Panel Data Econometrics in R: The plm Package

Yves Croissant
Université Lumière Lyon 2

Giovanni Millo
University of Trieste and Generali SpA

Abstract

This introduction to the **plm** package is a slightly modified version of **?**, published in the Journal of Statistical Software.

Panel data econometrics is obviously one of the main fields in the profession, but most of the models used are difficult to estimate with 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.

Keywords: panel data, covariance matrix estimators, generalized method of moments, R.

1. 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 statistic (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** (**?**) and **lme4** (**?**), is available in the R (**?**) 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 datasets 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 themselves 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 the need for automation of some basic data management

tasks 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.

The paper is organized as follows: Section 2 presents a very short overview of the typical model taxonomy[1]. Section 3 discusses the software approach used in the package. The next three sections present the functionalities of the package in more detail: data management (Section 4), estimation (Section 5) and testing (Section 6), giving a short description and illustrating them with examples. Section 7 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 8 concludes the paper.

2. The linear panel model

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

$$y_{it} = \alpha_{it} + \beta_{it}^{\top} x_{it} + u_{it} \tag{1}$$

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

Of course the latter 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

$$y_{it} = \alpha + \beta^{\top} x_{it} + u_{it} \tag{2}$$

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:

$$y_{it} = \alpha + \beta^{\top} x_{it} + \mu_i + \epsilon_{it} \tag{3}$$

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 from both the regressors x_{it} and the individual error component μ_i. The individual component may be in turn either independent from the regressors or correlated.

If it is correlated, the ordinary least squares (OLS) estimator for 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

[1] Comprehensive treatments are to be found in many econometrics textbooks, e.g. **?** or **?**: the reader is referred to these, especially to the first 9 chapters of **?**.

[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: $u_{it} = \mu_i + \lambda_t + \epsilon_{it}$.

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

$$\Delta y_{it} = \beta^\top \Delta x_{it} + \Delta u_{it} \tag{4}$$

(where $\Delta y_{it} = y_{it} - y_{i,t-1}$, $\Delta x_{it} = x_{it} - x_{i,t-1}$ and, from (3), $\Delta u_{it} = u_{it} - u_{i,t-1} = \Delta \epsilon_{it}$ for $t = 2, ..., T$) 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.

3. Software approach

3.1. 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 is an ordinary `data.frame`, but 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 `plm.data` function is then called, which returns a `data.frame` with the two first columns containing the individual and the time indexes.

3.2. Interface

Estimation interface

plm provides four functions for estimation:

- `plm`: estimation of the basic panel models, *i.e.* within, between and random effect models. Models are estimated using the `lm` function to 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.

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.

3.3. 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

$$\hat{\beta} = (X^\top \hat{V}^{-1} X)^{-1} (X^\top \hat{V}^{-1} y) \tag{5}$$

[3]Although in most models the individual and time effects cases are symmetric, there are exceptions: estimating the `fd` model on time effects is meaningless because cross-sections do not generally have a natural ordering, so here the `effect` will always be set to `"individual"`.

Nevertheless, in practice plain computation of $\hat{\beta}$ has long been an intractable problem even for moderate-sized datasets 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 (5) 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:

$$y_{it} - \theta\bar{y}_i = (X_{it} - \theta\bar{X}_i)\beta + (u_{it} - \theta\bar{u}_i) \tag{6}$$

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 3.4.

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 (5). 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 **?**) and packages **Matrix** (see **?**) and **SparseM** (see **?**). Some optimized linear algebra routines are available in the R package **kinship** (see **?**) 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 (5) to compute $\hat{\beta}$, resulting in speed comparable to that of the demeaning solution.

3.4. 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:

$$\hat{V}_R(\beta) = (X^\top X)^{-1} \sum_{i=1}^{n} X_i^\top E_i X_i (X^\top X)^{-1} \tag{7}$$

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** (**?**) 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.

