Package ‘plm’

February 20, 2015

Version 1.4-0
Date 2013-12-24
Title Linear Models for Panel Data
Depends R (>= 2.10), Formula (>= 0.2-0), stats
Imports MASS, bdsmatrix, zoo, nlme, sandwich
Suggests lattice, car, AER, lmtest
Description A set of estimators and tests for panel data.
License GPL (>= 2)
URL http://www.R-project.org
Author Yves Croissant [aut, cre],
       Giovanni Millo [aut],
       Arne Henningsen [ctb],
       Ott Toomet [ctb],
       Christian Kleiber [ctb],
       Achim Zeileis [ctb]
Maintainer Yves Croissant <yves.croissant@univ-reunion.fr>
Repository CRAN
Repository/R-Forge/Project plm
Repository/R-Forge/Revision 102
Repository/R-Forge/DateTimeStamp 2013-12-28 08:20:15
Date/Publication 2013-12-28 16:01:54
NeedsCompilation no

R topics documented:

  Cigar ................................................................. 3
  cipstest ........................................................... 4
  Crime ............................................................... 5
  dynformula ........................................................ 6
  EmplUK ............................................................. 7
R topics documented:

ercomp ......................................................... 8
fixef.plm ...................................................... 9
Gasoline ....................................................... 11
Grunfeld ....................................................... 12
Hedonic ....................................................... 13
index.plm .................................................... 14
LaborSupply .................................................. 15
Males ........................................................ 16
mtest ........................................................ 17
pggtest ......................................................... 18
pbtest ........................................................ 19
pbsytest ....................................................... 20
pecce ........................................................ 22
pcdtest ......................................................... 23
pdata.frame ................................................... 25
pdim ........................................................ 27
pdwtest ......................................................... 28
pFormula ....................................................... 29
pFtest ........................................................ 31
pglsl ......................................................... 32
pgmm ........................................................ 34
ph ............................................................ 37
phtest ......................................................... 38
plm .......................................................... 39
plm.data ..................................................... 42
plmtest ....................................................... 43
pmg .......................................................... 44
pmodel.response ............................................ 46
pooltest ....................................................... 47
Produc ....................................................... 48
pseries ....................................................... 49
purtest ....................................................... 51
pvar ........................................................ 53
pvcm ......................................................... 54
pwartest ...................................................... 56
pwildtest ..................................................... 57
pwtest ......................................................... 59
r.squared ..................................................... 60
sargan ....................................................... 61
Snmesp ....................................................... 62
SumHes ....................................................... 63
vcovBK ......................................................... 64
vcovHC ....................................................... 65
vcovSCC ..................................................... 68
Wages ........................................................ 69

Index ......................................................... 71
**Description**

a panel of 46 observations from 1963 to 1992  
*total number of observations*: 1380  
*observation*: regional  
*country*: United States

**Usage**

data(Cigar)

**Format**

A data frame containing:

- **state**: state abbreviation
- **year**: the year
- **price**: price per pack of cigarettes
- **pop**: population
- **pop16**: population above the age of 16
- **cpi**: consumer price index (1983=100)
- **ndi**: per capita disposable income
- **sales**: cigarette sales in packs per capita
- **pimin**: minimum price in adjoining states per pack of cigarettes

**Source**

Online complements to Baltagi (2001).  
http://www.wiley.com/legacy/wileychi/baltagi/.

**References**


Cross-sectionally Augmented IPS Test for Unit Roots in Panel Models

Description

Cross-sectionally augmented Im, Pesaran and Shin test for unit roots in panel models.

Usage

cipstest(x, lags = 2, type=c("trend", "drift", "none"),
       model = c("cmg","mg","dmg"), truncated=FALSE, ...)

Arguments

x          an object of class "pseries",
lags      lag order for Dickey-Fuller augmentation,
type       one of c("trend", "drift", "none"),
model      one of c("cmg","mg","dmg"),
truncated  logical specifying whether to calculate the truncated version of the test,
...        further arguments passed to critvals.

Details

This cross-sectionally augmented version of the IPS unit root test (H0: the pseries has a unit root) is a so-called second-generation panel unit root test: it is in fact robust against cross-sectional dependence, provided that the default type="cmg" is calculated. Else one can obtain the standard (model="mg") or cross-sectionally demeaned (model="dmg") versions of the IPS test.

Value

An object of class "htest".

Author(s)

Giovanni Millo

References


See Also

purtest
Examples

```r
data("Produc", package = "plm")
Produc <- pdata.frame(Produc, index=c("state","year"))
## check whether the log of GDP is trend-stationary
cipstest(Produc$gsp, type="trend")
```

---

**Crime in North Carolina**

**Description**

A panel of 90 observations (counties) from 1981 to 1987

- total number of observations: 630
- observation: regional
- country: United States

**Usage**

```r
data(Crime)
```

**Format**

A data frame containing:

- `county`: county identifier
- `year`: year from 1981 to 1987
- `crmrte`: crimes committed per person
- `prbarr`: 'probability' of arrest
- `prbconv`: 'probability' of conviction
- `prbpris`: 'probability' of prison sentence
- `avgse`: average sentence, days
- `polpc`: police per capita
- `density`: people per square mile
- `taxpc`: tax revenue per capita
- `region`: factor. One of 'other', 'west' or 'central'.
- `smsa`: factor. Does the individual reside in a SMSA (standard metropolitan statistical area)?
- `pctmin`: percentage minority in 1980
- `wcon`: weekly wage in construction
- `wtuc`: weekly wage in trans, util, commun
- `wtrd`: weekly wage in whole sales and retail trade
- `wfir`: weekly wage in finance, insurance and real estate
wser weekly wage in service industry
wmfg weekly wage in manufacturing
wfed weekly wage of federal employees
wsta weekly wage of state employees
wloc weekly wage of local governments employees
mix offence mix: face-to-face/other
pctymle percentage of young males

Source

Online complements to Baltagi (2001).
http://www.wiley.com/legacy/wileychi/baltagi/.

See also Journal of Applied Econometrics data archive entry for Baltagi (2006) at

References


--

dynformula Dynamic Formula

Description

A function to easily create a formula with lags and differences

Usage

dynformula(formula, lag.form = NULL, diff.form = NULL, log.form = NULL)

Arguments

formula a formula,
lag.form a list containing the lag structure of each variable in the formula,
diff.form a vector (or a list) of logical values indicating whether variables should be differ-
log.form a vector (or a list) of logical values indicating whether variables should be in logarithms.
Details

lag.form is a list, diff.form and log.form are vectors (or lists) that should be of length equal to the total number of variables. Each element of these lists/vectors is:

- either a vector of length 2 (c(1, 4) means lags 1,2,3 and 4) or a scalar (3 means lags 0,1,2,3 except for the left-hand side variable for which it is 1,2,3) for lag.form.
- logical values for diff.form and log.form.

It can also be an incomplete named list/vector (for example, to apply the transformation for only some variables) with eventually an unnamed element which then is the default value.

Value

An object of class c("dynformula","formula"), which is a formula with four additional attributes: var, the names of the variables in the formula, lag, diff, and log, which store the information about lags, differences and logs, respectively.

A formula method coerces the dynformula object to a standard formula.

Author(s)

Yves Croissant

Examples

# all variables in log, x1, x2 and x3 laged twice, y laged once and x3 differenced
z <- dynformula(y ~ x1 + x2 + x3, lag.form = list(z, y = 1),
              diff.form = c(x3 = TRUE), log.form = TRUE)
formula(z)

Data

An unbalanced panel of 140 observations from 1976 to 1984

total number of observations : 1031
observation : firms
country : United Kingdom

Usage

data(EmplUK)
**Format**

A data frame containing:
- **firm** firm index
- **year** year
- **sector** the sector of activity
- **emp** employment
- **wage** wages
- **capital** capital
- **output** output

**Source**


---

ercomp  
**Estimation of the error components**

**Description**

This function enables the estimation of the variance components of a panel model.

**Usage**

```r
ercomp(object, ...)  
## S3 method for class 'formula'
ercomp(object, data, effect = c("individual", "time", "twoways"),
method = c("swar", "walhus", "amemiya", "nerlove", "kinla"),
index = NULL, ...)

## S3 method for class 'plm'
ercomp(object, ...)

## S3 method for class 'ercomp'
print(x, digits = max(3,getOption("digits") - 3), ...)
```

**Arguments**

- **object**
  - a formula or a plm object,
- **data**
  - a data.frame,
- **method**
  - method of estimation for the variance components, see `lm` for details,
- **effect**
  - the effects introduced in the model, see `lm` for details,
- **index**
  - the indexes,
- **x**
  - an ercomp object,
- **digits**
  - digits,
- **...**
  - further arguments.
Value

An object of class "ercomp": a list containing a list called sigma2 which contains the estimates of the variance components, and theta which is the parameters used for the transformation of the variables.

Author(s)

Yves Croissant

References


See Also

plm where the estimates of the variance components are used if a random effects model is estimated

Examples

data("Produc", package = "plm")
# an example of the formula method
ercomp(log(gsp)-log(pcap)+log(pc)+log(emp)+unemp, data=Produc, method="walhus",effect="time")
# same with the plm method
z <- plm(log(gsp)-log(pcap)+log(pc)+log(emp)+unemp,
      data=Produc, random.method="walhus",
      effect="time",method="random")
ercomp(z)
# a two–ways model
ercomp(log(gsp)-log(pcap)+log(pc)+log(emp)+unemp, data=Produc, method="amemiya",effect="twoways")
Usage

```r
## S3 method for class 'plm'
fixef(object, effect = NULL,
    type = c("level", "dmean", "dmean"), ...)
## S3 method for class 'fixef'
print(x, digits = max(3,getOption("digits") - 2),
    width =getOption("width"), ...)
## S3 method for class 'fixef'
summary(object, ...)
## S3 method for class 'summary.fixef'
print(x, digits = max(3,getOption("digits") - 2),
    width =getOption("width"), ...)
```

Arguments

- `x, object`: an object of class "plm", an object of class "fixef" for the print and the summary method,
- `effect`: one of "individual" or "time", only relevant in case of two–ways effects models,
- `type`: one of level, dmean or dfirst.
- `digits`: digits,
- `width`: the maximum length of the lines in the print output,
- `...`: further arguments.

Details

The summary method prints the effects in deviation from the overall intercept, the standard error and the t–values.

Value

An object of class "fixef". It is a numeric vector containing the fixed effects with two attributes: `se` which contains the standard errors and `intercept` which is the overall intercept. With the `type` argument, the fixed effects may be returned in levels, as deviations from the overall mean or as deviations from the first value of the index.

