std.dev share
## idiosyncratic 2644.13   51.42 0.256
## individual    7452.02   86.33 0.721
## time           243.78   15.61 0.024
## theta: 0.868 (id) 0.2787 (time) 0.2776 (total)
```

Introducing time or two-ways effects

The default behavior of `plm` is to introduce individual effects. Using the `effect` argument, one may also introduce:

- time effects (`effect = "time"`),
- individual and time effects (`effect = "twoways"`).

For example, to estimate a two-ways effect model for the `Grunfeld` data:

```
grun.tways <- plm(inv~value+capital, data = Grunfeld, effect = "twoways",
                  model = "random", random.method = "amemiya")
summary(grun.tways)
```

```
## Twoways effects Random Effect Model
##    (Amemiya's transformation)
##
## Call:
## plm(formula = inv ~ value + capital, data = Grunfeld, effect = "twoways",
##     model = "random", random.method = "amemiya")
##
## Balanced Panel: n = 10, T = 20, N = 200
##
## Effects:
##                   var std.dev share
## idiosyncratic 2644.13   51.42 0.256
## individual    7452.02   86.33 0.721
## time           243.78   15.61 0.024
## theta: 0.868 (id) 0.2787 (time) 0.2776 (total)
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -176.9062  -18.0431    3.2697   17.1719  234.1735
##
## Coefficients:
##               Estimate Std. Error z-value             Pr(>|z|)
## (Intercept) -63.767791  29.851537 -2.1362              0.03267 *
## value         0.111386   0.010909 10.2102 < 0.0000000000000002 ***
## capital       0.323321   0.018772 17.2232 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Total Sum of Squares:    2066800
## Residual Sum of Squares: 518200
## R-Squared:      0.74927
## Adj. R-Squared: 0.74673
## Chisq: 588.717 on 2 DF, p-value: < 0.000000000000000222
```

In the “effects” section of the printed summary of the result, the variance of the three elements of the error term and the three parameters used in the transformation are printed.

Unbalanced panels

Estimations by `plm` support unbalanced panel models.

The following example is using data used by Harrison and Rubinfeld (1978) to estimate an hedonic housing prices function. It is reproduced in B. H. Baltagi and Chang (1994), table 2 (and in B. H. Baltagi (2005), pp. 172/4; B. H. Baltagi (2013), pp. 195/7 tables 9.1/3).

```
data("Hedonic", package = "plm")
Hed <- plm(mv~crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+blacks+lstat,
           data = Hedonic, model = "random", index = "townid")
summary(Hed)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = mv ~ crim + zn + indus + chas + nox + rm + age +
##     dis + rad + tax + ptratio + blacks + lstat, data = Hedonic,
##     model = "random", index = "townid")
##
## Unbalanced Panel: n = 92, T = 1-30, N = 506
##
## Effects:
##                    var std.dev share
## idiosyncratic 0.01696 0.13025 0.562
## individual    0.01324 0.11505 0.438
## theta:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2505  0.5483  0.6284  0.6141  0.7147  0.7976
##
## Residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.62902 -0.06712 -0.00156 -0.00216  0.06858  0.54973
##
## Coefficients:
##                 Estimate   Std. Error  z-value              Pr(>|z|)
## (Intercept)  9.685866695  0.197510264  49.0398 < 0.00000000000000022 ***
## crim        -0.007411967  0.001047812  -7.0738   0.00000000000150795 ***
```

```
## zn           0.000078877  0.000650012   0.1213             0.9034166
## indus        0.001556340  0.004034911   0.3857             0.6997051
## chasyes     -0.004424737  0.029211764  -0.1515             0.8796041
## nox         -0.005842506  0.001245183  -4.6921   0.00000270431168602 ***
## rm           0.009055167  0.001188629   7.6182   0.00000000000002573 ***
## age         -0.000857873  0.000467933  -1.8333             0.0667541 .
## dis         -0.144418433  0.044093739  -3.2753             0.0010557 **
## rad          0.095983935  0.026610945   3.6069             0.0003098 ***
## tax         -0.000377396  0.000176926  -2.1331             0.0329190 *
## ptratio     -0.029475776  0.009069842  -3.2499             0.0011546 **
## blacks       0.562775469  0.101973789   5.5188   0.00000003412743874 ***
## lstat       -0.291074917  0.023927306 -12.1650 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    987.94
## Residual Sum of Squares: 8.9988
## R-Squared:      0.99091
## Adj. R-Squared: 0.99067
## Chisq: 1199.5 on 13 DF, p-value: < 0.000000000000000222
```

Measures for the unbalancedness of a panel data set or the data used in estimated models are provided by function `punbalancedness`. It gives the measures γ and ν from Ahrens and Pincus (1981) where for both 1 represents balanced data and the more unbalanced the data the lower the value.

```
punbalancedness(Hed)
```

```
##     gamma        nu
## 0.4715336 0.5188292
```

Instrumental variable estimators

All of the models presented above may be estimated using instrumental variables. The instruments are specified at the end of the formula after a `|` sign (pipe).

The instrumental variables estimator used is indicated with the `inst.method` argument:

- `"bvk"`, from Balestra and Varadharajan–Krishnakumar (1987), the default value,
- `"baltagi"`, from B. H. Baltagi (1981),
- `"am"`, from Takeshi Amemiya and MaCurdy (1986),
- `"bms"`, from Trevor S. Breusch, Mizon, and Schmidt (1989).

An illustration is in the following example from B. H. Baltagi (2005), p. 120; B. H. Baltagi (2013), p. 137; B. H. Baltagi (2021), p. 165, table 7.3 (“G2SLS”).

```
data("Crime", package = "plm")
cr <- plm(lcrmrte ~ lprbarr + lpolpc + lprbconv + lprbpris + lavgsen +
          ldensity + lwcon + lwtuc + lwtrd + lwfir + lwser + lwmfg + lwfed +
          lwsta + lwloc + lpctymle + lpctmin + region + smsa + factor(year)
```

```
          | . - lprbarr - lpolpc + ltaxpc + lmix,
          data = Crime, model = "random")
summary(cr)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
## Instrumental variable estimation
##    (Balestra-Varadharajan-Krishnakumar's transformation)
##
## Call:
## plm(formula = lcrmrte ~ lprbarr + lpolpc + lprbconv + lprbpris +
##     lavgsen + ldensity + lwcon + lwtuc + lwtrd + lwfir + lwser +
##     lwmfg + lwfed + lwsta + lwloc + lpctymle + lpctmin + region +
##     smsa + factor(year) | . - lprbarr - lpolpc + ltaxpc + lmix,
##     data = Crime, model = "random")
##
## Balanced Panel: n = 90, T = 7, N = 630
##
## Effects:
##                   var std.dev share
## idiosyncratic 0.02227 0.14924 0.326
## individual    0.04604 0.21456 0.674
## theta: 0.7457
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -0.7485357 -0.0709883  0.0040648  0.0784455  0.4756273
##
## Coefficients:
##                  Estimate Std. Error z-value      Pr(>|z|)
## (Intercept)    -0.4538501  1.7029831 -0.2665      0.789852
## lprbarr        -0.4141383  0.2210496 -1.8735      0.060998 .
## lpolpc          0.5049461  0.2277778  2.2168      0.026634 *
## lprbconv       -0.3432506  0.1324648 -2.5913      0.009563 **
## lprbpris       -0.1900467  0.0733392 -2.5913      0.009560 **
## lavgsen        -0.0064389  0.0289407 -0.2225      0.823935
## ldensity        0.4343449  0.0711496  6.1047 0.00000000103 ***
## lwcon          -0.0042958  0.0414226 -0.1037      0.917403
## lwtuc           0.0444589  0.0215448  2.0636      0.039060 *
## lwtrd          -0.0085579  0.0419829 -0.2038      0.838476
## lwfir          -0.0040305  0.0294569 -0.1368      0.891166
## lwser           0.0105602  0.0215823  0.4893      0.624630
## lwmfg          -0.2018020  0.0839373 -2.4042      0.016208 *
## lwfed          -0.2134579  0.2151046 -0.9923      0.321029
## lwsta          -0.0601232  0.1203149 -0.4997      0.617275
## lwloc           0.1835363  0.1396775  1.3140      0.188846
## lpctymle       -0.1458703  0.2268086 -0.6431      0.520131
## lpctmin         0.1948763  0.0459385  4.2421 0.00002214292 ***
## regionwest     -0.2281821  0.1010260 -2.2586      0.023905 *
## regioncentral  -0.1987703  0.0607475 -3.2721      0.001068 **
```

```
## smsayes        -0.2595451  0.1499718 -1.7306      0.083518 .
## factor(year)82  0.0132147  0.0299924  0.4406      0.659500
## factor(year)83 -0.0847693  0.0320010 -2.6490      0.008074 **
## factor(year)84 -0.1062027  0.0387893 -2.7379      0.006183 **
## factor(year)85 -0.0977457  0.0511681 -1.9103      0.056097 .
## factor(year)86 -0.0719451  0.0605819 -1.1876      0.235004
## factor(year)87 -0.0396595  0.0758531 -0.5228      0.601081
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    30.169
## Residual Sum of Squares: 12.419
## R-Squared:      0.5923
## Adj. R-Squared: 0.57472
## Chisq: 542.478 on 26 DF, p-value: < 0.00000000000000222
```

The Hausman-Taylor model (see Hausman and Taylor (1981)) may be estimated with the `plm`[6] function by setting parameters `random.method = "ht"` and `inst.method = "baltagi"` like in the example below. The following replicates B. H. Baltagi (2005), pp. 129/30; B. H. Baltagi (2013), pp. 145/6, tables 7.4/5; B. H. Baltagi (2021), pp. 174/5 tables 7.5/6:

```
ht <- plm(lwage ~ wks + south + smsa + married + exp + I(exp ^ 2) +
              bluecol + ind + union + sex + black + ed |
              bluecol + south + smsa + ind + sex + black |
              wks + married + union + exp + I(exp ^ 2),
          data = Wages, index = 595,
          model = "random", random.method = "ht", inst.method = "baltagi")
summary(ht)
```

```
## Oneway (individual) effect Random Effect Model
##    (Hausman-Taylor's transformation)
## Instrumental variable estimation
##    (Baltagi's transformation)
##
## Call:
## plm(formula = lwage ~ wks + south + smsa + married + exp + I(exp^2) +
##     bluecol + ind + union + sex + black + ed | bluecol + south +
##     smsa + ind + sex + black | wks + married + union + exp +
##     I(exp^2), data = Wages, model = "random", random.method = "ht",
##     inst.method = "baltagi", index = 595)
##
## Balanced Panel: n = 595, T = 7, N = 4165
##
## Effects:
##                   var std.dev share
## idiosyncratic 0.02304 0.15180 0.025
## individual    0.88699 0.94180 0.975
## theta: 0.9392
##
```

```
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -12.643736  -0.466002   0.043285   0.524739  13.340263
##
## Coefficients:
##                 Estimate   Std. Error z-value             Pr(>|z|)
## (Intercept)  2.912726279  0.283652215 10.2687 < 0.00000000000000022 ***
## wks          0.000837403  0.000599732  1.3963              0.16263
## southyes     0.007439837  0.031955005  0.2328              0.81590
## smsayes     -0.041833367  0.018958129 -2.2066              0.02734 *
## marriedyes  -0.029850749  0.018979963 -1.5728              0.11578
## exp          0.113132791  0.002470954 45.7851 < 0.00000000000000022 ***
## I(exp^2)    -0.000418865  0.000054598 -7.6718   0.00000000000001696 ***
## bluecolyes  -0.020704707  0.013780948 -1.5024              0.13299
## ind          0.013603930  0.015237366  0.8928              0.37196
## unionyes     0.032771447  0.014908437  2.1982              0.02794 *
## sexfemale   -0.130923610  0.126658988 -1.0337              0.30129
## blackyes    -0.285747871  0.155701854 -1.8352              0.06647 .
## ed           0.137943957  0.021248489  6.4919   0.00000000008473689 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    243.04
## Residual Sum of Squares: 4163.6
## R-Squared:      0.60945
## Adj. R-Squared: 0.60833
## Chisq: 6891.87 on 12 DF, p-value: < 0.000000000000000222
```

Variable coefficients model

The `pvcm` function enables the estimation of variable coefficients models. Time or individual effects are introduced if argument `effect` is fixed to `"time"` or `"individual"` (the default value).

Coefficients are assumed to be fixed if `model="within"` or random if `model="random"`. In the first case, a different model is estimated for each individual (or time period). In the second case, the Swamy model (see Swamy (1970)) model is estimated. It is a generalized least squares model which uses the results of the previous model. Denoting $\hat{\beta}_i$ the vectors of coefficients obtained for each individual, we get:

$$\hat{\beta} = \left(\sum_{i=1}^{n} \left(\hat{\Delta} + \hat{\sigma}_i^2 (X_i^\top X_i)^{-1} \right)^{-1} \right) \left(\hat{\Delta} + \hat{\sigma}_i^2 (X_i^\top X_i)^{-1} \right)^{-1} \hat{\beta}_i$$

where $\hat{\sigma}_i^2$ is the unbiased estimator of the variance of the errors for individual i obtained from the preliminary estimation and:

$$\hat{\Delta} = \frac{1}{n-1} \sum_{i=1}^{n} \left(\hat{\beta}_i - \frac{1}{n} \sum_{i=1}^{n} \hat{\beta}_i \right) \left(\hat{\beta}_i - \frac{1}{n} \sum_{i=1}^{n} \hat{\beta}_i \right)^\top - \frac{1}{n} \sum_{i=1}^{n} \hat{\sigma}_i^2 (X_i^\top X_i)^{-1}$$

If this matrix is not positive-definite, the second term is dropped.

With the `Grunfeld` data, we get:

```
grun.varw <- pvcm(inv~value+capital, data=Grunfeld, model="within")
grun.varr <- pvcm(inv~value+capital, data=Grunfeld, model="random")
summary(grun.varr)
```

```
## Oneway (individual) effect Random coefficients model
##
## Call:
## pvcm(formula = inv ~ value + capital, data = Grunfeld, model = "random")
##
## Balanced Panel: n = 10, T = 20, N = 200
##
## Residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -211.486  -32.321   -4.283    9.048   12.714  579.216
##
## Estimated mean of the coefficients:
##              Estimate Std. Error z-value   Pr(>|z|)
## (Intercept) -9.629285  17.035040 -0.5653  0.5718946
## value        0.084587   0.019956  4.2387 0.00002248 ***
## capital      0.199418   0.052653  3.7874  0.0001522 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Estimated variance of the coefficients:
##             (Intercept)      value    capital
## (Intercept)  2344.24402 -0.6852340 -4.0276612
## value          -0.68523  0.0031182 -0.0011847
## capital        -4.02766 -0.0011847  0.0244824
##
## Total Sum of Squares: 474010000
## Residual Sum of Squares: 2194300
## Multiple R-Squared: 0.99537
## Chisq: 38.8364 on 2 DF, p-value: 0.0000000036878
```

Generalized method of moments estimator

The generalized method of moments is mainly used in panel data econometrics to estimate dynamic models (M. Arellano and Bond 1991; Holtz–Eakin, Newey, and Rosen 1988).

$$y_{it} = \rho y_{it-1} + \beta^\top x_{it} + \mu_i + \epsilon_{it}$$

The model is first differenced to get rid of the individual effect:

$$\Delta y_{it} = \rho \Delta y_{it-1} + \beta^\top \Delta x_{it} + \Delta \epsilon_{it}$$

Least squares are inconsistent because $\Delta\epsilon_{it}$ is correlated with Δy_{it-1} . y_{it-2} is a valid, but weak instrument (see Anderson and Hsiao (1981)). The GMM estimator uses the fact that the number of valid instruments is growing with t:

- $t = 3 : y_1$,
- $t = 4 : y_1, y_2$,
- $t = 5 : y_1, y_2, y_3$.

For individual i, the matrix of instruments is then:

$$W_i = \begin{pmatrix} y_1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & x_{i3} \\ 0 & y_1 & y_2 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & x_{i4} \\ 0 & 0 & 0 & y_1 & y_2 & y_3 & \dots & 0 & 0 & 0 & 0 & x_{i5} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \dots & \dots & y_1 & y_2 & \dots & y_{t-2} & x_{iT-2} \end{pmatrix}$$

The moment conditions are: $\sum_{i=1}^{n} W_i^\top e_i(\beta)$ where $e_i(\beta)$ is the vector of residuals for individual i. The GMM estimator minimizes:

$$\left(\sum_{i=1}^{n} e_i(\beta)^\top W_i \right) A \left(\sum_{i=1}^{n} W_i^\top e_i(\beta) \right)$$

where A is the weighting matrix of the moments.

One-step estimators are computed using a known weighting matrix. For the model in first differences, one uses:

$$A^{(1)} = \left(\sum_{i=1}^{n} W_i^\top H^{(1)} W_i \right)^{-1}$$

with:

$$H^{(1)} = d^\top d = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ 0 & -1 & 2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

Two-steps estimators are obtained using $H_i^{(2)} = \sum_{i=1}^{n} e_i^{(1)} e_i^{(1)\top}$ where $e_i^{(1)}$ are the residuals of the one step estimate.

Blundell and Bond (1998) show that with weak hypothesis on the data generating process, supplementary moment conditions exist for the equation in level:

$$y_{it} = \gamma y_{it-1} + \mu_i + \eta_{it}$$

More precisely, they show that $\Delta y_{it-2} = y_{it-2} - y_{it-3}$ is a valid instrument. The estimator is obtained using the residual vector in difference and in level:

$$e_i^+ = (\Delta e_i, e_i)$$

and the matrix of augmented moments:

$$Z_i^+ = \begin{pmatrix} Z_i & 0 & 0 & \dots & 0 \\ 0 & \Delta y_{i2} & 0 & \dots & 0 \\ 0 & 0 & \Delta y_{i3} & \dots & 0 \\ 0 & 0 & 0 & \dots & \Delta y_{iT-1} \end{pmatrix}$$

The moment conditions are then

$$\left(\sum_{i=1}^n Z_i^{+\top} \begin{pmatrix} \bar{e}_i(\beta) \\ e_i(\beta) \end{pmatrix}\right)^\top = \left(\sum_{i=1}^n y_{i1}\bar{e}_{i3}, \sum_{i=1}^n y_{i1}\bar{e}_{i4}, \sum_{i=1}^n y_{i2}\bar{e}_{i4}, \dots, \sum_{i=1}^n y_{i1}\bar{e}_{iT}, \sum_{i=1}^n y_{i2}\bar{e}_{iT}, \dots, \sum_{i=1}^n y_{iT-2}\bar{e}_{iT}, \sum_{i=1}^n \sum_{t=3}^T x_{it}\bar{e}_{it} \sum_{i=1}^n e_{i3}\Delta y_{i2}, \sum_{i=1}^n e_{i4}\Delta y_{i3}, \dots, \sum_{i=1}^n e_{iT}\Delta y_{iT-1}\right)^\top$$

The GMM estimator is provided by the `pgmm` function. By using a multi-part formula, the variables of the model and the lag structure are described.

In a GMM estimation, there are “normal instruments” and “GMM instruments”. GMM instruments are indicated in the second part of the formula. By default, all the variables of the model that are not used as GMM instruments are used as normal instruments, with the same lag structure; “normal” instruments may also be indicated in the third part of the formula.

The `effect` argument is either `NULL`, `"individual"` (the default), or `"twoways"`. In the first case, the model is estimated in levels. In the second case, the model is estimated in first differences to get rid of the individuals effects. In the last case, the model is estimated in first differences and time dummies are included.

The `model` argument specifies whether a one-step or a two-steps model is requested (`"onestep"` or `"twosteps"`).

The following example is from M. Arellano and Bond (1991). Employment is explained by past values of employment (two lags), current and first lag of wages and output and current value of capital.

```
emp.gmm <- pgmm(log(emp)~lag(log(emp), 1:2)+lag(log(wage), 0:1)+log(capital)+
                lag(log(output), 0:1) | lag(log(emp), 2:99),
                data = EmplUK, effect = "twoways", model = "twosteps")
summary(emp.gmm)
```

```
## Twoways effects Two-steps model Difference GMM
##
## Call:
## pgmm(formula = log(emp) ~ lag(log(emp), 1:2) + lag(log(wage),
##     0:1) + log(capital) + lag(log(output), 0:1) | lag(log(emp),
##     2:99), data = EmplUK, effect = "twoways", model = "twosteps")
```

```
##
## Unbalanced Panel: n = 140, T = 7-9, N = 1031
##
## Number of Observations Used: 611
## Residuals:
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.6190677 -0.0255683  0.0000000 -0.0001339  0.0332013  0.6410272
##
## Coefficients:
##                          Estimate Std. Error z-value    Pr(>|z|)
## lag(log(emp), 1:2)1      0.474151   0.185398  2.5575   0.0105437 *
## lag(log(emp), 1:2)2     -0.052967   0.051749 -1.0235   0.3060506
## lag(log(wage), 0:1)0    -0.513205   0.145565 -3.5256   0.0004225 ***
## lag(log(wage), 0:1)1     0.224640   0.141950  1.5825   0.1135279
## log(capital)             0.292723   0.062627  4.6741 0.000002953 ***
## lag(log(output), 0:1)0   0.609775   0.156263  3.9022 0.000095304 ***
## lag(log(output), 0:1)1 -0.446373   0.217302 -2.0542   0.0399605 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sargan test: chisq(25) = 30.11247 (p-value = 0.22011)
## Autocorrelation test (1): normal = -1.53845 (p-value = 0.12394)
## Autocorrelation test (2): normal = -0.2796829 (p-value = 0.77972)
## Wald test for coefficients: chisq(7) = 142.0353 (p-value = < 0.00000000000000222)
## Wald test for time dummies: chisq(6) = 16.97046 (p-value = 0.0093924)
```

The following example is from Blundell and Bond (1998). The “sys” estimator is obtained using `transformation = "ld"` for level and difference. The `robust` argument of the `summary` method enables to use the robust covariance matrix proposed by Windmeijer (2005). For all pgmm models, `robust = TRUE` is the default (but set in this example explicitly).

```
z2 <- pgmm(log(emp) ~ lag(log(emp), 1)+ lag(log(wage), 0:1) +
           lag(log(capital), 0:1) | lag(log(emp), 2:99) +
           lag(log(wage), 2:99) + lag(log(capital), 2:99),
           data = EmplUK, effect = "twoways", model = "onestep",
           transformation = "ld")
summary(z2, robust = TRUE)
```

```
## Twoways effects One-step model System GMM
##
## Call:
## pgmm(formula = log(emp) ~ lag(log(emp), 1) + lag(log(wage), 0:1) +
##     lag(log(capital), 0:1) | lag(log(emp), 2:99) + lag(log(wage),
##     2:99) + lag(log(capital), 2:99), data = EmplUK, effect = "twoways",
##     model = "onestep", transformation = "ld")
##
## Unbalanced Panel: n = 140, T = 7-9, N = 1031
##
## Number of Observations Used: 1642
```

```
## Residuals:
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.7530341 -0.0369030  0.0000000  0.0002882  0.0466069  0.6001503
##
## Coefficients:
##                           Estimate Std. Error z-value             Pr(>|z|)
## lag(log(emp), 1)          0.935605   0.026295 35.5810 < 0.00000000000000022 ***
## lag(log(wage), 0:1)0     -0.630976   0.118054 -5.3448    0.00000009050128 61 ***
## lag(log(wage), 0:1)1      0.482620   0.136887  3.5257            0.0004224 ***
## lag(log(capital), 0:1)0   0.483930   0.053867  8.9838 < 0.00000000000000022 ***
## lag(log(capital), 0:1)1 -0.424393   0.058479 -7.2572    0.00000000000039 52 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sargan test: chisq(100) = 118.763 (p-value = 0.097096)
## Autocorrelation test (1): normal = -4.808434 (p-value = 0.0000015212)
## Autocorrelation test (2): normal = -0.2800133 (p-value = 0.77947)
## Wald test for coefficients: chisq(5) = 11174.82 (p-value = < 0.000000000000000222)
## Wald test for time dummies: chisq(7) = 14.71138 (p-value = 0.039882)
```

General FGLS models

General FGLS estimators are based on a two-step estimation process: first an OLS model is estimated, then its residuals $\hat{u}_{it}$ are used to estimate an error covariance matrix more general than the random effects one for use in a feasible-GLS analysis. Formally, the estimated error covariance matrix is $\hat{V} = I_n \otimes \hat{\Omega}$, with

$$\hat{\Omega} = \sum_{i=1}^{n} \frac{\hat{u}_{it}\hat{u}_{it}^{\top}}{n}$$

(see Wooldridge (2002) 10.4.3 and 10.5.5).

This framework allows the error covariance structure inside every group (if `effect = "individual"`) of observations to be fully unrestricted and is therefore robust against any type of intragroup heteroskedasticity and serial correlation. This structure, by converse, is assumed identical across groups and thus general FGLS is inefficient under groupwise heteroskedasticity. Cross-sectional correlation is excluded a priori.

Moreover, the number of variance parameters to be estimated with $N = n \times T$ data points is $T(T+1)/2$, which makes these estimators particularly suited for situations where $n >> T$, as e.g., in labour or household income surveys, while problematic for "long" panels, where $\hat{V}$ tends to become singular and standard errors therefore become biased downwards.

In a pooled time series context (`effect = "time"`), symmetrically, this estimator is able to account for arbitrary cross-sectional correlation, provided that the latter is time-invariant (see Greene (2003) 13.9.1–2, pp. 321–2). In this case serial correlation has to be assumed away and the estimator is consistent with respect to the time dimension, keeping n fixed.

The function `pggls` estimates general FGLS models, with either fixed or "random" effects[7].

The "random effect" general FGLS is estimated by:

```
zz <- pggls(log(emp)~log(wage)+log(capital), data=EmplUK, model="pooling")
summary(zz)
```

```
## Oneway (individual) effect General FGLS model
##
## Call:
## pggls(formula = log(emp) ~ log(wage) + log(capital), data = EmplUK,
##     model = "pooling")
##
## Unbalanced Panel: n = 140, T = 7-9, N = 1031
##
## Residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.80696 -0.36552  0.06181  0.03230  0.44279  1.58719
##
## Coefficients:
##               Estimate Std. Error z-value              Pr(>|z|)
## (Intercept)   2.023480   0.158468 12.7690 < 0.00000000000000022 ***
## log(wage)    -0.232329   0.048001 -4.8401           0.000001298 ***
## log(capital)  0.610484   0.017434 35.0174 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Total Sum of Squares: 1853.6
## Residual Sum of Squares: 402.55
## Multiple R-squared: 0.78283
```

The fixed effects `pggls` (see Wooldridge (2002), p. 276) is based on the estimation of a within model in the first step; the rest follows as above. It is estimated by:

```
zz <- pggls(log(emp)~log(wage)+log(capital), data=EmplUK, model="within")
```

The `pggls` function is similar to `plm` in many respects. An exception is that the estimate of the group covariance matrix of errors (`zz$sigma`, a matrix, not shown) is reported in the model objects instead of the usual estimated variances of the two error components.

Tests

As sketched in Section linear panel model, specification testing in panel models involves essentially testing for poolability, for individual or time unobserved effects and for correlation between these latter and the regressors (Hausman-type tests). As for the other usual diagnostic checks, we provide a suite of serial correlation tests, while not touching on the issue of heteroskedasticity testing. Instead, we provide heteroskedasticity-robust covariance estimators, to be described in subsection robust covariance matrix estimation.

Tests of poolability

`pooltest` tests the hypothesis that the same coefficients apply to each individual. It is a standard F test, based on the comparison of a model obtained for the full sample and a model based on the estimation of an equation for each individual. The first argument of `pooltest` is a `plm` object. The second argument is a `pvcm` object obtained with `model="within"`. If the first argument is a pooling model, the test applies to all the coefficients (including the intercepts), if it is a within model, different intercepts are assumed.

To test the hypothesis that all the coefficients in the `Grunfeld` example, excluding the intercepts, are equal, we use :

```
znp <- pvcm(inv ~ value + capital, data = Grunfeld, model = "within")
zplm <- plm(inv ~ value + capital, data = Grunfeld, model = "within")
pooltest(zplm, znp)
```

```
##
##  F statistic
##
## data:  inv ~ value + capital
## F = 5.7805, df1 = 18, df2 = 170, p-value = 0.0000000001219
## alternative hypothesis: unstability
```

The same test can be computed using a formula as first argument of the `pooltest` function:

```
pooltest(inv ~ value + capital, data = Grunfeld, model = "within")
```

Tests for individual and time effects

`plmtest` implements Lagrange multiplier tests of individual or/and time effects based on the results of the pooling model. Its main argument is a `plm` object (the result of a pooling model) or a formula.

Two additional arguments can be added to indicate the kind of test to be computed. The argument `type` is one of:

- `"honda"`: Honda (1985), the default value,
- `"bp"`: T. S. Breusch and Pagan (1980),
- `"kw"`: King and Wu (1997)[8],
- `"ghm"`: Gourieroux, Holly, and Monfort (1982).

The effects tested are indicated with the `effect` argument (one of `"individual"`, `"time"`, or `"twoways"`). The test statistics implemented are also suitable for unbalanced panels.[9]

To test the presence of individual and time effects in the `Grunfeld` example, using the Gourieroux, Holly, and Monfort (1982) test, we use:

```
g <- plm(inv ~ value + capital, data=Grunfeld, model="pooling")
plmtest(g, effect="twoways", type="ghm")
```

```
##
```

```
##  Lagrange Multiplier Test - two-ways effects (Gourieroux, Holly and
##  Monfort)
##
## data:  inv ~ value + capital
## chibarsq = 798.16, df0 = 0.00, df1 = 1.00, df2 = 2.00, w0 = 0.25, w1 =
## 0.50, w2 = 0.25, p-value < 0.00000000000000022
## alternative hypothesis: significant effects
```

or

```
plmtest(inv~value+capital, data=Grunfeld, effect="twoways", type="ghm")
```

`pFtest` computes F tests of effects based on the comparison of the within and the pooling model. Its main arguments are either two `plm` objects (a pooling and a within model) or a formula.

```
gw <- plm(inv ~ value + capital, data=Grunfeld, effect="twoways", model="within")
gp <- plm(inv ~ value + capital, data=Grunfeld, model="pooling")
pFtest(gw, gp)
```

```
##
##  F test for twoways effects
##
## data:  inv ~ value + capital
## F = 17.403, df1 = 28, df2 = 169, p-value < 0.00000000000000022
## alternative hypothesis: significant effects
```

```
pFtest(inv~value+capital, data=Grunfeld, effect="twoways")
```

Hausman test

`phtest` computes the Hausman test (at times also called Durbin–Wu–Hausman test) which is based on the comparison of two sets of estimates (see Hausman (1978)).

Its main arguments are two `panelmodel` objects or a formula. A classical application of the Hausman test for panel data is to compare the fixed and the random effects models:

```
gw <- plm(inv ~ value + capital, data = Grunfeld, model="within")
gr <- plm(inv ~ value + capital, data = Grunfeld, model="random")
phtest(gw, gr)
```

```
##
##  Hausman Test
##
## data:  inv ~ value + capital
## chisq = 2.3304, df = 2, p-value = 0.3119
```

```
## alternative hypothesis: one model is inconsistent
```

The command also supports the auxiliary-regression-based version as described in, e.g., Wooldridge (2010) Sec.10.7.3 by using the formula interface and setting argument `test = "aux"`. This auxiliary-regression-based version can be robustified by specifying a robust covariance estimator as a function through the argument `vcov`:

```
phtest(inv ~ value + capital, data = Grunfeld, method = "aux", vcov = vcovHC)
```

```
##
##  Regression-based Hausman test, vcov: vcovHC
##
## data:  inv ~ value + capital
## chisq = 8.2998, df = 2, p-value = 0.01577
## alternative hypothesis: one model is inconsistent
```

Tests of serial correlation

A model with individual effects has composite errors that are serially correlated by definition. The presence of the time-invariant error component[10] gives rise to serial correlation which does not die out over time, thus standard tests applied on pooled data always end up rejecting the null of spherical residuals[11]. There may also be serial correlation of the "usual" kind in the idiosyncratic error terms, e.g., as an AR(1) process. By "testing for serial correlation" we mean testing for this latter kind of dependence.

For these reasons, the subjects of testing for individual error components and for serially correlated idiosyncratic errors are closely related. In particular, simple (*marginal*) tests for one direction of departure from the hypothesis of spherical errors usually have power against the other one: in case it is present, they are substantially biased towards rejection. *Joint* tests are correctly sized and have power against both directions, but usually do not give any information about which one actually caused rejection. *Conditional* tests for serial correlation that take into account the error components are correctly sized under presence of both departures from sphericity and have power only against the alternative of interest. While most powerful if correctly specified, the latter, based on the likelihood framework, are crucially dependent on normality and homoskedasticity of the errors.

In `plm` we provide a number of joint, marginal and conditional ML-based tests, plus some semiparametric alternatives which are robust vs. heteroskedasticity and free from distributional assumptions.

Unobserved effects test

The unobserved effects test *à la Wooldridge* (see Wooldridge (2002) 10.4.4), is a semiparametric test for the null hypothesis that $\sigma_\mu^2 = 0$, i.e. that there are no unobserved effects in the residuals. Given that under the null the covariance matrix of the residuals for each individual is diagonal, the test statistic is based on the average of elements in the upper (or lower) triangle of its estimate, diagonal excluded: $n^{-1/2} \sum_{i=1}^{n} \sum_{t=1}^{T-1} \sum_{s=t+1}^{T} \hat{u}_{it}\hat{u}_{is}$ (where $\hat{u}$ are the pooled OLS residuals), which must be "statistically close" to zero under the null, scaled by its standard deviation:

$$W = \frac{\sum_{i=1}^{n} \sum_{t=1}^{T-1} \sum_{s=t+1}^{T} \hat{u}_{it}\hat{u}_{is}}{[\sum_{i=1}^{n} (\sum_{t=1}^{T-1} \sum_{s=t+1}^{T} \hat{u}_{it}\hat{u}_{is})^2]^{1/2}}$$

This test is (n-) asymptotically distributed as a standard normal regardless of the distribution of the errors. It does also not rely on homoskedasticity.

It has power both against the standard random effects specification, where the unobserved effects are constant within every group, as well as against any kind of serial correlation. As such, it "nests" both random effects and serial correlation tests, trading some power against more specific alternatives in exchange for robustness.

While not rejecting the null favours the use of pooled OLS, rejection may follow from serial correlation of different kinds, and in particular, quoting Wooldridge (2002), "should not be interpreted as implying that the random effects error structure *must* be true".

Below, the test is applied to the data and model in Munnell (1990):

```
pwtest(log(gsp)~log(pcap)+log(pc)+log(emp)+unemp, data=Produc)
```

```
##
##  Wooldridge's test for unobserved individual effects
##
## data:  formula
## z = 3.9383, p-value = 0.00008207
## alternative hypothesis: unobserved effect
```

Locally robust tests for serial correlation or random effects

The presence of random effects may affect tests for residual serial correlation, and the opposite. One solution is to use a joint test, which has power against both alternatives. A joint LM test for random effects *and* serial correlation under normality and homoskedasticity of the idiosyncratic errors has been derived by B. Baltagi and Li (1991) and B. Baltagi and Li (1995) and is implemented as an option in `pbsytest`:

```
pbsytest(log(gsp)~log(pcap)+log(pc)+log(emp)+unemp, data=Produc, test="j")
```

```
##
##  Baltagi and Li AR-RE joint test
##
## data:  formula
## chisq = 4187.6, df = 2, p-value < 0.00000000000000022
## alternative hypothesis: AR(1) errors or random effects
```

Rejection of the joint test, though, gives no information on the direction of the departure from the null hypothesis, i.e.: is rejection due to the presence of serial correlation, of random effects or of both?

Bera, Sosa–Escudero, and Yoon (2001) (hereafter BSY) derive locally robust tests both for individual random effects and for first-order serial correlation in residuals as "corrected" versions of the standard

LM test (see `plmtest`). While still dependent on normality and homoskedasticity, these are robust to *local* departures from the hypotheses of, respectively, no serial correlation or no random effects. The authors observe that, although suboptimal, these tests may help detecting the right direction of the departure from the null, thus complementing the use of joint tests. Moreover, being based on pooled OLS residuals, the BSY tests are computationally far less demanding than likelihood-based conditional tests.

On the other hand, the statistical properties of these "locally corrected" tests are inferior to those of the non-corrected counterparts when the latter are correctly specified. If there is no serial correlation, then the optimal test for random effects is the likelihood-based LM test of Breusch and Godfrey (with refinements by Honda, see `plmtest`), while if there are no random effects the optimal test for serial correlation is, again, Breusch-Godfrey's test[12]. If the presence of a random effect is taken for granted, then the optimal test for serial correlation is the likelihood-based conditional LM test of B. Baltagi and Li (1995) (see `pbltest`).

The serial correlation version is the default:

```
pbsytest(log(gsp)~log(pcap)+log(pc)+log(emp)+unemp, data=Produc)
```

```
##
##  Bera, Sosa-Escudero and Yoon locally robust test
##
## data:  formula
## chisq = 52.636, df = 1, p-value = 0.0000000000004015
## alternative hypothesis: AR(1) errors sub random effects
```

The BSY test for random effects is implemented in the one-sided version[13], which takes heed that the variance of the random effect must be non-negative:

```
pbsytest(log(gsp)~log(pcap)+log(pc)+log(emp)+unemp, data=Produc, test="re")
```

```
##
##  Bera, Sosa-Escudero and Yoon locally robust test (one-sided)
##
## data:  formula
## z = 57.914, p-value < 0.00000000000000022
## alternative hypothesis: random effects sub AR(1) errors
```

Conditional LM test for AR(1) or MA(1) errors under random effects

B. Baltagi and Li (1991) and B. Baltagi and Li (1995) derive a Lagrange multiplier test for serial correlation in the idiosyncratic component of the errors under (normal, heteroskedastic) random effects. Under the null of serially uncorrelated errors, the test turns out to be identical for both the alternative of AR(1) and MA(1) processes. One- and two-sided versions are provided, the one-sided having power against positive serial correlation only. The two-sided is the default, while for the other one must specify the `alternative` option to `"onesided"`:

```
pbltest(log(gsp)~log(pcap)+log(pc)+log(emp)+unemp,
        data=Produc, alternative="onesided")
```

```
##
##  Baltagi and Li one-sided LM test
##
## data:  log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp
## z = 21.69, p-value < 0.00000000000000022
## alternative hypothesis: AR(1)/MA(1) errors in RE panel model
```

As usual, the LM test statistic is based on residuals from the maximum likelihood estimate of the restricted model (random effects with serially uncorrelated errors). In this case, though, the restricted model cannot be estimated by OLS anymore, therefore the testing function depends on `lme()` in the `nlme` package for estimation of a random effects model by maximum likelihood. For this reason, the test is applicable only to balanced panels.

No test has been implemented to date for the symmetric hypothesis of no random effects in a model with errors following an AR(1) process, but an asymptotically equivalent likelihood ratio test is available in the `nlme` package (see Section plm versus nlme and lme4).

General serial correlation tests

A general testing procedure for serial correlation in fixed effects (FE), random effects (RE) and pooled-OLS panel models alike can be based on considerations in Wooldridge (2002), 10.7.2.

Recall that `plm` model objects are the result of OLS estimation performed on "demeaned" data, where, in the case of individual effects (else symmetric), this means time-demeaning for the FE (`within`) model, quasi-time-demeaning for the RE (`random`) model and original data, with no demeaning at all, for the pooled OLS (`pooling`) model (see Section software approach).

For the random effects model, Wooldridge (2002) observes that under the null of homoskedasticity and no serial correlation in the idiosyncratic errors, the residuals from the quasi-demeaned regression must be spherical as well. Else, as the individual effects are wiped out in the demeaning, any remaining serial correlation must be due to the idiosyncratic component. Hence, a simple way of testing for serial correlation is to apply a standard serial correlation test to the quasi-demeaned model. The same applies in a pooled model, w.r.t. the original data.

The FE case needs some qualification. It is well-known that if the original model's errors are uncorrelated then FE residuals are negatively serially correlated, with $cor(\hat{u}_{it}, \hat{u}_{is}) = -1/(T-1)$ for each t, s (see Wooldridge (2002) 10.5.4). This correlation clearly dies out as T increases, so this kind of AR test is applicable to `within` model objects only for T "sufficiently large"[14]. On the converse, in short panels the test gets severely biased towards rejection (or, as the induced correlation is negative, towards acceptance in the case of the one-sided DW test with `alternative="greater"`). See below for a serial correlation test applicable to "short" FE panel models.

`plm` objects retain the "demeaned" data, so the procedure is straightforward for them. The wrapper functions `pbgtest` and `pdwtest` re-estimate the relevant quasi-demeaned model by OLS and apply, respectively, standard Breusch-Godfrey and Durbin-Watson tests from package `lmtest`:

```
pbgtest(grun.fe, order = 2)
```

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  inv ~ value + capital
## chisq = 42.587, df = 2, p-value = 0.0000000005655
## alternative hypothesis: serial correlation in idiosyncratic errors
```

The tests share the features of their OLS counterparts, in particular the `pbgtest` allows testing for higher-order serial correlation, which might turn useful, e.g., on quarterly data. Analogously, from the point of view of software, as the functions are simple wrappers towards `bgtest` and `dwtest`, all arguments from the latter two apply and may be passed on through the ellipsis (the `...` argument).