4. Managing data and formulae

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

[4] See packages **lmtest** (**?**) and **car** (**?**).

[5] Moreover, `coeftest()` provides a compact way of looking at coefficient estimates and significance diagnostics.

```
R> library("plm")
```

The four datasets used are `EmplUK` which was used by **?**, the `Grunfeld` data (**?**) which is used in several econometric books, the `Produc` data used by **?** and the `Wages` used by **?**.

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

4.1. 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 datasets but `Wages`. If this weren't the case, an `index` optional argument would have to be passed on to the estimating and testing functions.

As an example, as the `Wages` dataset lacks indexes altogether, but we know it is a balanced panel already sorted in (`id`, `time`) order, these can be explicitly reconstructed and put into the first two columns by `plm.data` specifying the number of individuals:

```
R> Wages <- plm.data(Wages, index = 595)
```

once for all. As an alternative, one could keep specifying `index = 595` into every command.

Datasets that have indexes but are, or are suspected to be, in the wrong order can be set in the same way, specifying the names of the `id` and `time` indexes instead of the number of individuals.

4.2. 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 performs 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. They have been therefore rewritten in order to handle correctly panel data.

The `Between`, `between`, `within`, `lag` and `diff` functions can be used directly in formulas to estimate models in all the estimation functions previously described.

4.3. Dynamic formula

Using the `lag` and `diff` functions to write a formula can be very cumbersome for dynamic models. A `dynformula` function is provided to ease the writing of such formulas. Suppose for example than one wants to estimate a model where employment depends on its own first two lags and on the second and third lags of wages and capital, all the variables except capital being in logs, and capital being differenced. The formula is then:

```
R> log(emp) ~ lag(log(emp), 1) + lag(log(emp), 2) + lag(log(wage),
+     2) + lag(log(wage), 3) + diff(capital, 2) + diff(capital,
+     3)
```

which can be easily obtained with `dynformula`:

```
R> dynformula(emp ~ wage + capital, log = list(capital = FALSE,
+     TRUE), lag = list(emp = 2, c(2, 3)), diff = list(FALSE, capital = TRUE))

log(emp) ~ lag(log(emp), 1) + lag(log(emp), 2) + lag(log(wage),
    2) + lag(log(wage), 3) + diff(capital, 2) + diff(capital,
    3)
```

The arguments `lag`, `diff` and `log` are lists which can be:

- unnamed, in this case, the length of the list must equal the number of variables,
- named, in this case the missing variables get the default values (no lags, no logs, no differences),
- partially named, in this case the unnamed element is the user defined default value.

The elements of `diff` and `log` lists should be logical values. For `lag`, it is either a number or a vector of two numbers: 3 means current value and first three lags, `c(1,2)` means first two lags.

5. Model estimation

5.1. 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 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:

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

R> 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 2784.458   52.768 0.282
individual    7089.800   84.201 0.718
theta:  0.86122

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
-178.00  -19.70    4.69   19.50  253.00

Coefficients :
             Estimate Std. Error t-value Pr(>|t|)
(Intercept) -57.834415  28.898935 -2.0013  0.04536 *
value         0.109781   0.010493 10.4627  < 2e-16 ***
capital       0.308113   0.017180 17.9339  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Total Sum of Squares:    2381400
Residual Sum of Squares: 548900
F-statistic: 328.837 on 2 and 197 DF, p-value: < 2.22e-16
```

For a `random` model, the `summary` method gives information about the variance of the components of the errors. Fixed effects may be extracted easily using `fixef`:

```
R> fixef(grun.fe)
```

```
         1          2          3          4          5          6
-70.296717 101.905814 -235.571841 -27.809295 -114.616813 -23.161295
         7          8          9         10
-66.553474 -57.545657 -87.222272  -6.567844
```

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.