Author(s)

Yves Croissant

See Also

`plm`
Gasoline

Examples

data("Grunfeld", package = "plm")
gi <- plm(inv ~ value + capital, data = Grunfeld, model = "within")
fixef(gi)
summary(fixef(gi))
# extract time effects in a twoways effects model
gi <- plm(inv ~ value + capital, data = Grunfeld,
model = "within", effect = "twoways")
fixef(gi,effect = "time")

Description

A panel of 18 observations from 1960 to 1978

*total number of observations*: 342

*observation*: country

*country*: OECD

Usage

data(Gasoline)

Format

A data frame containing:

*country* a factor with 18 levels

*year* the year

*lgaspcar* logarithm of motor gasoline consumption per car

*lincomep* logarithm of real per-capita income

*lrpmg* logarithm of real motor gasoline price

*lcarpcap* logarithm of the stock of cars per capita

Source

Online complements to Baltagi (2001).

http://www.wiley.com/legacy/wileychi/baltagi/.

References


Grunfeld

Grunfeld’s Investment Data

Description

A panel of 10 observations from 1935 to 1954

total number of observations : 200

observation : production units

country : United States

Usage

data(Grunfeld)

Format

A data frame containing :

firm  observation
year  date
inv  gross Investment
value  value of the firm
capital  stock of plant and equipment

Source

Online complements to Baltagi (2001).

http://www.wiley.com/legacy/wileychi/baltagi/

References


See Also

For the complete Grunfeld data (11 firms), see Grunfeld, in the AER package.
Hedonic

Hedonic Prices of Census Tracts in the Boston Area

Description

A cross-section

- number of observations: 506
- observation: regional
- country: United States

Usage

data(Hedonic)

Format

A dataframe containing:

- mv median value of owner-occupied homes
- crim crime rate
- zn proportion of 25,000 square feet residential lots
- indus proportion of nonretail business acres
- chas is the tract bounds the Charles River?
- nox annual average nitrogen oxide concentration in parts per hundred million
- rm average number of rooms
- age proportion of owner units built prior to 1940
- dis weighted distances to five employment centers in the Boston area
- rad index of accessibility to radial highways
- tax full value property tax rate ($/10,000)
- ptratio pupil/teacher ratio
- blacks proportion of blacks in the population
- lstat proportion of population that is lower status
- townid town identifier

Source

Online complements to Baltagi (2001).

http://www.wiley.com/legacy/wileychi/baltagi/.
index.plm

References


---

index.plm

Extract the indexes of panel data

Description

This function extracts the information about the structure of the individual and time dimensions of panel data.

Usage

```r
## S3 method for class 'pindex'
index(x, which = NULL, ...)
## S3 method for class 'pdata.frame'
index(x, which = NULL, ...)
## S3 method for class 'pseries'
index(x, which = NULL, ...)
## S3 method for class 'panelmodel'
index(x, which = NULL, ...)
```

Arguments

- `x` an object of class "pindex", "pdata.frame", "pseries" or "panelmodel",
- `which` the index(es) to be extracted (see details),
- `...` further arguments.

Details

Panel data are stored in a "pdata.frame" which has an "index" attribute. Fitted models in "plm" have a "model" element which is also a "pdata.frame" and therefore also have an "index" attribute. Finally, each series in a "pdata.frame" is of class "pseries", which also have this "index" attribute. "index" methods are available for all these objects. The argument "which" indicates which index should be extracted. If NULL, both indexes are extracted, but "which" can also be a vector of length 1 or 2 containing either characters (the names of the individual and of the time periods or "id" and "time") or integers (1 for the individual index and 2 for the time index.)

Value

A vector or a "data.frame" containing either one or both indexes.
**LaborSupply**

**Author(s)**
Yves Croissant

**See Also**
plm

**Examples**

data("Grunfeld", package = "plm")
Gr <- pdata.frame(Grunfeld, index = c("firm", "year"))
m <- plm(inv ~ value + capital, data = Gr)
index(m, "firm")
index(Gr, "time")
index(Gr$inv, c(2, 1))
index(m, "id")

---

**LaborSupply**  
**Wages and Hours Worked**

**Description**

A panel of 532 observations from 1979 to 1988  
*number of observations: 5320*

**Usage**

data(LaborSupply)

**Format**

A data frame containing:

- `lnhr` log of annual hours worked
- `lnwg` log of hourly wage
- `kids` number of children
- `age` age
- `disab` bad health
- `id` id
- `year` year

**Source**

Online complements to Ziliak (1997).  
References

Males  Wages and Education of Young Males

Description
A panel of 545 observations from 1980 to 1987

 total number of observations: 4360
 observation: individuals
country: United States

Usage
data(Males)

Format
A data frame containing:

 nr  identifier
 year  year
 school  years of schooling
 exper  years of experience (computed as age \(-\) school)
 union  wage set by collective bargaining?
 ethn  a factor with levels black, hisp, other
 married  married?
 health  health problem?
 wage  log of hourly wage
 industry  a factor with 12 levels
 occupation  a factor with 9 levels
 residence  a factor with levels rural area, north east, northern central, south

Source
Journal of Applied Econometrics data archive
References


---

mtest

*Arellano–Bond test of Serial Correlation*

Description

Test of serial correlation for models estimated by GMM

Usage

mtest(object, order=1, vcov=NULL)

Arguments

- **object**: an object of class "pgmm",
- **order**: the order of the serial correlation (1 or 2),
- **vcov**: a matrix of covariance for the coefficients or a function to compute it.

Details

The Arellano–Bond test is a test of correlation based on the residuals of the estimation. By default, the computation is done with the standard covariance matrix of the coefficients. A robust estimator of this covariance matrix can be supplied with the `vcov` argument.

Value

An object of class "htest".

Author(s)

Yves Croissant

References


See Also

`pgmm`
Examples

data("EmplUK", package = "plm")
ar <- pgmM(log(emp) ~ lag(log(emp), 1:2) + lag(log(wage), 0:1) +
lag(log(capital), 0:2) + lag(log(output), 0:2) | lag(log(emp), 2:99),
data = EmplUK, effect = "twoways", model = "twosteps")
mtest(ar, 1)
mtest(ar, 2, vcovHC)

pbgtest

Breusch–Godfrey Test for Panel Models

Description

Test of serial correlation for (the idiosyncratic component of) the errors in panel models.

Usage

pbgtest(x,...)
## S3 method for class 'panelmodel'
pbgtest(x, order = NULL, ...)
## S3 method for class 'formula'
1

Arguments

x an object of class "panelmodel" or of class "formula",
order an integer indicating the order of serial correlation to be tested for. Defaults to
the minimum number of observations over the time dimension,
... further arguments.

Details

This Lagrange multiplier test uses the auxiliary model on (quasi-)demeaned data taken from a
model of class plm which may be a pooling (the default), random or within model. It performs
a Breusch–Godfrey test (using bgtest from package lmtest) on the residuals of the (quasi-
demeaned model, which should be serially uncorrelated under the null of no serial correlation in
idiosyncratic errors, as illustrated in Wooldridge (2002). The function takes the demeaned data,
estimates the model and calls bgtest.

Unlike most other tests for serial correlation in panels, this one allows to choose the order of correlation to test for.

Value

An object of class "htest".
Author(s)
Giovanni Millo

References

See Also
*pdwtest* for the analogous panel Durbin–Watson test, *bgtest* for the Breusch–Godfrey test for serial correlation in the linear model. *pbltest*, *pbsytest*, *pwartest* and *pwfdtest* for other serial correlation tests for panel models.

Examples
```r
data("Grunfeld", package = "plm")
g <- plm(inv ~ value + capital, data = Grunfeld, model = "random")
pbgtest(g)
pbgtest(g, order = 4)

## formula interface
pbgtest(inv ~ value + capital, data = Grunfeld, model = "random")
```

---

**pbltest**

*Baltagi and Li Serial Dependence Test For Random Effects Models*

Description
Baltagi and Li (1995)'s Lagrange multiplier test for AR(1) or MA(1) idiosyncratic errors in panel models with random effects.

Usage
```
pbltest(x, data, alternative = c("twosided","onesided"), index = NULL, ...)
```

Arguments
- **x**: a model formula,
- **data**: a data.frame,
- **alternative**: one of "twosided", "onesided". Selects either $H_A : \rho \neq 0$ or $H_A : \rho = 0$ (i.e., the Normal or the Chi-squared version of the test),
- **index**: the index of the data.frame,
- **...**: further arguments.
Details

This is a Lagrange multiplier test for the null of no serial correlation, against the alternative of either an AR(1) or an MA(1) process, in the idiosyncratic component of the error term in a random effects panel model (as the analytical expression of the test turns out to be the same under both alternatives, see Baltagi and Li (1995, 1998)). The alternative argument, defaulting to twosided, allows testing for positive serial correlation only, if set to onesided.

Value

An object of class "htest".

Author(s)

Giovanni Millo

References


See Also

`pdwtest`, `bgtest`, `pbsytest`, `pwarstest` and `pwfdtest` for other serial correlation tests for panel models.

Examples

```r
data("Grunfeld", package = "plm")
pbltest(inv ~ value + capital, data = Grunfeld)
```

Description

Test for residual serial correlation (or individual random effects) locally robust vs. individual random effects (serial correlation) for panel models and joint test by Baltagi and Li.

Usage

```r
pbsytest(x,...)
## S3 method for class 'panelmodel'
pbsytest(x, test = c("ar","re","j"), ...)
## S3 method for class 'formula'
pbsytest(x, data, ..., test = c("ar","re","j"))
```
Arguments

- `x`: an object of class "formula" or of class `panelmodel`.
- `data`: a `data.frame`.
- `test`: a character string indicating which test to perform: first-order serial correlation (ar), random effects (re) or joint test for either of them (j).
- ... further arguments.

Details

These Lagrange multiplier tests are robust vs. local misspecification of the alternative hypothesis, i.e. they test the null of serially uncorrelated residuals against AR(1) residuals in a pooling model, allowing for local departures from the assumption of no random effects; or they test the null of no random effects allowing for local departures from the assumption of no serial correlation in residuals. They use only the residuals of the pooled OLS model and correct for local misspecification as outlined in Bera *et al.* (2001).

The joint test is due to Baltagi and Li (1991) and is added for convenience under this same interface.

Value

An object of class "htest".

Author(s)

Giovanni Millo

References


See Also

- `plmtest` for individual and/or time random effects tests based on a correctly specified model;
- `pbltest`, `pbgtest` and `pdwtest` for serial correlation tests in random effects models.