Wooldridge's test for serial correlation in "short" FE panels

For the reasons reported above, under the null of no serial correlation in the errors, the residuals of a FE model must be negatively serially correlated, with $cor(\hat{\epsilon}_{it}, \hat{\epsilon}_{is}) = -1/(T-1)$ for each t, s. Wooldridge suggests basing a test for this null hypothesis on a pooled regression of FE residuals on themselves, lagged one period:

$$\hat{\epsilon}_{i,t} = \alpha + \delta \hat{\epsilon}_{i,t-1} + \eta_{i,t}$$

Rejecting the restriction $\delta = -1/(T-1)$ makes us conclude against the original null of no serial correlation.

The building blocks available in `plm` make it easy to construct a function carrying out this procedure: first the FE model is estimated and the residuals retrieved, then they are lagged and a `pooling` AR(1) model is estimated. The test statistic is obtained by applying the above restriction on δ and supplying a heteroskedasticity- and autocorrelation-consistent covariance matrix (`vcovHC` with the appropriate options, in particular `method="arellano"`)[15].

```
pwartest(log(emp) ~ log(wage) + log(capital), data=EmplUK)
```

```
##
##  Wooldridge's test for serial correlation in FE panels
##
## data:  plm.model
## F = 312.3, df1 = 1, df2 = 889, p-value < 0.00000000000000022
## alternative hypothesis: serial correlation
```

The test is applicable to any FE panel model, and in particular to "short" panels with small T and large n.

Wooldridge's first-difference-based test

In the context of the first difference model, Wooldridge (2002), 10.6.3 proposes a serial correlation test that can also be seen as a specification test to choose the most efficient estimator between fixed effects (`within`) and first difference (`fd`).

The starting point is the observation that if the idiosyncratic errors of the original model u_{it} are uncorrelated, the errors of the (first) differenced model[16] $e_{it} \equiv u_{it} - u_{i,t-1}$ will be correlated, with $cor(e_{it}, e_{i,t-1}) = -0.5$, while any time-invariant effect, "fixed" or "random", is wiped out in the differencing. So a serial correlation test for models with individual effects of any kind can be based on estimating the model

$$\hat{u}_{i,t} = \delta \hat{u}_{i,t-1} + \eta_{i,t}$$

and testing the restriction $\delta = -0.5$, corresponding to the null of no serial correlation. Drukker (2003) provides Monte Carlo evidence of the good empirical properties of the test.

On the other extreme (see Wooldridge (2002) 10.6.1), if the differenced errors e_{it} are uncorrelated, as by definition $u_{it} = u_{i,t-1} + e_{it}$, then u_{it} is a random walk. In this latter case, the most efficient estimator is the first difference (`fd`) one; in the former case, it is the fixed effects one (`within`).

The function `pwfdtest` allows testing either hypothesis: the default behaviour `h0="fd"` is to test for serial correlation in *first-differenced* errors:

```
pwfdtest(log(emp) ~ log(wage) + log(capital), data=EmplUK)
```

```
##
##  Wooldridge's first-difference test for serial correlation in panels
##
## data:  plm.model
## F = 1.5251, df1 = 1, df2 = 749, p-value = 0.2172
## alternative hypothesis: serial correlation in differenced errors
```

while specifying `h0="fe"` the null hypothesis becomes no serial correlation in *original* errors, which is similar to the `pwartest`.

```
pwfdtest(log(emp) ~ log(wage) + log(capital), data=EmplUK, h0="fe")
```

```
##
##  Wooldridge's first-difference test for serial correlation in panels
##
## data:  plm.model
## F = 131.55, df1 = 1, df2 = 749, p-value < 0.00000000000000022
## alternative hypothesis: serial correlation in original errors
```

Not rejecting one of the two is evidence in favour of using the estimator corresponding to `h0`. Should the truth lie in the middle (both rejected), whichever estimator is chosen will have serially correlated errors: therefore it will be advisable to use the autocorrelation-robust covariance estimators from the subsection robust covariance matrix estimation in inference.

Tests for cross-sectional dependence

Next to the more familiar issue of serial correlation, over the last years a growing body of literature has been dealing with cross-sectional dependence (henceforth: XSD) in panels, which can arise, e.g., if

individuals respond to common shocks (as in the literature on *factor models*) or if spatial diffusion processes are present, relating individuals in a way depending on a measure of distance (*spatial models*).

The subject is huge, and here we touch only some general aspects of misspecification testing and valid inference. If XSD is present, the consequence is, at a minimum, inefficiency of the usual estimators and invalid inference when using the standard covariance matrix[17]. The plan is to have in `plm` both misspecification tests to detect XSD and robust covariance matrices to perform valid inference in its presence, like in the serial dependence case. For now, though, only misspecification tests are included.

CD and LM-type tests for global cross-sectional dependence

The function `pcdtest` implements a family of XSD tests which can be applied in different settings, ranging from those where T grows large with n fixed to "short" panels with a big n dimension and a few time periods. All are based on (transformations of–) the product-moment correlation coefficient of a model's residuals, defined as

$$\hat{\rho}_{ij} = \frac{\sum_{t=1}^{T} \hat{u}_{it}\hat{u}_{jt}}{(\sum_{t=1}^{T} \hat{u}_{it}^2)^{1/2}(\sum_{t=1}^{T} \hat{u}_{jt}^2)^{1/2}}$$

i.e., as averages over the time dimension of pairwise correlation coefficients for each pair of cross-sectional units.

The Breusch-Pagan (T. S. Breusch and Pagan 1980) LM test, based on the squares of ρ_{ij}, is valid for $T \to \infty$ with n fixed; defined as

$$LM = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} T_{ij}\hat{\rho}_{ij}^2$$

where in the case of an unbalanced panel only pairwise complete observations are considered, and $T_{ij} = min(T_i, T_j)$ with T_i being the number of observations for individual i; else, if the panel is balanced, $T_{ij} = T$ for each i, j. The test is distributed as $\chi^2_{n(n-1)/2}$. It is inappropriate whenever the n dimension is "large". A scaled version, applicable also if $T \to \infty$ and *then* $n \to \infty$ (as in some pooled time series contexts), is defined as

$$SCLM = \sqrt{\frac{1}{n(n-1)}}(\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} T_{ij}\hat{\rho}_{ij}^2 - 1)$$

and distributed as a standard normal (see M. H. Pesaran (2004)).

A bias-corrected scaled version, $BCSCLM$, for the *fixed effect model with individual effects* only is also available which is simply the $SCLM$ with a term correcting for the bias (Badi H. Baltagi, Feng, and Kao (2012))[18]. This statistic is also asymptotically distributed as standard normal.

$$BCSCLM = \sqrt{\frac{1}{n(n-1)}}(\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} T_{ij}\hat{\rho}_{ij}^2 - 1) - \frac{n}{2(T-1)}$$

Pesaran's (M. H. Pesaran (2004), M. Hashem Pesaran (2015)) CD test

$$CD = \sqrt{\frac{2}{n(n-1)}}(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sqrt{T_{ij}}\hat{\rho}_{ij})$$

based on ρ_{ij} without squaring (also distributed as a standard normal) is appropriate both in n– and in T–asymptotic settings. It has remarkable properties in samples of any practically relevant size and is robust to a variety of settings. The only big drawback is that the test loses power against the alternative of cross-sectional dependence if the latter is due to a factor structure with factor loadings averaging zero, that is, some units react positively to common shocks, others negatively.

The default version of the test is `"cd"` yielding Pesaran's CD test. These tests are originally meant to use the residuals of separate estimation of one time-series regression for each cross-sectional unit, so this is the default behaviour of `pcdtest`.

```
pcdtest(inv~value+capital, data=Grunfeld)
```

```
##
##  Pesaran CD test for cross-sectional dependence in panels
##
## data:  inv ~ value + capital
## z = 5.3401, p-value = 0.00000009292
## alternative hypothesis: cross-sectional dependence
```

If a different model specification (`within`, `random`, ...) is assumed consistent, one can resort to its residuals for testing[19] by specifying the relevant `model` type. The main argument of this function may be either a model of class `panelmodel` or a `formula` and a `data.frame`; in the second case, unless `model` is set to `NULL`, all usual parameters relative to the estimation of a `plm` model may be passed on. The test is compatible with any consistent `panelmodel` for the data at hand, with any specification of `effect`. E.g., specifying `effect = "time"` or `effect = "twoways"` allows to test for residual cross-sectional dependence after the introduction of time fixed effects to account for common shocks.

```
pcdtest(inv~value+capital, data=Grunfeld, model="within")
```

```
##
##  Pesaran CD test for cross-sectional dependence in panels
##
## data:  inv ~ value + capital
## z = 4.6612, p-value = 0.000003144
## alternative hypothesis: cross-sectional dependence
```

If the time dimension is insufficient and `model=NULL`, the function defaults to estimation of a `within` model and issues a warning.

CD(p) test for local cross-sectional dependence

A *local* variant of the CD test, called $CD(p)$ test (M. H. Pesaran 2004), takes into account an

appropriate subset of *neighbouring* cross-sectional units to check the null of no XSD against the alternative of *local* XSD, i.e. dependence between neighbours only. To do so, the pairs of neighbouring units are selected by means of a binary proximity matrix like those used in spatial models. In the original paper, a regular ordering of observations is assumed, so that the m-th cross-sectional observation is a neighbour to the $(m-1)$-th and to the $(m+1)$-th. Extending the $CD(p)$ test to irregular lattices, we employ the binary proximity matrix as a selector for discarding the correlation coefficients relative to pairs of observations that are not neighbours in computing the CD statistic. The test is then defined as

$$CD = \sqrt{\frac{1}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w(p)_{ij}}}\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}[w(p)]_{ij}\sqrt{T_{ij}}\hat{\rho}_{ij}\right)$$

where $[w(p)]_{ij}$ is the (i,j)-th element of the p-th order proximity matrix, so that if h, k are not neighbours, $[w(p)]_{hk} = 0$ and $\hat{\rho}_{hk}$ gets "killed"; this is easily seen to reduce to formula (14) in Pesaran (M. H. Pesaran 2004) for the special case considered in that paper. The same can be applied to the LM, $SCLM$, and $BCSCLM$ tests.

Therefore, the *local* version of either test can be computed supplying an $n \times n$ matrix (of any kind coercible to `logical`), providing information on whether any pair of observations are neighbours or not, to the `w` argument. If `w` is supplied, only neighbouring pairs will be used in computing the test; else, `w` will default to `NULL` and all observations will be used. The matrix needs not really be binary, so commonly used "row-standardized" matrices can be employed as well: it is enough that neighbouring pairs correspond to nonzero elements in `w` [20].

Panel unit root tests

Overview of functions for panel unit root testing

Below, first an overview is provided which tests are implemented per functions. A theoretical treatment is given for a few of those tests later on. The package `plm` offers several panel unit root tests contained in three functions:

- `purtest` (Levin-Lin-Chu test, IPS test, several Fisher-type tests, Hadri's test),
- `cipstest` (cross-sectionally augmented IPS test), and
- `phansitest` (Simes' test).

While `purtest` implements various tests which can be selected via its `test` argument, `cipstest` and `phansitest` are functions for a specific test each.

Function `purtest` offers the following tests by setting argument `test` to:

- `"levinlin"` (default), for the Levin-Lin-Chu test (Levin, Lin, and Chu (2002)), see below for a theoretical exposition (Levin-Lin-Chu test)),
- `"ips"`, for Im-Pesaran-Shin (IPS) test by Im, Pesaran, and Shin (2003), see below for a theoretical exposition (Im-Pesaran-Shin test)),
- `"madwu"`, is the inverse χ^2 test by Maddala and Wu (1999), also called P test by Choi (2001),
- `"Pm"`, is the modified P test proposed by Choi (2001) for large N,
- `"invnormal"`, is the inverse normal test (Choi (2001)),
- `"logit"`, is the logit test (Choi (2001)),
- `"hadri"`, for Hadri's test (Hadri (2000)).

The tests in `purtest` are often called first generation panel unit root tests as they do assume absence of cross-sectional correlation; all these, except Hadri's test, are based on the estimation of augmented Dickey-Fuller (ADF) regressions for each time series. A statistic is then computed using the t-statistics associated with the lagged variable. I a different manner, the Hadri residual-based LM statistic is the cross-sectional average of individual KPSS statistics (Kwiatkowski et al. (1992)), standardized by their asymptotic mean and standard deviation. Among the tests in `purtest`, `"madwu"`, `"Pm"`, `"invormal"`, and `"logit"` are Fisher-type tests.[21]

`purtest` returns an object of class `"purtest"` which contains details about the test performed, among them details about the individual regressions/statistics for the test. Associated `summary` and `print.summary` methods can be used to extract/display the additional information.