```
R> summary(fixef(grun.fe))
```

```
    Estimate Std. Error t-value  Pr(>|t|)
1   -70.2967    49.7080 -1.4142   0.15730
2   101.9058    24.9383  4.0863 4.383e-05 ***
3  -235.5718    24.4316 -9.6421 < 2.2e-16 ***
4   -27.8093    14.0778 -1.9754   0.04822 *
5  -114.6168    14.1654 -8.0913 6.661e-16 ***
6   -23.1613    12.6687 -1.8282   0.06752 .
7   -66.5535    12.8430 -5.1821 2.194e-07 ***
8   -57.5457    13.9931 -4.1124 3.915e-05 ***
9   -87.2223    12.8919 -6.7657 1.327e-11 ***
10   -6.5678    11.8269 -0.5553   0.57867
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

In case of a two-ways effect model, an additional argument `effect` is required to extract fixed effects:

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

```
      1935       1936       1937       1938       1939       1940       1941
 -32.83632  -52.03372  -73.52633  -72.06272 -102.30660  -77.07140  -51.64078
      1942       1943       1944       1945       1946       1947       1948
 -53.97611  -75.81394  -75.93509  -88.51936  -64.00560  -72.22856  -76.55283
      1949       1950       1951       1952       1953       1954
-106.33142 -108.73243  -95.31723  -97.46866 -100.55428 -126.36254
```

5.2. 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 **?**, the default value,
- `walhus`: from **?**,
- `amemiya`: from **?**,
- `nerlove`: from **?**.

For example, to use the `amemiya` estimator:

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

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:

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

Twoways effects Random Effect Model
   (Swamy-Arora'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 2644.135   51.421 0.2359
individual    8294.716   91.075 0.7400
time           270.529   16.448 0.0241
theta  : 0.87475 (id) 0.29695 (time) 0.29595 (total)

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
-176.00  -18.00    3.02   18.00  233.00

Coefficients :
             Estimate Std. Error t-value Pr(>|t|)
(Intercept) -64.351811  31.183651 -2.0636  0.03905 *
value         0.111593   0.011028 10.1192  < 2e-16 ***
capital       0.324625   0.018850 17.2214  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Total Sum of Squares:    2038000
Residual Sum of Squares: 514120
F-statistic: 291.965 on 2 and 197 DF, p-value: < 2.22e-16
```

In the "effects" section of the result, the variance of the three elements of the error term and the three parameters used in the transformation are now printed. The two-ways effect model is for the moment only available for balanced panels.

Instrumental variable estimators

One or all of the models may be estimated using instrumental variables. The instruments are specified either as a one side formula in the argument `instruments`, or at the end of the formula after a | sign.

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

- `bvk`, from **?**, the default value,
- `baltagi`, from **?**.

We illustrate instrumental variables estimation with the employment equation. We use the first lag of wage as instrument of wage. The four following commands are similar:

```
R> emp.iv <- plm(emp ~ wage + capital | lag(wage, 1) + capital,
+     data = EmplUK, model = "random")
R> emp.iv <- plm(emp ~ wage + capital | . - wage + lag(wage, 1),
+     data = EmplUK, model = "random")
R> emp.iv <- plm(emp ~ wage + capital, instruments = ~lag(wage,
+     1) + capital, data = EmplUK, model = "random")
R> emp.iv <- plm(emp ~ wage + capital, instruments = ~. - wage +
+     lag(wage, 1), data = EmplUK, model = "random")
```

The Hausman-Taylor model (see **?**) may be estimated with `plm` by equating the `model` argument to `"ht"`.

5.3. Variable coefficients model

The `pvcm` function enables the estimation of variable coefficients models. Time or individual effects are introduced if `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 **?**) 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:

$$\hat{\beta} = \left(\sum_{i=1}^{n} \left(\hat{\Delta} + \hat{\sigma}_i^2 (X_i^\top X_i)^{-1} \right)^{-1} \right) \left(\hat{\Delta} + \hat{\sigma}_i^2 (X_i^\top X_i)^{-1} \right)^{-1} \hat{\beta}_i \tag{8}$$

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

$$\hat{\Delta} = \frac{1}{n-1} \sum_{i=1}^{n} \left(\hat{\beta}_i - \frac{1}{n} \sum_{i=1}^{n} \hat{\beta}_i \right) \left(\hat{\beta}_i - \frac{1}{n} \sum_{i=1}^{n} \hat{\beta}_i \right)^\top - \frac{1}{n} \sum_{i=1}^{n} \hat{\sigma}_i^2 (X_i^\top X_i)^{-1} \tag{9}$$

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

With the `Grunfeld` data, we get:

```
R> grun.varw <- pvcm(inv ~ value + capital, data = Grunfeld, model = "within")
R> grun.varr <- pvcm(inv ~ value + capital, data = Grunfeld, model = "random")
R> 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.
-211.500  -32.320   -4.283    9.048   12.710  579.200

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 2.248e-05 ***
capital      0.199418   0.052653  3.7874 0.0001522 ***
---
```

```
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 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: 532410000
Residual Sum of Squares: 2194300
Multiple R-Squared: 0.99588
```

5.4. Generalized method of moments estimator

The generalized method of moments is mainly used in panel data econometrics to estimate dynamic models (**??**).

$$y_{it} = \rho y_{it-1} + \beta^\top x_{it} + \mu_i + \epsilon_{it} \tag{10}$$

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

$$\Delta y_{it} = \rho \Delta y_{it-1} + \beta^\top \Delta x_{it} + \Delta \epsilon_{it} \tag{11}$$

Least squares are inconsistent because $\Delta\epsilon_{it}$ is correlated with Δy_{it-1}. y_{it-2} is a valid, but weak instrument (see **?**). 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:

$$W_i = \begin{pmatrix} y_1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & x_{i3} \\ 0 & y_1 & y_2 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & x_{i4} \\ 0 & 0 & 0 & y_1 & y_2 & y_3 & \dots & 0 & 0 & 0 & 0 & x_{i5} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \dots & \dots & y_1 & y_2 & \dots & y_{t-2} & x_{iT-2} \end{pmatrix} \tag{12}$$

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 minimize:

$$\left(\sum_{i=1}^n e_i(\beta)^\top W_i \right) A \left(\sum_{i=1}^n W_i^\top e_i(\beta) \right) \tag{13}$$

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:

$$A^{(1)} = \left(\sum_{i=1}^{n} W_i^\top H^{(1)} W_i \right)^{-1} \tag{14}$$

with:

$$H^{(1)} = d^\top d = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ 0 & -1 & 2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix} \tag{15}$$

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.

The GMM estimator is provided by the `pgmm` function. It's main argument is a `dynformula` which describes the variables of the model and the lag structure.

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.

In a GMM estimation, there are "normal" instruments and "GMM" instruments. GMM instruments are indicated with the `gmm.inst` argument (a one side formula) and the lags with the `lag.gmm` argument. 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.

The complete list of instruments can also be specified with the argument `instruments` which should be a one side formula (or `dynformula`).

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

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

```
R> emp.gmm <- pgmm(dynformula(emp ~ wage + capital + output, lag = list(2,
+     1, 0, 1), log = TRUE), EmplUK, effect = "twoways", model = "twosteps",
+     gmm.inst = ~log(emp), lag.gmm = list(c(2, 99)))
R> summary(emp.gmm)

Twoways effects Two steps model

Call:
pgmm(formula = log(emp) ~ lag(log(emp), 1) + lag(log(emp), 2) +
    log(wage) + lag(log(wage), 1) + log(capital) + log(output) +
    lag(log(output), 1), data = EmplUK, effect = "twoways", model = "twosteps",
```

```
    gmm.inst = ~log(emp), lag.gmm = list(c(2, 99)))