Examples

```r
## Example in Bera et al.
data(Grunfeld, package = "plm")
## Bera et al. use a subset of the original Grunfeld data,
## so results are slightly different here
## default is AR testing
pbsytest(inv ~ value + capital, data = Grunfeld, index = c("firm","year"))
pbsytest(inv ~ value + capital, data = Grunfeld, test="re", index = c("firm","year"))
pbsytest(inv ~ value + capital, data = Grunfeld, test="j", index = c("firm","year"))
```
**pcce**  

**Common Correlated Effects estimators**

### Description

Common Correlated Effects Mean Groups (CCEMG) and Pooled (CCEP) estimators for panel data with common factors (balanced or unbalanced)

### Usage

```r
pcce(formula, data, subset, na.action,
model=c("mg", "p"),
residuals = c("defactored", "standard"),
index = NULL, trend = FALSE, ...)
```

### Arguments

- `formula` a symbolic description of the model to be estimated,
- `object, x` an object of class `pcce`,
- `data` a `data.frame`,
- `subset` see `lm`,
- `na.action` see `lm`,
- `model` one of `c("mg", "p")`, selects Mean Groups vs. Pooled CCE model,
- `residuals` one of `c("defactored", "standard")`, allows for returning different kinds of residuals,
- `index` the indexes, see `plm.data`,
- `trend` logical specifying whether an individual-specific trend has to be included,
- `digits` digits,
- `width` the maximum length of the lines in the print output,
- `...` further arguments.

### Details

`pcce` is a function for the estimation of linear panel models by the Common Correlated Effects Mean Groups or Pooled estimator, consistent under the hypothesis of unobserved common factors and idiosyncratic factor loadings; CCE estimators work by augmenting the model by cross-sectional averages of the dependent variable and regressors in order to account for the common factors, and adding individual intercepts and possibly trends.
An object of class c("pcce", "panelmodel") containing:

- **coefficients**: the vector of coefficients,
- **residuals**: the vector of residuals,
- **fitted.values**: the vector of fitted.values,
- **vcov**: the covariance matrix of the coefficients,
- **df.residual**: degrees of freedom of the residuals,
- **model**: a data.frame containing the variables used for the estimation,
- **call**: the call,
- **sigma**: always NULL, sigma is here only for compatibility reasons (to allow using the same summary and print methods as pggls),
- **indcoef**: the matrix of individual coefficients from separate time series regressions.

**Author(s)**

Giovanni Millo

**References**


**Examples**

data("Produc", package = "plm")
ccepmod <- pcce(log(gsp) ~ log(cap) + log(pc) + log(emp) + unemp, data = Produc, model="p")
summary(ccepmod)

---

Tests of cross-section dependence for panel models

**Description**

Pesaran’s CD or Breusch-Pagan’s LM (local or global) tests for cross sectional dependence in panel models

**Usage**

pcdtest(x, ...)

## S3 method for class 'panelmodel'
pcdtest(x, test = c("cd", "sclm", "lm", "rho", "absrho"),
        w = NULL, ...)

## S3 method for class 'formula'
pcdtest(x, data, index = NULL, model = NULL,
         test = c("cd", "sclm", "lm", "rho", "absrho"),
         w = NULL, ...)
Arguments

- **x**: an object of class `formula` or `panelmodel`, describing the model to be tested.
- **data**: a `data.frame`.
- **index**: an optional numerical index, in case `data` has to be formatted by `plm.data`.
- **model**: an optional character string indicating which type of model to estimate; if left to `NULL`, the original heterogeneous specification of Pesaran is used.
- **test**: the type of test statistic to be returned. One of "cd" for Pesaran’s CD statistic, "lm" for Breusch and Pagan’s original LM statistic, "sclm" for the scaled version of Breusch and Pagan’s LM statistic, or "rho", "absrho" for average (absolute) correlation coefficient.
- **w**: an `n x n` matrix describing proximity between observations, with `w_{ij} = a` where `a` is any number such that `as.logical(a)=TRUE`, if `i, j` are neighbours, 0 or any number `b` such that `as.logical(b)=FALSE` elsewhere.
- **...**: further arguments to be passed on to `plm`, such as e.g. `effect` or `random.method`.

Details

These tests are originally meant to use the residuals of separate estimation of one time-series regression for each cross-sectional unit in order to check for cross-sectional dependence. If a different model specification (within, random, ...) is assumed consistent, one can resort to its residuals for testing (which is common, e.g., when the time dimension’s length is insufficient for estimating the heterogeneous model). If the time dimension is insufficient and `model=NULL`, the function defaults to estimation of a `within` model and issues a warning. The main argument of this function may be either a model of class `panelmodel` or a `formula` and `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. A local version of either test can be computed supplying a proximity matrix (coercible to `logical`) providing information on whether any pair of observations are neighbours or not. 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 need not be binary, so commonly used “row-standardized” matrices can be employed as well. `nb` objects from `spdep` must instead be transformed into matrices by `nb2mat` before using.

Value

An object of class “htest”.

References


Examples

data(Grunfeld, package = "plm")
## test on heterogeneous model (separate time series regressions)
pcdtest(inv ~ value + capital, data=Grunfeld,
    index = c("firm","year"))
## test on two-way fixed effects homogeneous model
pcdtest(inv ~ value + capital, data=Grunfeld, model="within",
    effect="twoways", index = c("firm","year"))
## test on model object
g <- plm(inv ~ value + capital, data=Grunfeld, index = c("firm","year"))
pcdtest(g)
## scaled LM test
pcdtest(g,test="sclm")

pdata.frame

**data.frame for panel data**

Description

An object of this class is a data.frame with an attribute that describes its time and individual dimensions.

Usage

pdata.frame(x, index = NULL, drop.index = FALSE, row.names = TRUE)
## S3 method for class 'pdata.frame'
x[i, j, drop = TRUE]
## S3 method for class 'pdata.frame'
x[[y]]
## S3 method for class 'pdata.frame'
x$y
## S3 method for class 'pdata.frame'
print(x, ...)
## S3 method for class 'pdata.frame'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

Arguments

x a data.frame for the pdata.frame function and a pdata.frame for the methods,
i see Extract,
j see Extract,
y one of the columns of the data.frame,
index this argument indicates the individual and time indexes. See details,
drop see Extract,
drop.index should the indexes be removed from the data.frame?
optional see as.data.frame
row.names should “fancy” row names be computed?
... further arguments

Details

The index argument indicates the dimensions of the panel. It can be:

- a character string which is the name of the individual index variable, in this case a new variable called “time” which contains the time index is added,
- an integer, the number of individuals in case of balanced panel, in this case two new variables “time” and “id” which contain the individual and the time indexes are added,
- a vector of two character strings which contains the names of the individual and of the time indexes.

The index attribute is a data.frame which contains the individual and the time indexes. The "[" and "]" extract a series from the pdata.frame. The "index" attribute is then added to the series and a class attribute "pseries" is added. The "[" method behaves as for data.frame, except that the extraction is also applied to the index attribute. as.data.frame removes the index from the pdata.frame and adds it to every series.

Value

a pdata.frame object: this is a data.frame with an index attribute which is a data.frame with two variables, the individual and the time indexes.

Author(s)

Yves Croissant

Examples

data("Wages", package = "plm")
Wag <- pdata.frame(Wages, 595)

# Gasoline contains two variables which are individual and time indexes
data("Gasoline", package = "plm")
Gas <- pdata.frame(Gasoline, c("country","year"), drop = TRUE)

# Hedonic is an unbalanced panel, townid is the individual index
data("Hedonic", package="plm")
Hed <- pdata.frame(Hedonic, "townid", row.names = FALSE)
**Description**

This function checks the number of individuals and time observations in the panel and whether it is balanced or not.

**Usage**

```r
pdim(x,...)
# S3 method for class 'data.frame'
pdim(x, index = NULL, ...)
# S3 method for class 'panelmodel'
pdim(x, ...)
# S3 method for class 'pdata.frame'
pdim(x, ...)
```

**Arguments**

- `x`: a data.frame, a pdata.frame or a panelmodel object,
- `index`: see pdata.frame,
- `...`: further arguments.

**Details**

`pdim` is called by the estimation functions.

**Value**

An object of class `pdim` containing the following elements:

- `nT`: a list containing `n`, the number of individuals, `T`, the number of time observations, `N` the total number of observations,
- `TinT`: a list containing two vectors: `Ti` gives the number of observations for each individual and `nt` gives the number of individuals observed for each period,
- `balanced`: a logical value: TRUE for a balanced panel, FALSE for an unbalanced panel,
- `panel.names`: a list of character vectors: `id.names` contains the names of each individual and `time.names` contains the names of each period.

**Author(s)**

Yves Croissant

**See Also**

`pdata.frame`
Examples

```r
# There are 595 individuals
data("Wages", package = "plm")
pdim(Wages, 595)

# Gasoline contains two variables which are individual and time indexes
# and are the first two variables
data("Gasoline", package="plm")
pdim(Gasoline)

# Hedonic is an unbalanced panel, townid is the individual index
data("Hedonic", package = "plm")
pdim(Hedonic, "townid")

# An example of the panelmodel method
data("Produc", package = "plm")
z <- plm(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, data = Produc,
model = "random", subset = gsp > 5000)
pdim(z)
```

---

**pdwtest**  
*Durbin–Watson Test for Panel Models*

**Description**

Test of serial correlation for (the idiosyncratic component of) the errors in panel models.

**Usage**

```r
pdwtest(x, ...)
```

## S3 method for class 'panelmodel'

```r
pdwtest(x, ...)
```

## S3 method for class 'formula'

```r
pdwtest(x, data, ...)
```

**Arguments**

- `x` an object of class "panelmodel" or of class "formula".
- `data` a data.frame,
- `...` further arguments to be passed on to `dwtest`.  

---
Details

This Durbin–Watson test uses the auxiliary model on (quasi-)demeaned data taken from a model of class plm which may be a pooling (the default), random or within model. It performs a dw test (using dwtest from package lmtest) on the residuals of the (quasi-)demeaned model, which should be serially uncorrelated under the null of no serial correlation in idiosyncratic errors. The function takes the demeaned data, estimates the model and calls dwtest.

Value

An object of class "htest".

Author(s)

Giovanni Millo

References


See Also

pbgtest for the analogous Breusch–Godfrey test, dwtest for the Breusch–Godfrey test for serial correlation in the linear model. pbltest, pbsytest, pwartest and pwfdtest for other serial correlation tests for panel models.

Examples

```r
data("Grunfeld", package = "plm")
g <- plm(inv ~ value + capital, data = Grunfeld, model="random")
pdwtest(g)
pdwtest(g, alternative="two.sided")
## formula interface
pdwtest(inv ~ value + capital, data=Grunfeld, model="random")
```

Description

pFormula is a Formula object, with methods useful for panel data.
Usage

pFormula(object)

## S3 method for class 'pFormula'
as.Formula(x, ...)
## S3 method for class 'pFormula'
model.frame(formula, data, ...,
   lhs = NULL, rhs = NULL)
## S3 method for class 'pFormula'
model.matrix(object, data,
   model = c("pooling","within","Between",
   "between","mean","random","fd"),
   effect = c("individual","time","twoways"),
   rhs = 1,
   theta = NULL, ...)

Arguments

object, formula, x
   an object of class "pFormula",

   data
      a data.frame,

effect
   the effects introduced in the model, one of "individual", "time" or "twoways",

   model
      one of "pooling", "within", "between", "random", "fd" and "ht",

   theta
      the parameter for the transformation if model = "random",

   lhs

   rhs

... further arguments.