Function `cipstest` implements Pesaran's (M. Hashem Pesaran (2007)) cross-sectionally augmented version of the Im-Pesaran-Shin panel unit root test and is a so-called second-generation panel unit root test.

Function `phansitest` implements the idea of Hanck (2013) to apply Simes' testing approach for intersection of individual hypothesis tests to panel unit root testing, see below for a more thorough treatment of Simes' approach for intersecting hypotheses.

Preliminary results

We consider the following model:

$$y_{it} = \delta y_{it-1} + \sum_{L=1}^{p_i} \theta_i \Delta y_{it-L} + \alpha_{mi} d_{mt} + \epsilon_{it}$$

The unit root hypothesis is $\rho = 1$. The model can be rewritten in difference:

$$\Delta y_{it} = \rho y_{it-1} + \sum_{L=1}^{p_i} \theta_i \Delta y_{it-L} + \alpha_{mi} d_{mt} + \epsilon_{it}$$

So that the unit-root hypothesis is now $\rho = 0$.

Some of the unit-root tests for panel data are based on preliminary results obtained by running the above Augmented Dickey-Fuller (ADF) regression.

First, we have to determine the optimal number of lags p_i for each time-series. Several possibilities are available. They all have in common that the maximum number of lags have to be chosen first. Then, p_i can be chosen by using:

- the Schwarz information criterion (SIC) (also known as Bayesian information criterion (BIC)),
- the Akaike information criterion (AIC),
- the Hall's method, which consist in removing the higher lags while they are not significant.

The ADF regression is run on $T - p_i - 1$ observations for each individual, so that the total number of observations is $n \times \tilde{T}$ where $\tilde{T} = T - p_i - 1$

$\bar{p}$ is the average number of lags. Call e_i the vector of residuals.

Estimate the variance of the ϵ_i as:

$$\hat{\sigma}^2_{\epsilon_i} = \frac{\sum_{t=p_i+1}^{T} e^2_{it}}{df_i}$$

Levin-Lin-Chu model

Then, as per Levin, Lin, and Chu (2002), compute artificial regressions of Δy_{it} and y_{it-1} on Δy_{it-L} and d_{mt} and get the two vectors of residuals z_{it} and v_{it} .

Standardize these two residuals and run the pooled regression of $z_{it}/\hat{\sigma}_i$ on $v_{it}/\hat{\sigma}_i$ to get $\hat{\rho}$, its standard deviation $\hat{\sigma}(\hat{\rho})$ and the t-statistic $t_{\hat{\rho}} = \hat{\rho}/\hat{\sigma}(\hat{\rho})$.

Compute the long run variance of y_i :

$$\hat{\sigma}^2_{yi} = \frac{1}{T-1}\sum_{t=2}^{T}\Delta y^2_{it} + 2\sum_{L=1}^{\bar{K}} w_{\bar{K}L}\left[\frac{1}{T-1}\sum_{t=2+L}^{T}\Delta y_{it}\,\Delta y_{it-L}\right]$$

Define $\bar{s}_i$ as the ratio of the long and short term variance and $\bar{s}$ the mean for all the individuals of the sample

$$s_i = \frac{\hat{\sigma}_{yi}}{\hat{\sigma}_{\epsilon_i}}$$

$$\bar{s} = \frac{\sum_{i=1}^{n} s_i}{n}$$

$$t^*_\rho = \frac{t_\rho - n\bar{T}\bar{s}\hat{\sigma}^{-2}_{\tilde{\epsilon}}\hat{\sigma}(\hat{\rho})\mu^*_{m\tilde{T}}}{\sigma^*_{m\tilde{T}}}$$

follows a normal distribution under the null hypothesis of stationarity. $\mu^*_{m\tilde{T}}$ and $\sigma^*_{m\tilde{T}}$ are given in table 2 of the original paper and are also available in the package.

An example how the Levin-Lin-Chu test is performed with `purtest` using a lag of 2 and intercept and a time trend as exogenous variables in the ADF regressions is:

```
data("HousePricesUS", package = "pder")
lprice <- log(pdata.frame(HousePricesUS)$price)
(lev <- purtest(lprice, test = "levinlin", lags = 2, exo = "trend"))
```

```
##
##  Levin-Lin-Chu Unit-Root Test (ex. var.: Individual Intercepts and
##  Trend)
##
## data:  lprice
## z = -1.2573, p-value = 0.1043
## alternative hypothesis: stationarity
```

```
summary(lev) ### gives details
```

```
## Levin-Lin-Chu Unit-Root Test
## Exogenous variables: Individual Intercepts and Trend
## User-provided lags
## statistic: -1.257
## p-value: 0.104
##
##    lags obs          rho        trho      p.trho     sigma2ST     sigma2LT
## 1     2  26 -0.092065357 -1.66309731 0.767613204 0.0003143120 0.0004013788
## 4     2  26 -0.124093984 -1.29563385 0.888755668 0.0010950144 0.0014736172
## 5     2  26 -0.104647566 -1.10814627 0.926357866 0.0007296044 0.0007451534
## 6     2  26 -0.219022744 -2.94312106 0.148774635 0.0007716609 0.0048254402
## 8     2  26 -0.052471794 -0.95375744 0.948405601 0.0006375257 0.0028152736
## 9     2  26 -0.181914333 -2.73331072 0.222919642 0.0021489671 0.0064455696
## 10    2  26 -0.232215125 -3.37321191 0.054989191 0.0005566400 0.0024147067
## 11    2  26 -0.356452679 -4.35943612 0.002479709 0.0008542529 0.0045574510
## 12    2  26  0.279936991  1.83482002 0.999998365 0.0004172617 0.0012951914
## 13    2  26 -0.062610441 -0.84216587 0.960499065 0.0003168316 0.0002981994
## 16    2  26 -0.159254884 -2.29683734 0.435109226 0.0007437190 0.0010203969
## 17    2  26 -0.237065476 -4.05050006 0.007367490 0.0005512405 0.0009463645
## 18    2  26 -0.140788644 -2.08598977 0.553093684 0.0005079423 0.0005697978
## 19    2  26 -0.099218199 -1.83853581 0.686000079 0.0008756343 0.0020399374
## 20    2  26 -0.046049208 -0.85174237 0.959567857 0.0003914722 0.0011128571
## 21    2  26 -0.102633777 -1.81503721 0.697696805 0.0004063481 0.0002345858
## 22    2  26 -0.115700485 -1.72146553 0.741996511 0.0010094113 0.0047169602
## 23    2  26 -0.218251170 -2.90863990 0.159598845 0.0010530905 0.0031254005
## 24    2  26 -0.293126134 -3.65827755 0.025157448 0.0004297907 0.0012860757
## 25    2  26 -0.107476475 -2.33427946 0.414643019 0.0008718077 0.0083545213
## 26    2  26 -0.135655633 -2.06664416 0.563904448 0.0009443422 0.0007705242
## 27    2  26 -0.005168776 -0.06565125 0.995432637 0.0007059964 0.0017093982
## 28    2  26 -0.101736562 -1.02991147 0.938382427 0.0008552994 0.0003212357
## 29    2  26 -0.106917037 -1.44344289 0.848353136 0.0004659842 0.0004528307
## 30    2  26 -0.143955051 -1.60594256 0.791068910 0.0016589513 0.0022706981
## 31    2  26 -0.093688191 -1.92279670 0.642427798 0.0004025885 0.0009260538
## 32    2  26 -0.313691108 -2.30732500 0.429359754 0.0009640889 0.0019245305
## 33    2  26 -0.151599029 -2.54586869 0.305755540 0.0011446680 0.0072973098
## 34    2  26 -0.113830637 -2.06152082 0.566761534 0.0008514932 0.0055913854
## 35    2  26 -0.220363663 -1.72391205 0.740887358 0.0005138068 0.0007015982
## 36    2  26 -0.211779244 -3.98522621 0.009148144 0.0011004960 0.0062947120
## 37    2  26 -0.161851244 -2.19906397 0.489557509 0.0002334460 0.0001298656
## 38    2  26 -0.222507555 -1.45762738 0.843900682 0.0048242337 0.0019584921
## 39    2  26 -0.119405321 -2.41405422 0.372087172 0.0004737829 0.0009459741
## 40    2  26 -0.066956522 -0.94176615 0.949844273 0.0011477969 0.0044950987
## 41    2  26 -0.107420235 -2.09998836 0.545259606 0.0009669881 0.0033414294
## 42    2  26 -0.211640785 -3.51839705 0.037371037 0.0004302930 0.0020971822
## 44    2  26 -0.160491538 -2.32116399 0.421790841 0.0016866384 0.0067053791
## 45    2  26  0.013957358  0.21048073 0.998138553 0.0002474183 0.0001310960
## 46    2  26 -0.125206819 -1.36523187 0.871040437 0.0014782610 0.0005893232
## 47    2  26 -0.146576570 -2.35613125 0.402831448 0.0002851628 0.0001796349
## 48    2  26 -0.106184312 -1.41243370 0.857717409 0.0006722417 0.0029218865
```

```
## 49    2  26 -0.110328029 -2.31986075 0.422503260 0.0007413810 0.0032547907
## 50    2  26 -0.336849990 -3.07534065 0.112195130 0.0015064070 0.0017150678
## 51    2  26 -0.219041498 -2.26882562 0.450564573 0.0004175437 0.0010455238
## 53    2  26 -0.249921002 -2.67545341 0.246874228 0.0008780514 0.0016600448
## 54    2  26 -0.092856496 -1.36610183 0.870804641 0.0011141161 0.0011656778
## 55    2  26 -0.119379994 -2.19438511 0.492186948 0.0008204933 0.0007526503
## 56    2  26 -0.094196887 -1.53868461 0.816472629 0.0016308380 0.0062631397
```

Im-Pesaran-Shin (IPS) test

This test by Im, Pesaran, and Shin (2003) does not require that ρ is the same for all the individuals. The null hypothesis is still that all the series have an unit root, but the alternative is that some may have a unit root and others have different values of $\rho_i < 0$.

The test is based on the average of the student statistic of the ρ obtained for each individual:

$$\bar{t} = \frac{1}{n} \sum_{i=1}^{n} t_{\rho i}$$

The statistic is then:

$$z = \frac{\sqrt{n} \left(\bar{t} - E(\bar{t}) \right)}{\sqrt{V(\bar{t})}}$$

$\mu^*_{m\tilde{T}}$ and $\sigma^*_{m\tilde{T}}$ are given in table 2 of the original paper and are also available in the package.

An example of the IPS test with `purtest` with the same settings as in the previously performed Levin-Lin-Chu test is:

```
purtest(lprice, test = "ips", lags = 2, exo = "trend")
```

```
##
##  Im-Pesaran-Shin Unit-Root Test (ex. var.: Individual Intercepts and
##  Trend)
##
## data:  lprice
## Wtbar = 0.76622, p-value = 0.7782
## alternative hypothesis: stationarity
```

Simes' approach: intersecting hypotheses

A different approach to panel unit root testing can be drawn from the general Simes' test for intersection of individual hypothesis tests (Simes 1986). Hanck (2013) suggests to apply the approach for panel unit root testing: The tests works by combining p-values from single hypothesis tests (individual unit root tests) with a global (intersected) hypothesis and controls for the multiplicity in testing. Thus, it works "on top" of any panel unit root test which yield a p-value for each individual series. Unlike most other panel unit root tests, this approach allows to discriminate between

individuals for which the individual H0 (unit root present for individual series) is rejected/is not rejected and requires a pre-specified significance level. Further, the test is robust versus general patterns of cross-sectional dependence.

The function `phansitest` for this test takes as main input object either a numeric containing p-values of individual tests or a `"purtest"` object as produced by function `purtest` which holds a suitable pre-computed panel unit root test (one that produces p-values per individual series). The significance level is set by argument `alpha` (default 5 %). The function's return value is a list with detailed evaluation of the applied Simes test. The associated print method gives a verbal evaluation.