Unbalanced Panel: n=140, T=7-9, N=1031

Number of Observations Used:  611

Residuals
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-0.6191000 -0.0494800 -0.0004565 -0.0001841  0.0533500  0.6410000

Coefficients
                     Estimate Std. Error  z-value  Pr(>|z|)
lag(log(emp), 1)     0.474151   0.085303   5.5584 2.722e-08 ***
lag(log(emp), 2)    -0.052967   0.027284  -1.9413 0.0522200 .
log(wage)           -0.513205   0.049345 -10.4003 < 2.2e-16 ***
lag(log(wage), 1)    0.224640   0.080063   2.8058 0.0050192 **
log(capital)         0.292723   0.039463   7.4177 1.191e-13 ***
log(output)          0.609775   0.108524   5.6188 1.923e-08 ***
lag(log(output), 1) -0.446373   0.124815  -3.5763 0.0003485 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Sargan Test: chisq(25) = 30.11247 (p.value=0.22011)
Autocorrelation test (1): normal = -2.427829 (p.value=0.0075948)
Autocorrelation test (2): normal = -0.3325401 (p.value=0.36974)
Wald test for coefficients: chisq(7) = 371.9877 (p.value=< 2.22e-16)
Wald test for time dummies: chisq(6) = 26.90450 (p.value=0.00015090)
```

5.5. 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 **?**, 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 **?**, 13.9.1–2, p.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 of "random" effects[6].

The "random effect" general FGLS is estimated by:

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

 Random effects model

Call:
pggls(formula = log(emp) ~ log(wage) + log(capital), data = EmplUK,
    model = "random")

Unbalanced Panel: n=140, T=7-9, N=1031

Residuals
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-1.8440000 -0.3908000  0.0388700  0.0005525  0.4153000  1.4920000

Coefficients
              Estimate Std. Error z-value Pr(>|z|)
(Intercept)   1.751448   0.180100  9.7249  < 2e-16 ***
log(wage)    -0.132986   0.054292 -2.4495  0.01431 *
log(capital)  0.629621   0.018483 34.0648  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
Total Sum of Squares: 1853.6
Residual Sum of Squares: 386
Multiple R-squared: 0.79176
```

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

```
R> 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.

[6]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.

6. Tests

As sketched in Section 2, 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 6.6.

6.1. 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 :

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

        F statistic

data:  inv ~ value + capital
F = 5.7805, df1 = 18, df2 = 170, p-value = 1.219e-10
alternative hypothesis: unstability
```

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

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

6.2. 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:

- `bp`: ?,
- `honda`: ?, the default value,
- `kw`: ?,

- ghm: ?.

The effects tested are indicated with the `effect` argument (one of `individual`, `time` or `twoways`).
To test the presence of individual and time effects in the `Grunfeld` example, using the ? test, we use:

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

        Lagrange Multiplier Test - two-ways effects (Gourieroux, Holly and
        Monfort)

data:  inv ~ value + capital
chisq = 798.1615, df = 2, p-value < 2.2e-16
alternative hypothesis: significant effects
```

or

```
R> 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` models. Its main arguments are either two `plm` objects (the results of a `pooling` and a `within` model) or a formula.

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

        F test for twoways effects

data:  inv ~ value + capital
F = 17.4031, df1 = 28, df2 = 169, p-value < 2.2e-16
alternative hypothesis: significant effects

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

6.3. Hausman test

`phtest` computes the Hausman test which is based on the comparison of two sets of estimates (see ?). 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:

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

        Hausman Test

data:  inv ~ value + capital
chisq = 2.3304, df = 2, p-value = 0.3119
alternative hypothesis: one model is inconsistent
```

6.4. 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[7] 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[8]. 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 semi-parametric alternatives which are robust vs. heteroskedasticity and free from distributional assumptions.

Unobserved effects test

The unobserved effects test *à la Wooldridge* (see **?**, 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:

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

[7] Here we treat fixed and random effects alike, as components of the error term, according with the modern approach in econometrics (see **?**).

[8] Neglecting time effects may also lead to serial correlation in residuals (as observed in **?**, 10.4.1).

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 **?**, "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 **?**:

```
R> 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 = 8.207e-05
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 **?** and **?** and is implemented as an option in `pbsytest`:

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

        Baltagi and Li AR-RE joint test

data:  formula
chisq = 4187.597, df = 2, p-value < 2.2e-16
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?