Details

The lhs and rhs arguments are inherited from Formula. The model.frame method returns a
data.frame object. The model.matrix enables the transformation specified by the model and
effect arguments.

Value

An object of class c("pFormula", "Formula").

Author(s)

Yves Croissant
pFtest

F Test for Individual and/or Time Effects

Description

Test of individual and/or time effects based on the comparison of the within and the pooling model.

Usage

pFtest(x, ...)  
## S3 method for class 'plm'
pFtest(x, z, ...)  
## S3 method for class 'formula'
pFtest(x, data, ...)

Arguments

x an object of class "plm" or of class "formula",
z an object of class "plm",
data a data.frame,
... further arguments.

Details

For the plm method, the argument of this function is two plm objects, the first being a within model, the second a pooling model. The effects tested are either individual, time or twoways, depending on the effects introduced in the model.

Value

An object of class "htest".

Author(s)

Yves Croissant

See Also

plmtest for Lagrange multipliers tests of individuals and/or time effects.
Examples

data("Grunfeld", package="plm")
gi <- plm(inv ~ value + capital, data = Grunfeld, model = "pooling")
gt <- plm(inv ~ value + capital, data = Grunfeld,
effect = "time", model = "within")
gd <- plm(inv ~ value + capital, data = Grunfeld,
effect = "twoways", model = "within")
pFtest(gt, gi)
pFtest(gd, gi)
pFtest(inv ~ value + capital, data = Grunfeld, effect = "twoways")

pggls

Description

General FGLS estimators for panel data (balanced or unbalanced)

Usage

pggls(formula, data, subset, na.action, effect = c("individual", "time"),
       model = c("within", "random", "pooling", "fd"),
       index = NULL, ...)

## S3 method for class 'pggls'
summary(object, ...)

## S3 method for class 'summary.pggls'
print(x,digits = max(3,getOption("digits") - 2),
       width = getOption("width"),...)

Arguments

formula a symbolic description of the model to be estimated,
object, x an object of class pggls,
data a data.frame,
subset see lm,
na.action see lm,
effect the effects introduced in the model, one of "individual" or "time",
model one of "within", "pooling", "random" or "fd",
index the indexes, see plm.data,
digits digits,
width the maximum length of the lines in the print output,
... further arguments.
pggls is a function for the estimation of linear panel models by general feasible generalized least squares, either with or without fixed effects. General FGLS is based on a two-step estimation process: first a model is estimated by OLS (pooling), fixed effects (within) or first differences (fd), then its residuals are used to estimate an error covariance matrix for use in a feasible-GLS analysis. This framework allows the error covariance structure inside every group (if effect="individual", else symmetric) of observations to be fully unrestricted and is therefore robust against any type of intragroup heteroskedasticity and serial correlation. Conversely, this structure is assumed identical across groups and thus general FGLS estimation is inefficient under groupwise heteroskedasticity. Note also that this method requires estimation of \( T(T+1)/2 \) variance parameters, thus efficiency requires \( N >> T \) (if effect="individual", else the opposite). The model="random" and model="pooling" arguments both produce an unrestricted FGLS model as in Wooldridge, Ch.10, although the former is deprecated and included only for retrocompatibility reasons. If model="within" (the default) then a FEGLS (fixed effects GLS, see ibid.) is estimated; if model="fd" a FDGLS (first-difference GLS).

Value

An object of class c("pggls","panelmodel") containing:

- coefficients: the vector of coefficients,
- residuals: the vector of residuals,
- fitted.values: the vector of fitted.values,
- vcov: the covariance matrix of the coefficients,
- df.residual: degrees of freedom of the residuals,
- model: a data.frame containing the variables used for the estimation,
- call: the call,
- sigma: the estimated intragroup (or cross-sectional, if effect="time") covariance of errors.

Author(s)

Giovanni Millo

References


Examples

data("Produc", package = "plm")
zz <- pgpls(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, data = Produc, model = "pooling")
summary(zz)

Description

Generalized method of moments estimation for static or dynamic models with panel data.

Usage

pgmm(formula, data, subset, na.action, 
ext effect = c("twoways", "individual"),
model = c("onestep", "twosteps"),
collapse = FALSE,
lost.ts = NULL,
transformation = c("d", "ld"),
fsm = NULL, index = NULL, ...)
## S3 method for class 'pgmm'
summary(object, robust, time.dummies = FALSE, ...)
## S3 method for class 'summary.pgmm'
print(x, digits = max(3, getOption("digits") - 2),
width = getOption("width"), ...)

Arguments

formula a symbolic description for the model to be estimated. The preferred interface is now to indicate a multi-part formula, the first two parts describing the covariates and the gmm instruments and, if any, the third part the ‘normal’ instruments,
object, x an object of class "pgmm",
data a data.frame,
subset see lm,
na.action see lm,
effect the effects introduced in the model, one of "twoways" (the default) or "individual",
model one of "onestep" (the default) or "twosteps",
collapse if TRUE, the gmm instruments are collapsed,
lost.ts the number of lost time series: if NULL, this is automatically computed. Otherwise, it can be defined by the user as a numeric vector of length 1 or 2. The first element is the number of lost time series in the model in difference, the second one in the model in level. If the second element is missing, it is set to the first one minus one,
transformation  the kind of transformation to apply to the model: either "d" (the default value) for the “difference GMM” model or "ld" for the “system GMM”,

fsm  the matrix for the one step estimator: one of "I" (identity matrix) or "G" (= $D'D$ where $D$ is the first-difference operator) if transformation="d", one of "GI" or "full" if transformation="ld",

index  the indexes,

digits  digits,

width  the maximum length of the lines in the print output,

robust  if TRUE, robust inference is performed in the summary,

time.dummies  if TRUE, the estimated coefficients of time dummies are present in the table of coefficients,

...  further arguments.

Details

pgmm estimates a model for panel data with a generalized method of moments (GMM) estimator. The description of the model to estimate is provided with a multi-part formula which is (or which is coerced to) a Formula object. The first right-hand side part describe the covariates. The second one, which is mandatory, describes the gmm instruments. The third one, which is optional, describes the 'normal' instruments. By default, all the variables of the model which are not used as GMM instruments are used as normal instruments with the same lag structure as the one specified in the model.

\[
y \sim \text{lag}(y, 1:2) + \text{lag}(x1, 0:1) + \text{lag}(x2, 0:2) \mid \text{lag}(y, 2:99) + \text{lag}(x1, 0:1) + \text{lag}(x2, 0:2) + \text{lag}(y, 2:99) + \text{lag}(x1, 0:1) + \text{lag}(x2, 0:2)
\]

indicates that all lags from 2 of $y$ is used as gmm instruments.

transformation indicates how the model should be transformed for the estimation. "d" gives the "difference GMM" model (see Arellano and Bond (1991)), "ld" the "system GMM" model (see Blundell and Bond (1998)).

pgmm is an attempt to adapt GMM estimators available within the DPD library for GAUSS (see Arellano and Bond 1998) and Ox (see Doornik, Arellano and Bond 2006) and with the xtabond2 library for STATA (see Roodman 2009).

Value

An object of class c("pgmm", "panelmodel"), which has the following elements:

coefficients  the vector (or the list for fixed effects) of coefficients,

residuals  the vector of residuals,

fitted.values  the vector of fitted.values,

vcov  the covariance matrix of the coefficients,

df.residual  degrees of freedom of the residuals,

model  a list containing the variables used for the estimation for each individual,

W  a list containing the instruments for each individual (two lists in case of "sys–GMM"),
the weighting matrix for the one–step estimator,

the weighting matrix for the two–steps estimator,

the call.

It has print, summary and print.summary methods.

Author(s)

Yves Croissant

References


See Also

dynformula for dynamic formulas, sargan for Sargan tests and mtest for Arellano–Bond’s tests of multicollinearity.

Examples

data("EmplUK", package = "plm")

```r
## Arellano and Bond (1991), table 4b
z1 <- pgmm(log(emp) ~ lag(log(emp), 1:2) + lag(log(wage), 0:1) 
                      + log(capital) + lag(log(output), 0:1) | lag(log(emp), 2:99),
                      data = EmplUK, effect = "twoways", model = "twosteps")
summary(z1)

## Blundell and Bond (1998) table 4 (cf DPD for OX p.12 col.4)
z2 <- pgmm(log(emp) ~ lag(log(emp), 1)+ lag(log(wage), 0:1) + 
                        lag(log(capital), 0:1) | lag(log(emp), 2:99) + 
                        lag(log(wage), 2:99) + lag(log(capital), 2:99),
                        data = EmplUK, effect = "twoways", model = "onestep",
                        transformation = "1d")
summary(z2, robust = TRUE)
```

## Not run:
## Hausman-Taylor Estimator for Panel Data

The Hausman-Taylor estimator is an instrumental variable estimator without external instruments.

### Usage

```r
pht(formula, data, subset, na.action, model = c("ht", "am", "bmc"), index = NULL, ...)  
## S3 method for class 'pht'
summary(object, ...)  
## S3 method for class 'summary.pht'
print(x, digits = max(3,getOption("digits") - 2),  
    width = getOption("width"), subset = NULL, ...)
```

### Arguments

- **formula**: a symbolic description for the model to be estimated,
- **object, x**: an object of class "plm",
- **data**: a data.frame,
- **subset**: see `lm` for "plm", a character or numeric vector indicating a subset of the table of coefficient to be printed for "print.summary.plm",
- **na.action**: see `lm`,
- **model**: one of "ht" for Hausman-Taylor, "am" for Amemiya-MacCurdy and "bmc" for Breush-Mizon-Schmidt,
- **index**: the indexes,
- **digits**: digits,
- **width**: the maximum length of the lines in the print output,
- **...**: further arguments.
Details

pht estimates panels models using the Hausman-Taylor estimator. The model is specified a two-part formula, the second part containing the exogenous variables.

Value

An object of class c("pht", "plm", "panelmodel").

A "pht" object contains the same element as plm, with a further argument called varlist which describes the typology of the variables. It has summary and print.summary methods.

Author(s)

Yves Croissant

References


Examples

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

Description

Specification test for panel models.

Usage

phtest(x, ...)
## S3 method for class 'panelmodel'
phtest(x, x2, ...)
## S3 method for class 'formula'
phtest(x, data, ..., model = c("within","random"))
Arguments
- `x` an object of class "panelmodel" or "formula",
- `x2` an object of class "panelmodel",
- `model` a vector containing the names of two models,
- `data` a data.frame,
- `...` further arguments passed to `plm`

Details
The Hausman test is based on the difference of the vectors of coefficients of two different models.

Value
An object of class "htest".