The following examples shows both accepted ways of input, the first example replicates Hanck (2013), table 11 (left side), who applied some panel unit root test for a Purchasing Power Parity analysis per country (individual H0 hypotheses per series) to get the individual p-values and then used Simes' approach for testing the global (intersecting) hypothesis for the whole panel.

```
### input is numeric (p-values), replicates Hanck (2013), Table 11 (left side)
pvals <- c(0.0001,0.0001,0.0001,0.0001,0.0001,0.0001,0.0050,0.0050,0.0050,
           0.0050,0.0175,0.0175,0.0200,0.0250,0.0400,0.0500,0.0575,0.2375,0.2475)
countries <- c("Argentina","Sweden","Norway","Mexico","Italy","Finland","France",
               "Germany","Belgium","U.K.","Brazil","Australia","Netherlands",
               "Portugal","Canada", "Spain","Denmark","Switzerland","Japan")
names(pvals) <- countries
h <- phansitest(pvals)
print(h)
```

```
##
##         Simes Test as Panel Unit Root Test (Hanck (2013))
##
## H0: All individual series have a unit root
## HA: Stationarity for at least some individuals
##
## Alpha: 0.05
## Number of individuals: 19
##
## Evaluation:
##  H0 rejected (globally)
##
##  Individual H0 rejected for 10 individual(s) (integer id(s)):
##   1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

```
h$rejected # logical indicating the individuals with rejected individual H0
```

```
##   Argentina      Sweden      Norway      Mexico       Italy     Finland
##        TRUE        TRUE        TRUE        TRUE        TRUE        TRUE
##      France     Germany     Belgium        U.K.      Brazil   Australia
##        TRUE        TRUE        TRUE        TRUE       FALSE       FALSE
## Netherlands    Portugal      Canada       Spain     Denmark Switzerland
##       FALSE       FALSE       FALSE       FALSE       FALSE       FALSE
##       Japan
```

```
##       FALSE
```

```
### input is a (suitable) purtest object / different example
y <- data.frame(split(Grunfeld$inv, Grunfeld$firm))
obj <- purtest(y, pmax = 4, exo = "intercept", test = "madwu")
phansitest(obj, alpha = 0.06) # test with significance level set to 6 %
```

Robust covariance matrix estimation

Robust estimators of the covariance matrix of coefficients are provided, mostly for use in Wald-type tests, and this section provides some basics and examples. A more comprehensive exposition of the theory and the capabilities that come with the plm package is given in Millo (2017).

`vcovHC` estimates three "flavours" of White's heteroskedasticity-consistent covariance matrix[22] (known as the *sandwich* estimator). Interestingly, in the context of panel data the most general version also proves consistent vs. serial correlation.

All types assume no correlation between errors of different groups while allowing for heteroskedasticity across groups, so that the full covariance matrix of errors is $V = I_n \otimes \Omega_i; i = 1, . . , n$. As for the *intragroup* error covariance matrix of every single group of observations, `"white1"` allows for general heteroskedasticity but no serial correlation, *i.e.*

$$(\#eq : omegaW1)\Omega_i = \begin{bmatrix} \sigma_{i1}^2 & \dots & \dots & 0 \\ 0 & \sigma_{i2}^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & \dots & \sigma_{iT}^2 \end{bmatrix}$$

while `"white2"` is `"white1"` restricted to a common variance inside every group, estimated as $\sigma_i^2 = \sum_{t=1}^{T} \hat{u}_{it}^2/T$, so that $\Omega_i = I_T \otimes \sigma_i^2$ (see Greene (2003), 13.7.1–2 and Wooldridge (2002), 10.7.2; `"arellano"` (see ibid. and the original ref. Manuel Arellano (1987)) allows a fully general structure w.r.t. heteroskedasticity and serial correlation:

$$(\#eq : omegaArellano)\Omega_i = \begin{bmatrix} \sigma_{i1}^2 & \sigma_{i1,i2} & \dots & \dots & \sigma_{i1,iT} \\ \sigma_{i2,i1} & \sigma_{i2}^2 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \sigma_{iT-1}^2 & \sigma_{iT-1,iT} \\ \sigma_{iT,i1} & \dots & \dots & \sigma_{iT,iT-1} & \sigma_{iT}^2 \end{bmatrix}$$

The latter is, as already observed, consistent w.r.t. timewise correlation of the errors, but on the converse, unlike the White 1 and 2 methods, it relies on large n asymptotics with small T.

The fixed effects case, as already observed in Section tests of serial correlation on serial correlation, is complicated by the fact that the demeaning induces serial correlation in the errors. The original White estimator (`"white1"`) turns out to be inconsistent for fixed T as n grows, so in this case it is advisable to use the `"arellano"` version (see Stock and Watson (2008)).

The errors may be weighted according to the schemes proposed by J. G. MacKinnon and White (1985) and Cribari–Neto (2004) to improve small-sample performance[23].

The main use of `vcovHC` (and the other variance-covariance estimators provided in the package `vcovBK`, `vcovNW`, `vcovDC`, `vcovSCC`) is to pass it to plm's own functions like `summary`, `pwaldtest`, and `phtest` or together with testing functions from the `lmtest` and `car` packages. All of these typically allow passing the `vcov` or `vcov.` parameter either as a matrix or as a function (see also Zeileis (2004)). If one is happy with the defaults, it is easiest to pass the function itself[24]:

```
re <- plm(inv~value+capital, data = Grunfeld, model = "random")
summary(re, vcov = vcovHC) # gives usual summary output but with robust test
        statistics


## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Note: Coefficient variance-covariance matrix supplied: vcovHC
##
## Call:
## plm(formula = inv ~ value + capital, data = Grunfeld, model = "random")
##
## Balanced Panel: n = 10, T = 20, N = 200
##
## Effects:
##                   var std.dev share
## idiosyncratic 2784.46   52.77 0.282
## individual    7089.80   84.20 0.718
## theta: 0.8612
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -177.6063  -19.7350    4.6851   19.5105  252.8743
##
## Coefficients:
##               Estimate Std. Error z-value              Pr(>|z|)
## (Intercept) -57.834415  23.449626 -2.4663               0.01365 *
## value         0.109781   0.012984  8.4551 < 0.00000000000000022 ***
## capital       0.308113   0.051889  5.9379        0.000000002887 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    2381400
## Residual Sum of Squares: 548900
## R-Squared:      0.7695
## Adj. R-Squared: 0.76716
## Chisq: 78.7096 on 2 DF, p-value: < 0.000000000000000222


library("lmtest")
coeftest(re, vcovHC, df = Inf)
```

```
##
## z test of coefficients:
##
##               Estimate Std. Error z value             Pr(>|z|)
## (Intercept) -57.834415  23.449626 -2.4663              0.01365 *
## value         0.109781   0.012984  8.4551 < 0.00000000000000022 ***
## capital       0.308113   0.051889  5.9379        0.000000002887 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Another approach is to compute the covariance matrix inside the call and pass it on:

```
summary(re, vcov = vcovHC(re, method="white2", type="HC3"))
coeftest(re, vcovHC(re, method="white2", type="HC3"), df = Inf)
```

For some tests, e.g., for multiple model comparisons by `waldtest`, one should always provide a function[25]. In this case, optional parameters are provided as shown below (see also Zeileis (2004), p. 12):

```
waldtest(re, update(re, . ~ . -capital),
         vcov=function(x) vcovHC(x, method="white2", type="HC3"))
```

```
## Wald test
##
## Model 1: inv ~ value + capital
## Model 2: inv ~ value
##   Res.Df Df  Chisq            Pr(>Chisq)
## 1    197
## 2    198 -1 87.828 < 0.0000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Moreover, `linearHypothesis` from package `car` may be used to test for linear restrictions:

```
library("car")
linearHypothesis(re, "2*value=capital", vcov. = vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## 2 value - capital = 0
##
## Model 1: restricted model
## Model 2: inv ~ value + capital
##
```

```
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df  Chisq Pr(>Chisq)
## 1    198
## 2    197  1 3.4783    0.06218 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A specific methods are also provided for `pcce` and `pgmm` objects, for the latter `vcovHC` provides the robust covariance matrix proposed by Windmeijer (2005) for generalized method of moments estimators.

plm versus nlme and lme4

The models termed *panel* by the econometricians have counterparts in the statistics literature on *mixed* models (or *hierarchical models*, or *models for longitudinal data*), although there are both differences in jargon and more substantial distinctions. This language inconsistency between the two communities, together with the more complicated general structure of statistical models for longitudinal data and the associated notation in the software, is likely to scare some practicing econometricians away from some potentially useful features of the `R` environment, so it may be useful to provide here a brief reconciliation between the typical panel data specifications used in econometrics and the general framework used in statistics for mixed models[26].

`R` is particularly strong on mixed models' estimation, thanks to the long-standing `nlme` package (see J. Pinheiro et al. (2007)) and the more recent `lme4` package, based on S4 classes (see Bates (2007))[27]. In the following we will refer to the more established `nlme` to give some examples of "econometric" panel models that can be estimated in a likelihood framework, also including some likelihood ratio tests. Some of them are not feasible in `plm` and make a useful complement to the econometric "toolbox" available in `R`.

Fundamental differences between the two approaches

Econometrics deal mostly with non-experimental data. Great emphasis is put on specification procedures and misspecification testing. Model specifications tend therefore to be very simple, while great attention is put on the issues of endogeneity of the regressors, dependence structures in the errors and robustness of the estimators under deviations from normality. The preferred approach is often semi- or non-parametric, and heteroskedasticity-consistent techniques are becoming standard practice both in estimation and testing.

For all these reasons, although the maximum likelihood framework is important in testing[28] and sometimes used in estimation as well, panel model estimation in econometrics is mostly accomplished in the generalized least squares framework based on Aitken's Theorem and, when possible, in its special case OLS, which are free from distributional assumptions (although these kick in at the diagnostic testing stage). On the contrary, longitudinal data models in `nlme` and `lme4` are estimated by (restricted or unrestricted) maximum likelihood. While under normality, homoskedasticity and no serial correlation of the errors OLS are also the maximum likelihood estimator, in all the other cases there are important differences.

The econometric GLS approach has closed-form analytical solutions computable by standard linear

algebra and, although the latter can sometimes get computationally heavy on the machine, the expressions for the estimators are usually rather simple. ML estimation of longitudinal models, on the contrary, is based on numerical optimization of nonlinear functions without closed-form solutions and is thus dependent on approximations and convergence criteria. For example, the "GLS" functionality in `nlme` is rather different from its "econometric" counterpart. "Feasible GLS" estimation in `plm` is based on a single two-step procedure, in which an inefficient but consistent estimation method (typically OLS) is employed first in order to get a consistent estimate of the errors' covariance matrix, to be used in GLS at the second step; on the converse, "GLS" estimators in `nlme` are based on iteration until convergence of two-step optimization of the relevant likelihood.

Some false friends

The *fixed/random effects* terminology in econometrics is often recognized to be misleading, as both are treated as random variates in modern econometrics (see, e.g., Wooldridge (2002) 10.2.1). It has been recognized since Mundlak's classic paper (Mundlak (1978)) that the fundamental issue is whether the unobserved effects are correlated with the regressors or not. In this last case, they can safely be left in the error term, and the serial correlation they induce is cared for by means of appropriate GLS transformations. On the contrary, in the case of correlation, "fixed effects" methods such as least squares dummy variables or time-demeaning are needed, which explicitly, although inconsistently[29], estimate a group– (or time–) invariant additional parameter for each group (or time period).

Thus, from the point of view of model specification, having *fixed effects* in an econometric model has the meaning of allowing the intercept to vary with group, or time, or both, while the other parameters are generally still assumed to be homogeneous. Having *random effects* means having a group– (or time–, or both) specific component in the error term.

In the mixed models literature, on the contrary, *fixed effect* indicates a parameter that is assumed constant, while *random effects* are parameters that vary randomly around zero according to a joint multivariate normal distribution.

So, the FE model in econometrics has no counterpart in the mixed models framework, unless reducing it to OLS on a specification with one dummy for each group (often termed *least squares dummy variables*, or LSDV model) which can trivially be estimated by OLS. The RE model is instead a special case of a mixed model where only the intercept is specified as a random effect, while the "random" type variable coefficients model can be seen as one that has the same regressors in the fixed and random sets. The unrestricted generalized least squares can in turn be seen, in the `nlme` framework, as a standard linear model with a general error covariance structure within the groups and errors uncorrelated across groups.

A common taxonomy

To reconcile the two terminologies, in the following we report the specification of the panel models in `plm` according to the general expression of a mixed model in Laird-Ware form (see the web appendix to Fox 2002) and the `nlme` estimation commands for maximum likelihood estimation of an equivalent specification[30].

The Laird-Ware representation for mixed models

A general representation for the linear mixed effects model is given in Laird and Ware (1982).

$$
\begin{aligned}
y_{it} &= \beta_1 x_{1ij} + \cdots + \beta_p x_{pij} \\
&\quad b_1 z_{1ij} + \cdots + b_p z_{pij} + \epsilon_{ij} \\
b_{ik} &\sim N(0, \psi_k^2), \quad Cov(b_k, b_{k'}) = \psi_{kk'} \\
\epsilon_{ij} &\sim N(0, \sigma^2 \lambda_{ijj}), \quad Cov(\epsilon_{ij}, \epsilon_{ij'}) = \sigma^2 \lambda_{ijj'}
\end{aligned}
$$

where the $x_1, \dots x_p$ are the fixed effects regressors and the $z_1, \dots z_p$ are the random effects regressors, assumed to be normally distributed across groups. The covariance of the random effects coefficients $\psi_{kk'}$ is assumed constant across groups and the covariances between the errors in group i, $\sigma^2 \lambda_{ijj'}$, are described by the term $\lambda_{ijj'}$ representing the correlation structure of the errors within each group (e.g., serial correlation over time) scaled by the common error variance σ^2.

Pooling and Within

The *pooling* specification in `plm` is equivalent to a classical linear model (i.e., no random effects regressor and spherical errors: $b_{iq} = 0 \ \ \forall i, q, \ \ \lambda_{ijj} = \sigma^2$ for $j = j'$, 0 else). The *within* one is the same with the regressors' set augmented by $n - 1$ group dummies. There is no point in using `nlme` as parameters can be estimated by OLS which is also ML.

Random effects

In the Laird and Ware notation, the RE specification is a model with only one random effects regressor: the intercept. Formally, $z_{1ij} = 1 \ \ \forall i, j, \ \ z_{qij} = 0 \ \ \forall i, \forall j, \forall q \neq 1 \ \lambda_{ij} = 1$ for $i = j, 0$ else). The composite error is therefore $u_{ij} = 1 b_{i1} + \epsilon_{ij}$. Below we report coefficients of Grunfeld's model estimated by GLS and then by ML:

```
library(nlme)
reGLS <- plm(inv~value+capital, data=Grunfeld, model="random")

reML <- lme(inv~value+capital, data=Grunfeld, random=~1|firm)

coef(reGLS)
```

```
## (Intercept)       value     capital
## -57.8344149   0.1097812   0.3081130
```

```
summary(reML)$coefficients$fixed
```

```
## (Intercept)       value     capital
## -57.8644245   0.1097897   0.3081881
```

Variable coefficients, "random"

Swamy's variable coefficients model (Swamy 1970) has coefficients varying randomly (and independently of each other) around a set of fixed values, so the equivalent specification is $z_q = x_q \ \ \forall q$, i.e. the fixed effects and the random effects regressors are the same, and $\psi_{kk'} = \sigma_\mu^2 I_N$, and $\lambda_{ijj} = 1$, $\lambda_{ijj'} = 0$ for $j \neq j'$, that's to say they are not correlated.