? 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[9]. 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 **?** (see `pbltest`).

The serial correlation version is the default:

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

        Bera, Sosa-Escudero and Yoon locally robust test

data:  formula
chisq = 52.6359, df = 1, p-value = 4.015e-13
alternative hypothesis: AR(1) errors sub random effects
```

The BSY test for random effects is implemented in the one-sided version[10], which takes heed that the variance of the random effect must be non-negative:

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

        Bera, Sosa-Escudero and Yoon locally robust test

data:  formula
z = 57.9143, p-value < 2.2e-16
alternative hypothesis: random effects sub AR(1) errors
```

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

? and **?** 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`:

```
R> 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 < 2.2e-16
alternative hypothesis: AR(1)/MA(1) errors in RE panel models
```

[9] LM_3 in **?**.

[10] Corresponding to RSO^*_μ in the original paper.

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 any more, 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 7)..

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 (**?**, 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 3).

For the random effects model, **?** 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 **?**, 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"[11]. 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`:

```
R> grun.fe <- plm(inv ~ value + capital, data = Grunfeld, model = "within")
R> pbgtest(grun.fe, order = 2)

        Breusch-Godfrey/Wooldridge test for serial correlation in panel models
```

[11] 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 **?**, 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 **?**.

```
data:  inv ~ value + capital
chisq = 181.9437, df = 2, p-value < 2.2e-16
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 '...' operator.

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**, together with the function `linear.hypothesis()` in package **car**, 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 `linear.hypothesis()` to the latter model to test the above restriction on δ, supplying a heteroskedasticity- and autocorrelation-consistent covariance matrix (`vcovHC` with the appropriate options, in particular `method="arellano"`)[12].

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

        Wooldridge's test for serial correlation in FE panels

data:  plm.model
chisq = 312.2975, p-value < 2.2e-16
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, **?**, 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`).

[12] see Subsection 6.6.

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[13] $e_{it} \equiv u_{it} - u_{i,t-1}$ 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. **?** provides Monte-carlo evidence of the good empirical properties of the test.

On the other extreme (see **?**, 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:

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

ca roule

        Wooldridge's first-difference test for serial correlation in panels

data:  plm.model
chisq = 1.5251, p-value = 0.2169
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`.

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

ca roule

        Wooldridge's first-difference test for serial correlation in panels

data:  plm.model
chisq = 131.5482, p-value < 2.2e-16
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 6.6 in inference.

[13] Here, e_{it} for notational simplicity (and as in Wooldridge): equivalent to $\Delta\epsilon_{it}$ in the general notation of the paper.

6.5. 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[14].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 (**?**) 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

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

and distributed as a standard Normal.

Pesaran's (**?**) CD test

[14]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 **?**).

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

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"`. 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`.

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

        Pesaran CD test for cross-sectional dependence in panels

data:  formula
z = 5.3401, p-value = 9.292e-08
alternative hypothesis: cross-sectional dependence
```

If a different model specification (`within`, `random`, ...) is assumed consistent, one can resort to its residuals for testing[15] 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.

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

        Pesaran CD test for cross-sectional dependence in panels

data:  formula
z = 4.6612, p-value = 3.144e-06
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 (**?**), takes into account an appropriate subset of *neighbouring* cross-sectional units to check the null of no XSD against the alternative of *local*

[15] This is also the only solution when the time dimension's length is insufficient for estimating the heterogeneous model.