Author(s)
Yves Croissant

References

Examples
```r
data("Gasoline", package = "plm")
form <- lgaspcar ~ lincomep + lrpmg + lcarpcap
wi <- plm(form, data = Gasoline, model = "within")
re <- plm(form, data = Gasoline, model = "random")
phtest(wi, re)
phtest(form, data = Gasoline)
```

---

**plm**  
*Panel Data Estimators*

Description
Linear models for panel data estimated using the `lm` function on transformed data.
Usage

```r
plm(formula, data, subset, na.action, effect = c("individual", "time", "twoways"),
     model = c("within", "random", "ht", "between", "pooling", "fd"),
     random.method = c("swar", "walhus", "amemiya", "nerlove", "kinla"),
     inst.method = c("bvk", "baltagi"), restrict.matrix = NULL,
     restrict.rhs = NULL, index = NULL, ...)
```

```r
## S3 method for class 'plm'
summary(object, .vcov = NULL, ...)
```

```r
## S3 method for class 'summary.plm'
print(x, digits = max(3, getOption("digits") - 2),
     width = getOption("width"), subset = NULL, ...)
```

```r
## S3 method for class 'plm'
plot(x, dx = 1, N = NULL, ...)
```

Arguments

- `formula`: a symbolic description for the model to be estimated,
- `object`: an object of class "plm".
- `data`: a data.frame,
- `subset`: see `lm` for "plm", a character or numeric vector indicating a subset of the table of coefficient to be printed for "print.summary.plm".
- `na.action`: see `lm`,
- `effect`: the effects introduced in the model, one of "individual", "time" or "twoways".
- `model`: one of "pooling", "within", "between", "random", "fd" and "ht",
- `random.method`: method of estimation for the variance components in the random effects model, one of "swar" (the default value), "amemiya", "walhus", "nerlove" and "kinla",
- `inst.method`: the instrumental variable transformation: one of "bvk" and "baltagi",
- `index`: the indexes,
- `restrict.matrix`: a matrix which defines linear restrictions on the coefficients,
- `restrict.rhs`: the right hand side vector of the linear restrictions on the coefficients,
- `.vcov`: a covariance matrix furnished by the user,
- `digits`: digits,
- `width`: the maximum length of the lines in the printed output,
- `dx`: the half-length of the individual lines for the plot method,
- `N`: the number of individual to plot,
- `...`: further arguments.

Details

`plm` is a general function for the estimation of linear panel models. It supports the following estimation methods: pooled OLS (model="pooling"), fixed effects ("within"), random effects ("random"), first-differences ("fd") and between ("between"). It supports unbalanced panels and two-way effects (although not with all methods).
For random effects models, 4 estimators of the transformation parameter are available: swar (Swamy and Arora), amemiya, walhus (Wallace and Hussain) and nerlove.

Instrumental variables estimation is obtained using two-part formulas, the second part indicating the instrumental variables used. This can be a complete list of instrumental variables or an update of the first part. If, for example, the model is $y \sim x_1 + x_2 + x_3$, with $x_1$ and $x_2$ endogenous and $z_1$ and $z_2$ external instruments, the model can be estimated with:

- $\text{formula} = y \sim x_1 + x_2 + x_3 \mid x_3 + z_1 + z_2$,
- $\text{formula} = y \sim x_1 + x_2 + x_3 \mid -x_1 - x_2 + z_1 + z_2$.

Balestra and Varadharajan–Krishnakumar’s or Baltagi’s method is used if \text{inst.method} = "bvk" or if \text{inst.method} = "baltagi".

The Hausman and Taylor estimator is computed if \text{model} = "ht".

Value

An object of class c("plm","panelmodel").

A "plm" object has the following elements:

- \text{coefficients} the vector of coefficients,
- \text{vcov} the covariance matrix of the coefficients,
- \text{residuals} the vector of residuals,
- \text{df.residual} degrees of freedom of the residuals,
- \text{formula} an object of class \text{pformula} describing the model,
- \text{model} a data.frame of class \text{pdata.frame} containing the variables used for the estimation: the response is in first position and the two indexes in the last positions,
- \text{ercomp} an object of class \text{ercomp} providing the estimation of the components of the errors (for random effects models only),
- \text{call} the call,

It has print, summary and print.summary methods.

Author(s)

Yves Croissant

References


Examples

```r
data("Produc", package = "plm")
zz <- plm(log(gsp) - log(pcap) + log(pc) + log(emp) + unemp,
       data = Produc, index = c("state","year"))
summary(zz)
```

---

**plm.data**

*Data Frame Special Format for Panel Data*

**Description**

This function transforms a data frame in a format suitable for using with the estimation functions of *plm*.

**Usage**

```r
plm.data(x, indexes = NULL)
```

**Arguments**

- `x` a *data.frame*,
- `indexes` a vector (of length one or two) indicating the (individual and time) indexes.

**Details**

indexes can be:

- a character string which is the name of the individual index variable, in this case a new variable called “time” containing the time index is added,
- an integer, the number of individuals in the case of balanced panel, in this case two new variables “time” and “id” containing the individual and the time indexes are added,
- a vector of two character strings which contains the names of the individual and of the time indexes.

**Value**

A *data.frame*. 
Author(s)
Yves Croissant

Examples

# There are 595 individuals
data("Wages", package = "plm")
Wages <- plm.data(Wages, 595)

# Gasoline contains two variables which are individual and time indexes
# The pdata.frame is called gas
data("Gasoline", package = "plm")
Gasoline <- plm.data(Gasoline, c("country", "year"))
summary(Gasoline)

# Hedonic is an unbalanced panel, townid is the individual index
data("Hedonic", package = "plm")
Hedonic <- plm.data(Hedonic, "townid")

plmtest

Lagrange Multiplier Tests for Panel Models

Description
Test of individual and/or time effects for panel models.

Usage

plmtest(x, ...)
## S3 method for class 'plm'
plmtest(x, effect = c("individual", "time", "twoways"),
        type = c("honda", "bp", "ghm", "kw"), ...)
## S3 method for class 'formula'
plmtest(x, data, ..., effect = c("individual", "time", "twoways"),
        type = c("honda", "bp", "ghm", "kw"))

Arguments

x an object of class "plm".
data a data.frame,
effect a character string indicating which effects are tested: individual effects ("individual"),
time effects ("time") or both ("twoways"),
type a character string indicating the test to be computed: "bp" for Breush-Pagan (1980), "ghm" for Gourieroux, Holly and Monfort (1982), "honda" for Honda (1985) or "kw" for King and Wu (1997),
... further arguments passed to plm.
Details

These Lagrange multiplier tests use only the residuals of the pooling model. The main argument of this function may be either a pooling model of class plm or an object of class formula.

Value

An object of class "htest".

Author(s)

Yves Croissant

References


See Also

pFtest for individual and/or time effects tests based on the within model.

Examples

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

Description

Mean Groups (MG), Demeaned MG and CCE MG estimators

Mean Groups (MG), Demeaned MG (DMG) and Common Correlated Effects MG (CCEMG) estimators for heterogeneous panel models, possibly with common factors (CCEMG)
pmg

Usage

pmg(formula, data, subset, na.action,
    model = c("mg", "cmg","dmg"),
    index = NULL, trend = FALSE, ...)

## S3 method for class 'pmg'
summary(object, ...)

## S3 method for class 'summary.pmg'
print(x,digits = max(3, getOption("digits") -
2), width = getOption("width"),...)

Arguments

formula a symbolic description of the model to be estimated,
object, x an object of class pmg,
data a data.frame,
subset see lm,
na.action see lm,
model one of c("mg", "cmg","dmg"),
index the indexes, see plm.data,
trend logical specifying whether an individual-specific trend has to be included,
digits
digits,
width the maximum length of the lines in the print output,
... further arguments.

Details

pmg is a function for the estimation of linear panel models with heterogeneous coefficients by the Mean Groups estimator. model="mg" specifies the standard Mean Groups estimator, based on the average of individual time series regressions. If model="dmg" the data are demeaned cross-sectionally, which is believed to reduce the influence of common factors (and is akin to what is done in homogeneous panels when model="within" and effect="time". Lastly, if model="cmy" then the CCEMG estimator is employed: this latter is consistent under the hypothesis of unobserved common factors and idiosyncratic factor loadings; it works by augmenting the model by cross-sectional averages of the dependent variable and regressors in order to account for the common factors, and adding individual intercepts and possibly trends.

Value

An object of class c("pmg","panelmodel") containing:

coefficients the vector of coefficients,
residuals the vector of residuals,
fitted.values the vector of fitted.values,
vcov the covariance matrix of the coefficients,
df.residual degrees of freedom of the residuals,
pmodel.response

model a data.frame containing the variables used for the estimation,
call the call,
sigma always NULL, sigma is here only for compatibility reasons (to allow using the same summary and print methods as pggls),
indcoef the matrix of individual coefficients from separate time series regressions.

Author(s)

Giovanni Millo

References


Examples

data("Produc", package = "plm")
## Mean Groups estimator
mgmod <- pmg(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, data = Produc)
summary(mgmod)
## demeaned Mean Groups
dmgmod <- pmg(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, data = Produc, model="dmg")
summary(dmgmod)
## Common Correlated Effects Mean Groups
ccemgmod <- pmg(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, data = Produc, model="cmg")
summary(ccemgmod)

pmodel.response A function to extract the model.response

Description

pmodel.response has several methods to conveniently extract the response of several objects.

Usage

pmodel.response(object, ...)
## S3 method for class 'data.frame'
model = c("pooling","within","Between",
          "between","mean","random","fd"),
effect = c("individual","time","twoways"),
ls = NULL,
theta = NULL, ...)
## S3 method for class 'pFormula'
pmodel.response(object, data, 
model = c("pooling","within","Between",  
"between","mean","random","fd"),
effect = c("individual","time","twoways"),
lhs = NULL,
theta = NULL, ...)

## S3 method for class 'plm'
pmodel.response(object, ...)

### Arguments

- **object**
  - an object of class "plm",
- **data**
  - a data.frame,
- **effect**
  - the effects introduced in the model, one of "individual", "time" or "twoways",
- **model**
  - one of "pooling", "within", "between", "random", "fd" and "ht",
- **theta**
  - the parameter for the transformation if model = "random",
- **lhs**
  - 
  - further arguments.

### Details

The model response is extracted (from a data.frame, a pformula or a plm object, and the transformation specified by effect and model is applied to it.

### Value

A numeric vector.

### Author(s)

Yves Croissant

---

**pooltest**

**Test of Poolability**

### Description

A Chow test for the poolability of the data.

### Usage

- `pooltest(x, ...)`
  - ## S3 method for class 'plm'
- `pooltest(x, z, ...)`
  - ## S3 method for class 'formula'
- `pooltest(x, data, ...)"
Arguments

x an object of class "plm".
z an object of class "pvcmb" obtained with model="within", data a data.frame, ...
... further arguments passed to plm.

Details

pooltest is an F test of stability (or Chow test) for the coefficients of a panel model. The estimated plm object should be a "pooling" model or a "within" model (the default); intercepts are assumed to be identical in the first case and different in the second case.

Value

An object of class "htest".