Estimation of a mixed model with random coefficients on all regressors is rather demanding from the computational side. Some models from our examples fail to converge. The below example is estimated on the Grunfeld data and model with time effects.

```
vcm <- pvcm(inv~value+capital, data=Grunfeld, model="random", effect="time")

vcmML <- lme(inv~value+capital, data=Grunfeld, random=~value+capital|year)

coef(vcm)
```

```
## (Intercept)       value     capital
## -18.5538638   0.1239595   0.1114579
```

```
summary(vcmML)$coefficients$fixed
```

```
## (Intercept)       value     capital
## -26.3558395   0.1241982   0.1381782
```

Variable coefficients, "within"

This specification actually entails separate estimation of T different standard linear models, one for each group in the data, so the estimation approach is the same: OLS. In `nlme` this is done by creating an `lmList` object, so that the two models below are equivalent (output suppressed):

```
vcmf <- pvcm(inv~value+capital, data=Grunfeld, model="within", effect="time")

vcmfML <- lmList(inv~value+capital|year, data=Grunfeld)
```

General FGLS

The general, or unrestricted, feasible GLS (FGLS), `pggls` in the `plm` nomenclature, is equivalent to a model with no random effects regressors ($b_{iq} = 0 \ \ \forall i, q$) and an error covariance structure which is unrestricted within groups apart from the usual requirements. The function for estimating such models with correlation in the errors but no random effects is `gls()`.

This very general serial correlation and heteroskedasticity structure is not estimable for the original Grunfeld data, which have more time periods than firms, therefore we restrict them to firms 4 to 6.

```
sGrunfeld <- Grunfeld[Grunfeld$firm %in% 4:6, ]
```

```
ggls <- pggls(inv~value+capital, data=sGrunfeld, model="pooling")

gglsML <- gls(inv~value+capital, data=sGrunfeld,
              correlation=corSymm(form=~1|year))

coef(ggls)
```

```
## (Intercept)       value     capital
##  1.19679342  0.10555908  0.06600166
```

```
summary(gglsML)$coefficients
```

```
## (Intercept)       value     capital
##  -2.4156266   0.1163550   0.0735837
```

The *within* case is analogous, with the regressor set augmented by $n-1$ group dummies.

Some useful “econometric” models in nlme

Finally, amongst the many possible specifications estimable with `nlme`, we report a couple cases that might be especially interesting to applied econometricians.

AR(1) pooling or random effects panel

Linear models with groupwise structures of time-dependence[31] may be fitted by `gls()`, specifying the correlation structure in the `correlation` option[32]:

```
Grunfeld$year <- as.numeric(as.character(Grunfeld$year))
lmAR1ML <- gls(inv~value+capital,data=Grunfeld,
               correlation=corAR1(0,form=~year|firm))
```

and analogously the random effects panel with, e.g., AR(1) errors (see B. H. Baltagi (2005); B. H. Baltagi (2013); B. H. Baltagi (2021), ch. 5), which is a very common specification in econometrics, may be fit by `lme` specifying an additional random intercept:

```
reAR1ML <- lme(inv~value+capital, data=Grunfeld,random=~1|firm,
               correlation=corAR1(0,form=~year|firm))
```

The regressors’ coefficients and the error’s serial correlation coefficient may be retrieved this way:

```
summary(reAR1ML)$coefficients$fixed
```

```
##  (Intercept)       value     capital
```

```
## -40.27650822   0.09336672   0.31323330
```

```
coef(reAR1ML$modelStruct$corStruct, unconstrained=FALSE)
```

```
##      Phi
## 0.823845
```

Significance statistics for the regressors' coefficients are to be found in the usual `summary` object, while to get the significance test of the serial correlation coefficient one can do a likelihood ratio test as shown in the following.

An LR test for serial correlation and one for random effects

A likelihood ratio test for serial correlation in the idiosyncratic residuals can be done as a nested models test, by `anova()`, comparing the model with spherical idiosyncratic residuals with the more general alternative featuring AR(1) residuals. The test takes the form of a zero restriction test on the autoregressive parameter.

This can be done on pooled or random effects models alike. First we report the simpler case.

We already estimated the pooling AR(1) model above. The GLS model without correlation in the residuals is the same as OLS, and one could well use `lm()` for the restricted model. Here we estimate it by `gls()`.

```
lmML <- gls(inv~value+capital, data=Grunfeld)
anova(lmML, lmAR1ML)
```

```
##         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## lmML        1  4 2400.217 2413.350 -1196.109
## lmAR1ML     2  5 2094.936 2111.352 -1042.468 1 vs 2 307.2813  <.0001
```

The AR(1) test on the random effects model is to be done in much the same way, using the random effects model objects estimated above:

```
anova(reML, reAR1ML)
```

```
##         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## reML        1  5 2205.851 2222.267 -1097.926
## reAR1ML     2  6 2094.802 2114.501 -1041.401 1 vs 2 113.0496  <.0001
```

A likelihood ratio test for random effects compares the specifications with and without random effects and spherical idiosyncratic errors:

```
anova(lmML, reML)
```

```
##      Model df      AIC      BIC    logLik   Test L.Ratio p-value
## lmML     1  4 2400.217 2413.350 -1196.109
## reML     2  5 2205.851 2222.267 -1097.926 1 vs 2 196.366  <.0001
```

The random effects, AR(1) errors model in turn nests the AR(1) pooling model, therefore a likelihood ratio test for random effects sub AR(1) errors may be carried out, again, by comparing the two autoregressive specifications:

```
anova(lmAR1ML, reAR1ML)
```

```
##         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## lmAR1ML     1  5 2094.936 2111.352 -1042.468
## reAR1ML     2  6 2094.802 2114.501 -1041.401 1 vs 2 2.134349   0.144
```

whence we see that the Grunfeld model specification doesn't seem to need any random effects once we control for serial correlation in the data.

Conclusions

With `plm` we aim at providing a comprehensive package containing the standard functionalities that are needed for the management and the econometric analysis of panel data. In particular, we provide: functions for data transformation; estimators for pooled, random and fixed effects static panel models and variable coefficients models, general GLS for general covariance structures, and generalized method of moments estimators for dynamic panels; specification and diagnostic tests. Instrumental variables estimation is supported. Most estimators allow working with unbalanced panels. While among the different approaches to longitudinal data analysis we take the perspective of the econometrician, the syntax is consistent with the basic linear modeling tools, like the `lm` function.

On the input side, `formula` and `data` arguments are used to specify the model to be estimated. Special functions are provided to make writing formulas easier, and the structure of the data is indicated with an `index` argument.

On the output side, the model objects (of the new class `panelmodel`) are compatible with the general restriction testing frameworks of packages `lmtest` and `car`. Specialized methods are also provided for the calculation of robust covariance matrices; heteroskedasticity- and correlation-consistent testing is accomplished by passing these on to testing functions, together with a `panelmodel` object.

The main functionalities of the package have been illustrated here by applying them on some well-known data sets from the econometric literature. The similarities and differences with the maximum likelihood approach to longitudinal data have also been briefly discussed.

Acknowledgments

While retaining responsibility for any error, we thank Jeffrey Wooldridge, Achim Zeileis and three anonymous referees for useful comments. We also acknowledge kind editing assistance by Lisa Benedetti.

Bibliography

Ahrens, H., and R. Pincus. 1981. “On Two Measures of Unbalancedness in a One-Way Model and Their Relation to Efficiency.” *Biometrical Journal* 23 (3): 227–35. https://doi.org/10.1002/bimj.4710230302.

Amemiya, T. 1971. “The Estimation of the Variances in a Variance–Components Model.” *International Economic Review* 12: 1–13.

Amemiya, Takeshi, and Thomas E MaCurdy. 1986. “Instrumental-Variable Estimation of an Error-Components Model.” *Econometrica* 54 (4): 869–80.

Anderson, T. W., and C. Hsiao. 1981. “Estimation of Dynamic Models with Error Components.” *Journal of the American Statistical Association* 76: 598–606.

Arellano, Manuel. 1987. “Computing Robust Standard Errors for Within-Groups Estimators.” *Oxford Bulletin of Economics and Statistics* 49 (4): 431–34.

Arellano, M., and S. Bond. 1991. “Some Tests of Specification for Panel Data : Monte Carlo Evidence and an Application to Employment Equations.” *Review of Economic Studies* 58: 277–97.

Balestra, P., and J. Varadharajan–Krishnakumar. 1987. “Full Information Estimations of a System of Simultaneous Equations with Error Components.” *Econometric Theory* 3: 223–46.

Baltagi, B. H. 1981. “Simultaneous Equations with Error Components.” *Journal of Econometrics* 17: 21–49.

Baltagi, B. H. 2005. *Econometric Analysis of Panel Data*. 3rd ed. John Wiley; Sons ltd.

———. 2013. *Econometric Analysis of Panel Data*. 5th ed. John Wiley; Sons ltd.

———. 2021. *Econometric Analysis of Panel Data*. 6th ed. Springer.

Baltagi, B. H., and Y. J. Chang. 1994. “Incomplete Panels: A Comparative Study of Alternative Estimators for the Unbalanced One-Way Error Component Regression Model.” *Journal of Econometrics* 62: 67–89.

Baltagi, B. H., Y. J. Chang, and Q. Li. 1992. “Monte Carlo Results on Several New and Existing Tests for the Error Components Model.” *Journal of Econometrics* 54: 95–120.

Baltagi, B. H., and Q. Li. 1990. “A Lagrange Multiplier Test for the Error Components Model with Incomplete Panels.” *Econometric Reviews* 9: 103–7.

Baltagi, Badi H., Qu Feng, and Chihwa Kao. 2012. “A Lagrange Multiplier Test for Cross-Sectional Dependence in a Fixed Effects Panel Data Model.” *Journal of Econometrics* 170 (1): 164–77. https://www.sciencedirect.com/science/article/pii/S030440761200098X.

Baltagi, Badi H., and Ping X. Wu. 1999. “Unequally Spaced Panel Data Regressions with AR(1) Disturbances.” *Econometric Theory* 15 (6): 814–23.

Baltagi, Badi, YA Chang, and Q Li. 1998. “Testing for Random Individual and Time Effects Using Unbalanced Panel Data.” *Advances in Econometrics* 13 (January): 1–20.

Baltagi, B., and Q. Li. 1991. “A Joint Test for Serial Correlation and Random Individual Effects.” *Statistics and Probability Letters* 11: 277–80.

———. 1995. “Testing AR(1) Against MA(1) Disturbances in an Error Component Model.” *Journal of Econometrics* 68: 133–51.

Bates, Douglas. 2004. “Least Squares Calculations in .” *–News* 4 (1): 17–20.

———. 2007. *: Linear Mixed–Effects Models Using Classes*. https://CRAN.r-project.org/package=lme4.

Bates, Douglas, and Martin Maechler. 2016. *: Sparse and Dense Matrix Classes and Methods*. https://CRAN.R-project.org/package=Matrix.

Bera, A. K., W. Sosa–Escudero, and M. Yoon. 2001. “Tests for the Error Component Model in the Presence of Local Misspecification.” *Journal of Econometrics* 101: 1–23.

Bhargava, A., L. Franzini, and W. Narendranathan. 1982. “Serial Correlation and the Fixed Effects Model.” *The Review of Economic Studies* 49 (4): 533–49.

Bivand, Roger. 2008. *Spdep: Spatial Dependence: Weighting Schemes, Statistics and Models*.

Blundell, R., and S. Bond. 1998. “Initital Conditions and Moment Restrictions in Dynamic Panel Data Models.” *Journal of Econometrics* 87: 115–43.