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 (**?**) for the special case considered in that paper. The same can be applied to the LM and $SCLM$ 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` [16].

6.6. Robust covariance matrix estimation

Robust estimators of the covariance matrix of coefficients are provided, mostly for use in Wald-type tests. `vcovHC` estimates three "flavours" of White's heteroskedasticity-consistent covariance matrix[17] (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.*

$$\Omega_i = \begin{bmatrix} \sigma_{i1}^2 & \dots & \dots & 0 \\ 0 & \sigma_{i2}^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & \dots & \sigma_{iT}^2 \end{bmatrix} \tag{16}$$

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 **?**, 13.7.1–2 and **?**, 10.7.2; `"arellano"` (see

[16] The very comprehensive package **spdep** for spatial dependence analysis (see **?**) 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 lagging the corresponding `nb`s through `nblag`.

[17] See **?** and **?**.

ibid. and the original ref. **?**) allows a fully general structure w.r.t. heteroskedasticity and serial correlation:

$$\Omega_i = \begin{bmatrix} \sigma_{i1}^2 & \sigma_{i1,i2} & \dots & \dots & \sigma_{i1,iT} \\ \sigma_{i2,i1} & \sigma_{i2}^2 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \sigma_{iT-1}^2 & \sigma_{iT-1,iT} \\ \sigma_{iT,i1} & \dots & \dots & \sigma_{iT,iT-1} & \sigma_{iT}^2 \end{bmatrix} \tag{17}$$

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 6.4 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 **?**).

The errors may be weighted according to the schemes proposed by **?** and **?** to improve small-sample performance[18].

The main use of `vcovHC` is together with testing functions from the **lmtest** and **car** packages. These typically allow passing the `vcov` parameter either as a matrix or as a function (see **?**). If one is happy with the defaults, it is easiest to pass the function itself:

```
R> library("lmtest")
R> re <- plm(inv ~ value + capital, data = Grunfeld, model = "random")
R> coeftest(re, vcovHC)

t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept) -57.834415  23.449626 -2.4663   0.01451 *
value         0.109781   0.012984  8.4551 6.186e-15 ***
capital       0.308113   0.051889  5.9379 1.284e-08 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

else one may do the covariance computation inside the call to `coeftest`, thus passing on a matrix:

```
R> coeftest(re, vcovHC(re, method = "white2", type = "HC3"))
```

For some tests, e.g. for multiple model comparisons by `waldtest`, one should always provide a function[19]. In this case, optional parameters are provided as shown below (see also **?**, p.12):

[18] 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.

[19] Joint zero-restriction testing still allows providing the `vcov` of the unrestricted model as a matrix, see the documentation of package **lmtest**.

```
R> 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  < 2.2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

Moreover, `linear.hypothesis` from package **car** may be used to test for linear restrictions:

```
R> library("car")
R> linear.hypothesis(re, "2*value=capital", vcov. = vcovHC)

Linear hypothesis test

Hypothesis:
2 value - capital = 0

Model 1: inv ~ value + capital
Model 2: restricted model

Note: Coefficient covariance matrix supplied.

  Res.Df  Df  Chisq Pr(>Chisq)
1    197
2    198  -1 3.4783    0.06218 .
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

A specific `vcovHC` method for `pgmm` objects is also provided which implements the robust covariance matrix proposed by **?** for generalized method of moments estimators.

7. plm versus nlme/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[20].

R is particularly strong on mixed models' estimation, thanks to the long-standing **nlme** package (see **?**) and the more recent **lme4** package, based on S4 classes (see **?**)[21]. 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.

7.1. 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[22] 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.

7.2. Some false friends

The *fixed/random effects* terminology in econometrics is often recognized to be misleading,

[20] 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.

[21] The standard reference on the subject of mixed models in S/R is **?**.

[22] 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 **?**.

as both are treated as random variates in modern econometrics (see e.g. **?**, 10.2.1). It has been recognized since Mundlak's classic paper (**?**) 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[23], 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 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.

7.3. 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 **?**) and the **nlme** estimation commands for maximum likelihood estimation of an equivalent specification[24].

The Laird-Ware representation for mixed models

A general representation for the linear mixed effects model is given in **?**.

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

[23] 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.

[24] 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).

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 $u_{ij} = 1b_{i1} + \epsilon_{ij}$. Below we report coefficients of Grunfeld's model estimated by GLS and then by ML