Author(s)

Yves Croissant

Examples

data("Gasoline", package = "plm")
form <- lgaspcar ~ lincomep + 1rpmg + 1carpcap
gasw <- plm(form, data = Gasoline, model = "within")
gasp <- plm(form, data = Gasoline, model = "pooling")
gasnp <- pvcmb(form, data = Gasoline, model = "within")
pooltest(gasw, gasnp)
pooltest(gasp, gasnp)

pooltest(form, data = Gasoline, effect = "individual", model = "within")
pooltest(form, data = Gasoline, effect = "individual", model = "pooling")

Description

A panel of 48 observations from 1970 to 1986

total number of observations : 816
observation : regional
country : United States

Usage

data(Produc)
Format

A data frame containing:

- **state**  the state
- **year**  the year
- **pcap**  private capital stock
- **hwy**  highway and streets
- **water**  water and sewer facilities
- **util**  other public buildings and structures
- **pc**  public capital
- **gsp**  gross state products
- **emp**  labor input measured by the employment in non-agricultural payrolls
- **unemp**  state unemployment rate

Source

Online complements to Baltagi (2001).

http://www.wiley.com/legacy/wileychi/baltagi/.

References


---

**pseries**  
*panel series*

Description

A class for panel series for which several useful computations are available.

Usage

- `between(x, ...)`
- `Between(x, ...)`
- `Within(x, ...)`
  ```r
  ## S3 method for class 'pseries'
  lag(x, k = 1, ...)
  ## S3 method for class 'pseries'
  diff(x, lag = 1, ...)
  ```
## S3 method for class 'pseries'
\texttt{as.matrix}(x, idbyrow = \texttt{TRUE}, \ldots)

## S3 method for class 'pseries'
\texttt{between}(x, effect = \texttt{c("individual", "time"), \ldots})

## S3 method for class 'pseries'
\texttt{Between}(x, effect = \texttt{c("individual", "time"), \ldots})

## S3 method for class 'pseries'
\texttt{Within}(x, effect = \texttt{c("individual", "time"), \ldots})

## S3 method for class 'pseries'
\texttt{summary}(object, \ldots)

## S3 method for class 'summary.pseries'
\texttt{print}(x, \ldots)

### Arguments

\texttt{x}, \texttt{object} \quad \text{a pseries or a summary.pseries object,}

\texttt{effect} \quad \text{the (individual or time) effect,}

\texttt{k, lag} \quad \text{the number of lags for the lag and diff methods,}

\texttt{idbyrow} \quad \text{if \texttt{TRUE}, the lines of the matrix are the individuals,}

\ldots \quad \text{further arguments.}

#### Details

A \texttt{pseries} is obtained when a series is extracted from a \texttt{pdata.frame} object. It consists on the original series with the index attribute of the \texttt{pdata.frame}. Specific transformations (within and between) are available and special \texttt{lag} and \texttt{diff} methods are provided.

#### Value

All these functions return an object of class \texttt{pseries}, except \texttt{between} which is a numeric vector.

#### Author(s)

Yves Croissant

#### Examples

```r
# Create first a \texttt{pdata.frame}
data("EmplUK", package = "plm")
Em <- \texttt{pdata.frame(EmplUK)}
# Then extract a serie, which becomes a pseries
z <- Em$\texttt{output}$class(z)
# obtain the matrix representation
\texttt{as.matrix}(z)
# compute the between, within transformation
\texttt{between}(z)
\texttt{Within}(z)
# Between replicate the values for each time observations
```
purtest

Between(z)
# compute the first and third lag, and the difference laged twice
lag(z)
lag(z, 3)
diff(z, 2)

purtest  
Unit root tests for panel data

Description

purtest implements several testing procedures that have been proposed to test unit root hypotheses with panel data.

Usage

purtest(object, data = NULL, index = NULL,
        test = c("levinlin", "ips", "madwu", "hadri"),
        exo = c("none", "intercept", "trend"),
        lags = c("SIC", "AIC", "Hall"), pmax = 10, Hcons = TRUE,
        q = NULL, dfcor = FALSE, fixedT = TRUE, ...)

## S3 method for class 'purtest'
print(x, ...)
## S3 method for class 'purtest'
summary(object, ...)
## S3 method for class 'summary.purtest'
print(x, ...)

Arguments

object, x  Either a 'data.frame' or a matrix containing the time series, a 'pseries' object, a formula, or the name of a column of a 'data.frame', or a 'pdata.frame' on which the test has to be computed; a 'purtest' object for the print and summary methods,
data  a 'data.frame' or a 'pdata.frame' object,
index  the indexes,
test  the test to be computed: one of levinlin for Levin, Lin and Chu (2002), ips for Im, Pesaran and Shin (2003), madwu for Maddala and Wu (1999), and hadri for Hadri (2000),
exo  the exogenous variables to introduce in the augmented Dickey-Fuller regressions: this can be nothing ('none'), individual intercepts ('intercept') or individual intercepts and trends ('trend'),
lags  the number of lags to be used for the augmented Dickey-Fuller regressions: either an integer (the number of lags for all time series), a vector of integers (one for each time series), or a character string for an automatic computation of the number of lags, based on either the AIC ('AIC'), the SIC ('SIC') or on Hall’s method ('Hall'),
purtest

\begin{itemize}
\item `pmax` maximum number of lags,
\item `hcons` a boolean indicating whether the heteroscedasticity-consistent test of Hadri should be computed,
\item `q` the bandwidth for the estimation of the long-run variance,
\item `dfcor` should the standard deviation of the regressions be computed using a degrees-of-freedom correction?
\item `fixedT` should the different ADF regressions be computed using the same number of observations?
\item ... further arguments.
\end{itemize}

Details

All these tests except 'hadri' are based on the estimation of augmented Dickey-Fuller regressions for each time series. A statistic is then computed using the t-statistic associated with the lagged variable.

The kind of test to be computed can be specified in several ways:

A formula/data interface (if \texttt{data} is a \texttt{data.frame}, an additional \texttt{index} argument should be specified); the formula should be of the form: `y~0`, `y~1`, or `y~trend` for a test with no exogenous variables, with an intercept, or with a time trend, respectively.

A \texttt{data.frame}, a \texttt{matrix}, a \texttt{pseries}: in this case, the exogenous variables are specified using the \texttt{exo} argument.

The Hadri statistic is the cross-sectional average of the individual KPSS statistics, standardized by their asymptotic mean and standard deviation.

Value

An object of class 'purtest': a list with the elements 'statistic' (a \texttt{htest} object), 'call', 'args', 'idres' (containing results from the individual regressions), and 'adjval' (containing the simulated means and variances needed to compute the statistics).

Author(s)

Yves Croissant

References


Examples

data("Grunfeld", package = "plm")
y <- data.frame(split(Grunfeld$inv, Grunfeld$firm))

purtest(y, pmax = 4, exo = "intercept", test = "madwu")

## same via formula interface
purtest(inv ~ 1, data = Grunfeld, index = "firm", pmax = 4, test = "madwu")

---

pvar Check for Cross-Sectional and Time Variation

Description

This function checks for each variable of a panel if it varies cross-sectionally and over time.

Usage

pvar(x, ...)
## S3 method for class 'data.frame'
pvar(x, index = NULL, ...)
## S3 method for class 'pdata.frame'
pvar(x, ...)
## S3 method for class 'pvar'
print(x, ...)

Arguments

x a data.frame,
index see plm.data,
... further arguments.

Details

pvar can be time consuming for "big" panels.

Value

An object of class pvar containing the following elements:

id.var a logical vector with TRUE values if the variable has individual variation, FALSE otherwise,
time.var a logical vector with TRUE values if the variable has time variation, FALSE otherwise,

Author(s)

Yves Croissant
Examples

# Gasoline contains two variables which are individual and time indexes
# and are the first two variables
data("Gasoline", package = "plm")
pvar(Gasoline)

# Hedonic is an unbalanced panel, townid is the individual index;
# the drop.index argument is passed to pdata.frame
data("Hedonic", package = "plm")
pvar(Hedonic,"townid", drop.index = TRUE)

# same using pdata.frame
Hed <- pdata.frame(Hedonic, "townid", drop.index = TRUE)
pvar(Hed)

pvcm

Variable Coefficients Models for Panel Data

Description

Estimators for random and fixed effects models with variable coefficients.

Usage

pvcm(formula, data, subset, na.action, effect = c("individual","time"),
      model = c("within","random"), index = NULL, ...)

## S3 method for class 'pvcm'
summary(object, ...)

## S3 method for class 'summary.pvcm'
print(x, digits = max(3,getOption("digits") -2),
      width =getOption("width"), ...)

Arguments

formula a symbolic description for the model to be estimated,
object, x an object of class "pvcm",
data a data.frame,
subset see lm,
na.action see lm,
effect the effects introduced in the model: one of "individual","time",
model one of "within","random",
index the indexes, see plm.data,
digits digits,
width the maximum length of the lines in the print output,
... further arguments.
Details

`pvc`m estimates variable coefficients models. Time or individual effects are introduced, respectively, if `effect="time"` or `effect="individual"` (the default value).

Coefficients are assumed to be fixed if `model="within"` and 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 (1970) model is estimated. It is a generalized least squares model which uses the results of the previous model.

Value

An object of class `c("pvc", "panelmodel")`, which has the following elements:

- `coefficients` the vector (or the list for fixed effects) of coefficients,
- `residuals` the vector of residuals,
- `fitted.values` the vector of fitted.values,
- `vcov` the covariance matrix of the coefficients,
- `df.residual` degrees of freedom of the residuals,
- `model` a `data.frame` containing the variables used for the estimation,
- `call` the call,
- `Delta` the estimation of the covariance matrix of the coefficients (random effect models only),
- `std.error` the standard errors for all the coefficients for each individual (within models only).

`pvc`m objects have `print`, `summary` and `print.summary` methods.

Author(s)

Yves Croissant

References


Examples

data("Produc", package = "plm")
zw <- `pvc`m(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, data = Produc, model = "within")
zr <- `pvc`m(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, data = Produc, model = "random")
**pwartest**  
Wooldridge Test for AR(1) Errors in FE Panel Models

**Description**

Test of serial correlation for (the idiosyncratic component of) the errors in fixed-effects panel models.

**Usage**

```r
pwartest(x, ...)  
## S3 method for class 'panelmodel'
pwartest(x, ...)  
## S3 method for class 'formula'
pwartest(x, data, ...)
```

**Arguments**

- `x` an object of class `formula` or of class `panelmodel`,
- `data` a `data.frame`,
- `...` further arguments to be passed on to `linearHypothesis` or to `vcovHC`.