Breusch, T. S., and A. R. Pagan. 1980. “The Lagrange Multiplier Test and Its Applications to Model

Specification in Econometrics." *Review of Economic Studies* 47: 239–53.
Breusch, Trevor S, Grayham E Mizon, and Peter Schmidt. 1989. "Efficient Estimation Using Panel Data." *Econometrica* 57 (3): 695–700.
Choi, In. 2001. "Unit Root Tests for Panel Data." *Journal of International Money and Finance* 20 (2): 249–72. https://www.sciencedirect.com/science/article/pii/S0261560600000486.
Cornwell, C., and P. Rupert. 1988. "Efficient Estimation with Panel Data: An Empirical Comparison of Instrumental Variables Estimators." *Journal of Applied Econometrics* 3: 149–55.
Cribari–Neto, F. 2004. "Asymptotic Inference Under Heteroskedasticity of Unknown Form." *Computational Statistics & Data Analysis* 45: 215–33.
Croissant, Yves, and Giovanni Millo. 2008. "Panel Data Econometrics in : The Package." *Journal of Statistical Software* 27 (2): 1–43. https://www.jstatsoft.org/article/view/v027i02.
De Hoyos, R. E., and V. Sarafidis. 2006. "Testing for Cross–Sectional Dependence in Panel–Data Models." *The Stata Journal* 6 (4): 482–96.
Development Core Team. 2008. *: A Language and Environment for Statistical Computing*. Vienna, Austria: Foundation for Statistical Computing. https://www.r-project.org/.
Drukker, D. M. 2003. "Testing for Serial Correlation in Linear Panel–Data Models." *The Stata Journal* 3 (2): 168–77.
Fox, John. 2002. *An and Companion to Applied Regression*. Sage.
———. 2016. *: Companion to Applied Regression*. https://CRAN.R-project.org/package=car.
Gourieroux, C., A. Holly, and A. Monfort. 1982. "Likelihood Ratio Test, Wald Test, and Kuhn–Tucker Test in Linear Models with Inequality Constraints on the Regression Parameters." *Econometrica* 50: 63–80.
Greene, W. H. 2003. *Econometric Analysis*. 5th ed. Prentice Hall.
Hadri, Kaddour. 2000. "Testing for Stationarity in Heterogeneous Panel Data." *The Econometrics Journal* 3 (2): 148–61.
Hanck, Christoph. 2013. "An Intersection Test for Panel Unit Roots." *Econometric Reviews* 32: 183–203.
Harrison, D., and D. L. Rubinfeld. 1978. "Hedonic Housing Prices and the Demand for Clean Air." *Journal of Environmental Economics and Management* 5: 81–102.
Hausman, J. A. 1978. "Specification Tests in Econometrics." *Econometrica* 46: 1251–71.
Hausman, J. A., and W. E. Taylor. 1981. "Panel Data and Unobservable Individual Effects." *Econometrica* 49: 1377–98.
Holtz–Eakin, D., W. Newey, and H. S. Rosen. 1988. "Estimating Vector Autoregressions with Panel Data." *Econometrica* 56: 1371–95.
Honda, Y. 1985. "Testing the Error Components Model with Non–Normal Disturbances." *Review of Economic Studies* 52: 681–90.
Hothorn, T., A. Zeileis, R. W. Farebrother, C. Cummins, G. Millo, and D. Mitchell. 2015. *: Testing Linear Regression Models*. https://CRAN.R-project.org/package=lmtest.
Im, K. S., M. H. Pesaran, and Y. Shin. 2003. "Testing for Unit Roots in Heterogenous Panels." *Journal of Econometrics* 115(1): 53–74.
King, M. L., and P. X. Wu. 1997. "Locally Optimal One–Sided Tests for Multiparameter Hypothese." *Econometric Reviews* 33: 523–29.
Kleiber, Christian, and Achim Zeileis. 2008. *Applied Econometrics with R*. New York: Springer-Verlag. https://CRAN.R-project.org/package=AER.
Koenker, Roger, and Pin Ng. 2016. *: Sparse Linear Algebra*. https://CRAN.R-project.org/package=SparseM.
Kwiatkowski, Denis, Peter C. B. Phillips, Peter Schmidt, and Yongcheol Shin. 1992. "Testing the Null Hypothesis of Stationarity Against the Alternative of a Unit Root: How Sure Are We That Economic Time Series Have a Unit Root?" *Journal of Econometrics* 54 (1): 159–78. https://www.sciencedirect.com/science/article/pii/030440769290104Y.

Laird, N. M., and J. H. Ware. 1982. “Random–Effects Models for Longitudinal Data.” *Biometrics* 38: 963–74.
Levin, A., C. F. Lin, and C. S. J. Chu. 2002. “Unit Root Tests in Panel Data : Asymptotic and Finite-Sample Properties.” *Journal of Econometrics* 108: 1–24.
Lumley, T., and A. Zeileis. 2015. *: Robust Covariance Matrix Estimators*. https://CRAN.R-project.org/package=sandwich.
MacKinnon, J. G., and H. White. 1985. “Some Heteroskedasticity–Consistent Covariance Matrix Estimators with Improved Finite Sample Properties.” *Journal of Econometrics* 29: 305–25.
MacKinnon, James G. 1994. “Approximate Asymptotic Distribution Functions for Unit-Root and Cointegration Tests.” *Journal of Business & Economic Statistics* 12 (2): 167–76.
———. 1996. “Numerical Distribution Functions for Unit Root and Cointegration Tests.” *Journal of Applied Econometrics* 11 (6): 601–18.
Maddala, G. S., and S. Wu. 1999. “A Comparative Study of Unit Root Tests with Panel Data and a New Simple Test.” *Oxford Bulletin of Economics and Statistics* 61: 631–52.
Millo, G. 2017. “Robust Standard Error Estimators for Panel Models: A Unifying Approach.” *Journal of Statistical Software* 82 (3): 1–27.
Mundlak, Yair. 1978. “On the Pooling of Time Series and Cross Section Data.” *Econometrica* 46 (1): 69–85.
Munnell, A. 1990. “Why Has Productivity Growth Declined? Productivity and Public Investment.” *New England Economic Review*, 3–22.
Nerlove, M. 1971. “Further Evidence on the Estimation of Dynamic Economic Relations from a Time–Series of Cross–Sections.” *Econometrica* 39: 359–82.
Pesaran, M Hashem. 2007. “A Simple Panel Unit Root Test in the Presence of Cross-Section Dependence.” *Journal of Applied Econometrics* 22 (2): 265–312.
Pesaran, M. H. 2004. “General Diagnostic Tests for Cross Section Dependence in Panels.”
Pesaran, M. Hashem. 2015. “Testing Weak Cross-Sectional Dependence in Large Panels.” *Econometric Reviews* 34 (6-10): 1089–1117. https://doi.org/10.1080/07474938.2014.956623.
Pfaff, Bernhard. 2008. *Analysis of Integrated and Cointegrated Time Series with r*. Second. New York: Springer. https://CRAN.r-project.org/package=urca.
Pinheiro, J. C., and D. Bates. 2000. *Mixed–Effects Models in and* . Springer-Verlag.
Pinheiro, Jose, Douglas Bates, Saikat DebRoy, and Deepayan Sarkar the Core team. 2007. *: Linear and Nonlinear Mixed Effects Models*. https://CRAN.r-project.org/package=nlme.
Simes, R. J. 1986. “An Improved Bonferroni Procedure for Multiple Tests of Significance.” *Biometrika* 73: 751–54.
Stock, James H., and Mark W. Watson. 2008. “Heteroskedasticity–Robust Standard Errors for Fixed Effects Panel Data Regression.” *Econometrica* 76 (1): 155–74.
Swamy, P. A. V. B. 1970. “Efficient Inference in a Random Coefficient Regression Model.” *Econometrica* 38: 311–23.
Swamy, P. A. V. B., and S. S Arora. 1972. “The Exact Finite Sample Properties of the Estimators of Coefficients in the Error Components Regression Models.” *Econometrica* 40: 261–75.
Therneau, Terry. 2014. *: Routines for Block Diagonal Symmetric Matrices*. https://CRAN.R-project.org/package=bdsmatrix.
Wallace, T. D., and A. Hussain. 1969. “The Use of Error Components Models in Combining Cross Section with Time Series Data.” *Econometrica* 37 (1): 55–72.
White, H. 1984. *Asymptotic Theory for Econometricians*. New York: Academic press.
White, Halbert. 1980. “A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity.” *Econometrica* 48 (4): 817–38.
Windmeijer, F. 2005. “A Finite Sample Correction for the Variance of Linear Efficient Two–Steps GMM Estimators.” *Journal of Econometrics* 126: 25–51.
Wooldridge, J. M. 2002. *Econometric Analysis of Cross–Section and Panel Data*. MIT Press.

———. 2010. *Econometric Analysis of Cross–Section and Panel Data*. 2nd ed. MIT Press.
Zeileis, A. 2004. “Econometric Computing with HC and HAC Covariance Matrix Estimators.” *Journal of Statistical Software* 11 (10): 1–17. https://www.jstatsoft.org/article/view/v011i10.

---

1. Comprehensive treatments are to be found in many econometrics textbooks, e.g., B. H. Baltagi (2005), B. H. Baltagi (2013), B. H. Baltagi (2021) or Wooldridge (2002), Wooldridge (2010): the reader is referred to these, especially to the first 9 chapters of B. H. Baltagi (2005), B. H. Baltagi (2013), B. H. Baltagi (2021).↩
2. For the sake of exposition we are considering only the individual effects case here. There may also be time effects, which is a symmetric case, or both of them, so that the error has three components: $u_{it} = \mu_i + \lambda_t + \epsilon_{it}$.↩
3. Although in most models the individual and time effects cases are symmetric, there are exceptions: estimating the *first-difference* model on time effects is meaningless because cross-sections do not generally have a natural ordering, so trying `effect = "time"` stops with an error message as does `effect = "twoways"` which is not defined for first-difference models.↩
4. See packages `lmtest` (Hothorn et al. (2015)) and `car` (Fox (2016)).↩
5. Moreover, `coeftest()` provides a compact way of looking at coefficient estimates and significance diagnostics.↩
6. Function `pht` is a deprecated way to estimate this type of model: `ht <- pht(lwage~wks+south+smsa+married+exp+I(exp^2)+ bluecol+ind+union+sex+black+ed | sex+black+bluecol+south+smsa+ind, data=Wages,index=595)`.↩
7. The “random effect” is better termed “general FGLS” model, as in fact it does not have a proper random effects structure, but we keep this terminology for general language consistency.↩
8. NB: Oneway King-Wu (`"kw"`) statistics (`"individual"` and `"time"`) coincide with the respective Honda statistics (`"honda"`); however, the twoway statistics of `"kw"` and `"honda"` differ.↩
9. The `"bp"` test for unbalanced panels was derived in B. H. Baltagi and Li (1990), the `"kw"` test for unbalanced panels in Badi Baltagi, Chang, and Li (1998). The `"ghm"` test and the `"kw"` test were extended to two–way effects in B. H. Baltagi, Chang, and Li (1992). For a concise overview of all these statistics see B. H. Baltagi (2013) Sec. 4.2, pp. 68–76 (for balanced panels) and Sec. 9.5, pp. 200–203 (for unbalanced panels) or B. H. Baltagi (2021), Sec. 4.2, pp. 81-84 (balanced), Sec. 9.6, pp. 243-246 (unbalanced).↩
10. Here we treat fixed and random effects alike, as components of the error term, according with the modern approach in econometrics (see Wooldridge (2002), Wooldridge (2010)).↩
11. Neglecting time effects may also lead to serial correlation in residuals (as observed in Wooldridge (2002) 10.4.1).↩
12. LM_3 in B. Baltagi and Li (1995).↩
13. Corresponding to RSO^*_μ in the original paper.↩
14. Baltagi and Li derive a basically analogous T-asymptotic test for first-order serial correlation in a FE panel model as a Breusch-Godfrey LM test on within residuals (see B. Baltagi and Li (1995) par. 2.3 and formula 12). They also observe that the test on within residuals can be used for testing on the RE model, as “the within transformation [time-demeaning, in our terminology] wipes out the individual effects, whether fixed or random”. Generalizing the Durbin-Watson test to FE models by applying it to fixed effects residuals is documented in Bhargava, Franzini, and

Narendranathan (1982), a (modified) version for unbalanced and/or non-consecutive panels is implemented in `pbnftest` as is Baltagi-Wu's LBI statistic (for both see Badi H. Baltagi and Wu (1999)).↩

15. see subsection robust covariance matrix estimation.↩
16. Here, e_{it} for notational simplicity (and as in Wooldridge): equivalent to $\Delta\epsilon_{it}$ in the general notation of the paper.↩
17. This is the case, e.g., if in an unobserved effects model when XSD is due to an unobservable factor structure, with factors that are uncorrelated with the regressors. In this case the within or random estimators are still consistent, although inefficient (see De Hoyos and Sarafidis (2006)).↩
18. The unbalanced version of this statistic uses max(Tij) for T in the bias-correction term.↩
19. This is also the only solution when the time dimension's length is insufficient for estimating the heterogeneous model.↩
20. The very comprehensive package `spdep` for spatial dependence analysis (see Bivand (2008)) contains features for creating, lagging and manipulating *neighbour list* objects of class `nb`, that can be readily converted to and from proximity matrices by means of the `nb2mat` function. Higher orders of the $CD(p)$ test can be obtained by lagging the corresponding `nb`s through `nblag`.↩
21. The individual p-values for the Fisher-type tests are approximated as described in James G. MacKinnon (1996) if the package `urca` (Pfaff (2008)) is available, otherwise as described in James G. MacKinnon (1994).↩
22. See Halbert White (1980) and H. White (1984).↩
23. The HC3 and HC4 weighting schemes are computationally expensive and may hit memory limits for nT in the thousands, where on the other hand it makes little sense to apply small sample corrections.↩
24. For `coeftest` set `df = Inf` to have the coefficients' tests be performed with standard normal distribution instead of t distribution as we deal with a random effects model here. For these types of models, the precise distribution of the coefficients estimates is unknown.↩
25. Joint zero-restriction testing still allows providing the `vcov` of the unrestricted model as a matrix, see the documentation of package `lmtest`.↩
26. This discussion does not consider GMM models. One of the basic reasons for econometricians not to choose maximum likelihood methods in estimation is that the strict exogeneity of regressors assumption required for consistency of the ML models reported in the following is often inappropriate in economic settings.↩
27. The standard reference on the subject of mixed models in `S/R` is J. C. Pinheiro and Bates (2000).↩
28. Lagrange Multiplier tests based on the likelihood principle are suitable for testing against more general alternatives on the basis of a maintained model with spherical residuals and find therefore application in testing for departures from the classical hypotheses on the error term. The seminal reference is T. S. Breusch and Pagan (1980).↩
29. For fixed effects estimation, as the sample grows (on the dimension on which the fixed effects are specified) so does the number of parameters to be estimated. Estimation of individual fixed effects is T– (but not n–) consistent, and the opposite.↩
30. In doing so, we stress that "equivalence" concerns only the specification of the model, and

neither the appropriateness nor the relative efficiency of the relevant estimation techniques, which will of course be dependent on the context. Unlike their mixed model counterparts, the specifications in `plm` are, strictly speaking, distribution-free. Nevertheless, for the sake of exposition, in the following we present them in the setting which ensures consistency and efficiency (e.g., we consider the hypothesis of spherical errors part of the specification of pooled OLS and so forth).↩

31. Take heed that here, in contrast to the usual meaning of serial correlation in time series, we always speak of serial correlation *between the errors of each group*.↩
32. note that the time index is coerced to numeric before the estimation.↩