```
R> reGLS <- plm(inv ~ value + capital, data = Grunfeld, model = "random")
R> reML <- lme(inv ~ value + capital, data = Grunfeld, random = ~1 |
+     firm)
R> coef(reGLS)

(Intercept)       value     capital
-57.8344149   0.1097812   0.3081130

R> summary(reML)$coef$fixed

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

Variable coefficients, "random"

Swamy's variable coefficients model (**?**) 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 $\psi_{kk'} = \sigma^2_\mu I_N$, 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.

```
R> vcm <- pvcm(inv ~ value + capital, data = Grunfeld, model = "random",
+     effect = "time")
R> vcmML <- lme(inv ~ value + capital, data = Grunfeld, random = ~value +
+     capital | year)
R> coef(vcm)
```

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

R> summary(vcmML)$coef$fixed

(Intercept)       value     capital
-26.3558394   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):

```
R> vcmf <- pvcm(inv ~ value + capital, data = Grunfeld, model = "within",
+     effect = "time")
R> vcmfML <- lmList(inv ~ value + capital | year, data = Grunfeld)
```

Unrestricted fgls

The general, or unrestricted, feasible GLS, `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.

```
R> sGrunfeld <- Grunfeld[Grunfeld$firm %in% 4:6, ]
R> ggls <- pggls(inv ~ value + capital, data = sGrunfeld, model = "random")
R> gglsML <- gls(inv ~ value + capital, data = sGrunfeld, correlation = corSymm(form = ~1
+     year))
R> coef(ggls)

(Intercept)       value     capital
 1.19679342  0.10555908  0.06600166

R> summary(gglsML)$coef

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

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

7.4. 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[25] may be fitted by `gls()`, specifying the correlation structure in the `correlation` option[26]:

```
R> Grunfeld$year <- as.numeric(as.character(Grunfeld$year))
R> 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 **?**, chap 5), which is a very common specification in econometrics, may be fit by `lme` specifying an additional random intercept:

```
R> 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:

```
R> summary(reAR1ML)$coef$fixed

 (Intercept)        value      capital
-40.27650822   0.09336672   0.31323330

R> 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.

[25] 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*.

[26] note that the time index is coerced to numeric before the estimation.

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()`.

```
R> lmML <- gls(inv ~ value + capital, data = Grunfeld)
R> 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:

```
R> 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:

```
R> 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 2 196.3660  <.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:

```
R> 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.

8. 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 datasets from the econometric literature. The similarities and differences with the maximum likelihood approach to longitudinal data have also been briefly discussed.

We plan to expand the methods in this paper to systems of equations and to the estimation of models with autoregressive errors. Addition of covariance estimators robust vs. cross-sectional correlation are also in the offing. Lastly, conditional visualization features in the R environment seem to offer a promising toolbox for visual diagnostics, which is another subject for future work.

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.

Affiliation:

Yves Croissant
LET-ISH
Avenue Berthelot
F-69363 Lyon cedex 07
Telephone: +33/4/78727249
Fax: +33/4/78727248
E-mail: yves.croissant@let.ish-lyon.cnrs.fr

Giovanni Millo
DiSES, Un. of Trieste and R&D Dept., Generali SpA
Via Machiavelli 4
34131 Trieste (Italy)
Telephone: +39/040/671184
Fax: +39/040/671160
E-mail: Giovanni_Millo@Generali.com