**Details**

As Wooldridge (2003, Sec. 10.5.4) observes, under the null of no serial correlation in the errors, the residuals of a FE model must be negatively serially correlated, with $\text{cor}(\hat{u}_{it}, \hat{u}_{is}) = -1/(T-1)$ for each $t, s$. He suggests basing a test for this null hypothesis on a pooled regression of FE residuals on their first lag: $\hat{u}_{i,t} = \alpha + \delta \hat{u}_{i,t-1} + \eta_{i,t}$. Rejecting the restriction $\delta = -1/(T-1)$ makes us conclude against the original null of no serial correlation.

`pwartest` estimates the within model and retrieves residuals, then estimates an AR(1) pooling model on them. The test statistic is obtained by applying `linearHypothesis()` to the latter model to test the above restriction on $\delta$, setting the covariance matrix to `vcovHC` with the option `method="arellano"` to control for serial correlation.

Unlike the `pbgtest` and `pdwtest`, this test does not rely on large-$T$ asymptotics and has therefore good properties in “short” panels. Furthermore, it is robust to general heteroskedasticity.

**Value**

An object of class "htest".

**Author(s)**

Giovanni Millo


**References**


**See Also**

`pwfdtest`, `pdwtest`, `pbgtest`, `pbltest`, `pbsytest`.

**Examples**

```r
data("EmplUK", package = "plm")
pwtest(log(emp) ~ log(wage) + log(capital), data = EmplUK)
```

---

**pwfdtest**

*Wooldridge first-difference-based test for AR(1) errors in levels or first-differenced panel models*

**Description**

First-differencing-based test of serial correlation for (the idiosyncratic component of) the errors in either levels or first-differenced panel models.

**Usage**

```r
pwfdtest(x, ...)  
## S3 method for class 'panelmodel'  
 pwfdtest(x, ..., h0 = c("fd","fe"))  
## S3 method for class 'formula'  
 pwfdtest(x, data, ..., h0 = c("fd","fe"))
```

**Arguments**

- `x`: an object of class formula,
- `data`: a data frame,
- `h0`: the null hypothesis: one of "fd", "fe"
- `...`: further arguments to be passed on to `linearHypothesis` or to `vcovHC`

**Details**

As Wooldridge (2003, 10.6.3) observes, if the idiosyncratic errors in the model in levels are uncorrelated (which we label hypothesis "fe"), then the errors of the model in first differences (FD) must be serially correlated with \( \text{cor}(\hat{e}_{it}, \hat{e}_{is}) = -0.5 \) for each \( t,s \). If on the contrary the levels model’s errors are a random walk, then there must be no serial correlation in the FD errors (hypothesis "fd"). Both the fixed effects (FE) and the first-differenced (FD) estimators remain consistent under either assumption, but the relative efficiency changes: FE is more efficient under "fe", FD under "fd".
Wooldridge (ibid.) suggests basing a test for either hypothesis on a pooled regression of FD residuals on their first lag: 

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

Rejecting the restriction $$\delta = -0.5$$ makes us conclude against the null of no serial correlation in errors of the levels equation (“fe”). The null hypothesis of no serial correlation in differenced errors (“fd”) is tested in a similar way, but based on the zero restriction on $$\delta$$. Rejecting “fe” favours the use of the first-differences estimator and the contrary, although it is possible that both be rejected.

$pwfdtest$ estimates the fd model and retrieves residuals, then estimates an AR(1) pooling model on them. The test statistic is obtained by applying $\text{linearHypothesis()}$ to the latter model to test the relevant restriction on $$\delta$$, setting the covariance matrix to $\text{vcovHC}$ with the option $\text{method} = "\text{arellano}"$ to control for serial correlation.

Unlike the $\text{pbgtest}$ and $\text{pdwtest}$, this test does not rely on large-T asymptotics and has therefore good properties in "short" panels. Furthermore, it is robust to general heteroskedasticity. The "fe" version can be used to test for error autocorrelation regardless of whether the maintained specification has fixed or random effects (see Drukker, 2003).

**Value**

An object of class "htest".

**Author(s)**

Giovanni Millo

**References**


**See Also**

$\text{pdwtest, pbgtest, pwartest}$

**Examples**

```stata
data(EmplUK)
pwfdtest(log(emp) ~ log(wage) + log(capital), data = EmplUK)
pwfdtest(log(emp) ~ log(wage) + log(capital), data = EmplUK, h0 = "fe")
```
Wooldridge’s Test for Unobserved Effects in Panel Models

Description

Semi-parametric test for the presence of (individual or time) unobserved effects in panel models.

Usage

pwtest(x,...)
## S3 method for class 'panelmodel'
pwtest(x,...)
## S3 method for class 'formula'
pwtest(x, data, ...)

Arguments

x an object of class "formula",
data a data.frame,
... further arguments passed to plm.

Details

This semi-parametric test checks the null hypothesis of zero correlation between errors of the same group. Therefore, it has power both against individual effects and, more generally, any kind of serial correlation.

The test relies on large-N asymptotics. It is valid under error heteroskedasticity and departures from normality.

The above is valid if effect="individual", which is the most likely usage. If effect="time", symmetrically, the test relies on large-T asymptotics and has power against time effects and, more generally, against cross-sectional correlation.

Value

An object of class "htest".

Author(s)

Giovanni Millo

References

See Also

pbltest, pbgtest, pdwtest, pbsytest, pwartest, pwfdtest for tests for serial correlation in panel models. plmtest for tests for random effects.

Examples

data("Produc", package = "plm")
pwtest(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, data = Produc)

---

**r.squared**

*R squared for panel data*

Description

This function computes the R squared for plm objects. It allows to define on which transformation of the data the R squared has to be computed and which formula should be used.

Usage

```
r.squared(object, model = NULL, type = c('cor', 'rss', 'ess'), dfcor = FALSE)
```

Arguments

- **object**: an object of class "plm",
- **model**: on which transformation of the data the R-squared has to be computed? If null the transformation used to estimate the model is also used for the computation of the R squared.
- **type**: should the R-squared be computed using the residual sum of squares ("rss"), the explained sum of squares ("ess") or the coefficient of correlation between the fitted values and the response (cor).
- **dfcor**: if TRUE, the adjusted R squared is computed.

Value

a numerical value
**Hansen–Sargan Test of Overidentifying Restrictions**

**Description**

A test of overidentifying restrictions for models estimated by GMM.

**Usage**

```r
sargan(object, weights = c("twosteps", "onestep"))
```

**Arguments**

- `object`: an object of class "pgmm".
- `weights`: the weighting matrix to be used for the computation of the test.

**Details**

The Hansen–Sargan test calculates the quadratic form of the moment restrictions that is minimized while computing the GMM estimator. It follows asymptotically a chi-square distribution with number of degrees of freedom equal to the difference between the number of moment conditions and the number of coefficients.

**Value**

An object of class "htest".

**Author(s)**

Yves Croissant

**References**


**See Also**

`pgmm`

**Examples**

```r
data("EmplUK", package = "plm")
ar <- pgmm(log(emp) ~ lag(log(emp), 1:2) + lag(log(wage), 0:1) +
          lag(log(capital), 0:2) + lag(log(output), 0:2) | lag(log(emp), 2:99),
data = EmplUK, effect = "twoways", model = "twosteps")
sargan(ar)
```
Snmesp

Employment and Wages in Spain

Description
A panel of 738 observations from 1983 to 1990

*total number of observations*: 5904

*observation*: firms

*country*: Spain

Usage
data(Snmesp)

Format
A data frame containing:

- **firm**: firm index
- **year**: year
- **n**: log of employment
- **w**: log of wages
- **y**: log of real output
- **i**: log of intermediate inputs
- **k**: log of real capital stock
- **f**: real cash flow

Source
Journal of Business Economics and Statistics data archive:

References
Description

A panel of 125 observations from 1960 to 1985

*total number of observations*: 3250

*observation*: country

*country*: World

Usage

data(SumHes)

Format

A data frame containing:

- **year**: the year
- **country**: the country name (factor)
- **opec**: OPEC member?
- **com**: communist regime?
- **pop**: country’s population (in thousands)
- **gdp**: real GDP per capita (in 1985 US dollars)
- **sr**: saving rate (in percent)

Source

Online supplements to Hayashi (2000).

http://fhayashi.fc2web.com/datasets.htm

References


Beck and Katz Robust Covariance Matrix Estimators

Description

Unconditional Robust covariance matrix estimators a la Beck and Katz for panel models.

Usage

```r
## S3 method for class 'plm'
vcovBK(x, type = c("HC0", "HC1", "HC2", "HC3", "HC4"),
cluster = c("group", "time"),
diagonal = FALSE,
...)
```

Arguments

- `x`: an object of class "plm"
- `type`: one of "HC0", "HC1", "HC2", "HC3", "HC4".
- `cluster`: one of "group", "time".
- `diagonal`: a logical value specifying whether to force nondiagonal elements to zero.
- `...`: further arguments.

Details

vcovBK is a function for estimating a robust covariance matrix of parameters for a panel model according to the Beck and Katz (1995) method, a.k.a. Panel Corrected Standard Errors (PCSE), which uses an unconditional estimate of the error covariance across time periods (groups) inside the standard formula for coefficient covariance. Observations may be clustered either by "group" to account for timewise heteroskedasticity and serial correlation or by "time" to account for cross-sectional heteroskedasticity and correlation. It must be borne in mind that the Beck and Katz formula is based on N- (T-) asymptotics and will not be appropriate elsewhere.

The diagonal logical argument can be used, if set to `TRUE`, to force to zero all nondiagonal elements in the estimated error covariances; this is appropriate if both serial and cross-sectional correlation are assumed out, and yields a timewise- (groupwise-) heteroskedasticity-consistent estimator.

Weighting schemes are analogous to those in vcovHC in package sandwich and are justified theoretically (although in the context of the standard linear model) by MacKinnon and White (1985) and Cribari-Neto (2004) (see Zeileis (2004)).

The main use of vcovBK is to be an argument to other functions, e.g. for Wald-type testing: as vcov to coefTest(), waldtest() and other methods in the lmtest package; and as vcov to linearHypothesis() in the car package (see the examples). Notice that the vcov argument may be supplied a function (which is the safest) or a matrix (see Zeileis (2004), 4.1-2 and examples below).
Value

An object of class "matrix" containing the estimate of the covariance matrix of coefficients.

Author(s)

Giovanni Millo

References


Examples

```r
library(lmtest)
library(car)
data("Produc", package="plm")
zz <- plm(log(gsp)-log(pcap)+log(pc)+log(emp)+unemp, data=Produc, model="random")
## standard coefficient significance test
coeftest(zz)
## robust significance test, cluster by group
coeftest(zz, vcov=vcovBK)
## idem with parameters, pass vcov as a function argument
coeftest(zz, vcov=function(x) vcovBK(x, type="HC1"))
## idem, cluster by time period
coeftest(zz, vcov=function(x) vcovBK(x, type="HC1", cluster="time"))
## idem with parameters, pass vcov as a matrix argument
coeftest(zz, vcov=vcovBK(zz, type="HC1"))
## joint restriction test
waldtest(zz, update(zz, .-log(emp)-unemp), vcov=vcovBK)
## test of hyp.: 2*log(pc)=log(emp)
linearHypothesis(zz, "2*log(pc)=log(emp)", vcov=vcovBK)
```

---

**R**

**Robust Covariance Matrix Estimators**

**Description**

Robust covariance matrix estimators *a la White* for panel models.
Usage

```r
## S3 method for class 'plm'
vcovHC(x, method = c("arellano", "white1", "white2"),
    type = c("HC0", "HC1", "HC2", "HC3", "HC4"),
    cluster = c("group", "time"), ...)
## S3 method for class 'pgmm'
vcovHC(x, ...)
```

Arguments

- `x` an object of class "plm" which should be the result of a random effects or a within model or a model of class "pgmm".
- `method` one of "arellano", "white1", "white2".
- `type` one of "HC0", "HC1", "HC2", "HC3", "HC4".
- `cluster` one of "group", "time".
- `...` further arguments.

Details

`vcovHC` is a function for estimating a robust covariance matrix of parameters for a fixed effects or random effects panel model according to the White method (White 1980, 1984; Arellano 1987). Observations may be clustered by "group" ("time") to account for serial (cross-sectional) correlation.

All types assume no intragroup (serial) correlation between errors and allow for heteroskedasticity across groups (time periods). As for the error covariance matrix of every single group of observations, "white1" allows for general heteroskedasticity but no serial (cross-sectional) correlation; "white2" is "white1" restricted to a common variance inside every group (time period) (see Greene (2003), 13.7.1-2 and Wooldridge (2002), 10.7.2); "arellano" (see ibid. and the original ref. Arellano (1987)) allows a fully general structure w.r.t. heteroskedasticity and serial (cross-sectional) correlation.

Weighting schemes are analogous to those in `vcovHC` in package `sandwich` and are justified theoretically (although in the context of the standard linear model) by MacKinnon and White (1985) and Cribari-Neto (2004) (see Zeileis, 2004).

The main use of `vcovHC` is to be an argument to other functions, e.g. for Wald-type testing: as `vcov` to `coeftest()`, `waldtest()` and other methods in the `lmtest` package; and as `vcov` to `linearHypothesis()` in the `car` package (see the examples). Notice that the `vcov` argument allows to supply a function (which is the safest) or a matrix (see Zeileis (2004), 4.1-2 and examples below).

A special procedure for pgmm objects, proposed by Windmeijer (2005), is also provided.

Value

An object of class "matrix" containing the estimate of the asymptotic covariance matrix of coefficients.

Author(s)

Giovanni Millo \& Yves Croissant
References


Examples

```r
library(lmtest)
library(car)
data("Produc", package = "plm")
z <- plm(log(gsp) ~ log(pc) + log(emp) + unemp, data = Produc, model = "random")
## standard coefficient significance test
coeftest(z)
## robust significance test, cluster by group
## (robust vs. serial correlation)
coeftest(z, vcov=vcovHC)
## idem with parameters, pass vcov as a function argument
coeftest(z, vcov=function(x) vcovHC(x, method="arellano", type="HC1"))
## idem, cluster by time period
## (robust vs. cross-sectional correlation)
coeftest(z, vcov=function(x) vcovHC(x, method="arellano", type="HC1", cluster="group"))
## idem with parameters, pass vcov as a matrix argument
coeftest(z, vcov=vcovHC(z, method="arellano", type="HC1"))
## joint restriction test
waldtest(z, update(z, .-log(emp)-unemp), vcov=vcovHC)
## test of hyp.: 2*log(pc)=log(emp)
linearHypothesis(z, "2*log(pc)=log(emp)", vcov=vcovHC)
## Robust inference for GMM models
data("EmplUK", package="plm")
ar <- pgmm(dynformula(log(emp) ~ log(wage) + log(capital) + log(output), list(2, 1, 2, 2)), data = EmplUK, effect = "twoways", model = "twosteps", gmm.inst = ~ log(emp),
```

```
lag.gmm = list(c(2, 99))
rv <- vcovHC(ar)
mtest(ar, order = 2, vcov = rv)
```

---

**vcovSCC**

*Driscoll and Kraay (1998) Robust Covariance Matrix Estimator*

**Description**

Nonparametric robust covariance matrix estimators *a la Driscoll and Kraay* for panel models with cross-sectional and serial correlation.

**Usage**

```r
## S3 method for class 'plm'
vcovSCC(x, type = c("HC0", "HC1", "HC2", "HC3", "HC4"),
        maxlag=NULL, ...)
```

**Arguments**

- `x`: an object of class "plm"
- `type`: one of "HC0", "HC1", "HC2", "HC3", "HC4".
- `maxlag`: either NULL or a positive integer specifying the maximum lag order before truncation
- `...`: further arguments.

**Details**

`vcovSCC` is a function for estimating a robust covariance matrix of parameters for a panel model according to the Driscoll and Kraay (1998) method, which is consistent with cross-sectional and serial correlation in a T-asymptotic setting and irrespective of the N dimension. The use with random effects models is undocumented.

Weighting schemes are analogous to those in `vcovHC` in package sandwich and are justified theoretically (although in the context of the standard linear model) by MacKinnon and White (1985) and Cribari-Neto (2004) (see Zeileis, 2004).

The main use of `vcovSCC` is to be an argument to other functions, e.g. for Wald-type testing: as `vcov` to `coeftest()`, `waldtest()` and other methods in the `lmtest` package; and as `vcov` to `linearHypothesis()` in the `car` package (see the examples). Notice that the `vcov` argument may be supplied a function (which is the safest) or a matrix (see Zeileis (2004), 4.1-2 and examples below).

**Value**

An object of class "matrix" containing the estimate of the covariance matrix of coefficients.
Wages

Author(s)
Giovanni Millo, partially ported from Daniel Hoechle’s Stata code

References

Examples

```r
library(lmtest)
library(car)
data("Produc", package="plm")
zz <- plm(log(gsp)-log(pcap)+log(pc)+log(emp)+unemp, data=Produc, model="pooling")
## standard coefficient significance test
coeftest(zz)
## SCC robust significance test, default
coeftest(zz, vcov=vcovSCC)
## idem with parameters, pass vcov as a function argument
coeftest(zz, vcov=function(x) vcovSCC(x, type="HC1", maxlag=4))
## joint restriction test
waldtest(zz, update(zz, .~.-log(emp)-unemp), vcov=vcovSCC)
## test of hyp: 2*log(pc)=log(emp)
linearHypothesis(zz, "2*log(pc)=log(emp)", vcov=vcovSCC)
```

---

| Wages | Panel Data of Individual Wages |

Description

A panel of 595 observations from 1976 to 1982, taken from the Panel Study of Income Dynamics (PSID).

*total number of observations*: 4165

*observation*: individuals

*country*: United States

Usage

```r
data(Wages)
```
Format

A data frame containing:

- **exp** years of full-time work experience.
- **wks** weeks worked.
- **bluecol** blue collar?
- **ind** works in a manufacturing industry?
- **south** resides in the south?
- **smsa** resides in a standard metropolitan statistical area?
- **married** married?
- **sex** a factor with levels "male", "female"
- **union** individual’s wage set by a union contract?
- **ed** years of education.
- **black** is the individual black?
- **lwage** logarithm of wage.

Source

Online complements to Baltagi (2001).

http://www.wiley.com/legacy/wileychi/baltagi/

References


Index

*Topic attribute
  index.plm, 14
  pdim, 27
  plm.data, 42
  pvar, 53
*Topic classes
dynformula, 6
  pdata.frame, 25
  pFormula, 29
  pseries, 49
*Topic datasets
  Cigar, 3
  Crime, 5
  EmplUK, 7
  Gasoline, 11
  Grunfeld, 12
  Hedonic, 13
  LaborSupply, 15
  Males, 16
  Produc, 48
  Snnesp, 62
  SumHes, 63
  Wages, 69
*Topic htest
cipstest, 4
  mtest, 17
  pbgtest, 18
  pblltest, 19
  pbsytest, 20
  pcdtest, 23
  pdwtest, 28
  pFtest, 31
  phtest, 38
  plmtest, 43
  pooltest, 47
  purtest, 51
  pwartest, 56
  pwfddtest, 57
  pwtest, 59
  r.squared, 60
  sargan, 61
*Topic manip
  pmodel.response, 46
*Topic regression
ercomp, 8
  fixef.plm, 9
  pcce, 22
  pggls, 32
  pgmm, 34
  pht, 37
  plm, 39
  pmg, 44
  pvcm, 54
  vcovBK, 64
  vcovHC, 65
  vcovSCC, 68
$.pdata.frame (pdata.frame), 25
[[.pdata.frame (pdata.frame), 25
as.data.frame.pdata.frame
  (pdata.frame), 25
as.Formula.pFormula (pFormula), 29
as.matrix.pseries (pseries), 49
  Between (pseries), 49
  between (pseries), 49
  Between.pseries (pseries), 49
  between.pseries (pseries), 49
  Cigar, 3
  cipstest, 4
  Crime, 5
  diff.pseries (pseries), 49
dynformula, 6, 36
  EmplUK, 7
ercomp, 8
fixef (fixef.plm), 9
fixef.plm, 9
formula.dynformula (dynformula), 6

Gasoline, 11
Grunfeld, 12

has.intercept (plm), 39
Hedonic, 13

index (index.plm), 14
index.plm, 14

LaborSupply, 15
lag.pseries (pseries), 49
lm, 8, 34, 37, 40

Males, 16
model.frame.pFormula (pFormula), 29
model.matrix.pFormula (pFormula), 29
mtest, 17, 36

pbgtest, 18, 21, 29, 56, 57, 60
pbltest, 19, 19, 21, 29, 57, 60
pbsytest, 19, 20, 29, 57, 60
pce, 22
pcdtest, 23
pdata.frame, 25, 27
pdim, 27
pdwtest, 19–21, 28, 56, 57, 60
pFormula, 29
pftest, 31, 44
pggls, 32
pgmm, 17, 34, 61
pht, 37
phtest, 38
plm, 9, 10, 15, 39
plm.data, 22, 22, 42, 45, 53, 54
plmtest, 21, 31, 43, 60
plot.plm (plm), 39
pvar, 53
pvcovHC (vcovHC), 65
pvtartest, 19, 20, 29, 56, 60
pwftest, 19, 20, 29, 57, 57, 60
pwtest, 59
r.squared, 60
sargan, 36, 51
Snnesp, 62
SumHes, 63
summary.fixef (fixef.plm), 9
summary.pce (pce), 22
summary.pggls (pggls), 32
summary.pgmm (pgmm), 34
summary.pht (pht), 37
summary.plm (plm), 39
summary.pmg (pmg), 44
summary.pseries (pseries), 49
summary.purtest (purtest), 51
summary.pvcovcm (pvcovcm), 54
vcovBK, 64
vcovHC, 65
vcovSCC, 68

Wages, 69
Within (pseries), 49