Package ‘splm’

September 28, 2017

Title Econometric Models for Spatial Panel Data
Version 1.4-9
Date 2017-09-10
Author Giovanni Millo [aut, cre],
Gianfranco Piras [aut]
Maintainer Giovanni Millo <giovanni.millo@generali.com>
Description ML and GM estimation and diagnostic testing of econometric models for spatial panel data.
Depends R (>= 2.12.0), spdep
Imports plm, maxLik, MASS, bdsmatrix, ibdreg, nlme, Matrix, spam, methods
License GPL-2
LazyLoad yes
Repository CRAN
Repository/R-Forge/Project splm
Repository/R-Forge/Revision 216
Repository/R-Forge/DateTimeStamp 2017-09-10 18:23:58
Date/Publication 2017-09-28 20:51:53 UTC
NeedsCompilation no

R topics documented:

- bsktest
- bsktest
- effects.splm
- Insurance
- itaww
- listw2dgCMatrix
- print.splm
- RiceFarms
- riceww
Baltagi, Song, Jung and Koh LM test for spatial panels

**Description**

Baltagi, Song, Jung and Koh joint or conditional LM test for spatial error correlation or serial correlation sub spatial, serial correlation and random effects in panel models

**Usage**

```r
bsjktest(x, ...)
```

**Arguments**

- `x` an object of class formula
- `data` a data.frame or pdata.frame containing the variables in the model
- `index` either NULL (default) or a character vector to identify the indexes among the columns of the data.frame
- `listw` either a matrix or a listw representing the spatial structure
- `test` one of c("C.1", "C.2", "C.3", "J"), the test to be performed.
- `...` additional arguments to be passed

**Value**

an object of class htest

**Author(s)**

Giovanni Millo
References


See Also

bsktest

Examples

```r
data(Produc, package="plm")
data(usaww)
fm <- log(gsp)~log(pcap)+log(pc)+log(emp)+unemp
bsktest(fm, data=Produc, listw = usaww, test="C.1")
```

bsktest  

*Baltagi, Song and Koh LM test for spatial panels*

Description

Baltagi, Song and Koh marginal or conditional LM test for spatial error correlation or random effects in panel models

Usage

```r
bsktest(x,...)

## S3 method for class 'formula'
bsktest(x, data, index=NULL, listw, test=c("LMH","LM1","LM2","CLMlambda","CLMmu"),
standardize=FALSE, method = "eigen", ...)
```

Arguments

- **x**: a formula
- **data**: a `data.frame` or `pdata.frame` containing the variables in the model
- **index**: either `NULL` (default) or a character vector to identify the indexes among the columns of the `data.frame`
- **listw**: a `listw` representing the spatial structure
- **test**: one of `c("LMH","LM1","LM2","CLMlambda","CLMmu"))`, the test to be performed
- **standardize**: whether to standardize the test statistic or not (applies only to LM1 and LM2)
- **method**: select a method for ML in "CLMmu". the default is "eigen"
- **...**: additional arguments to be passed

Value

an object of class `htest`
Author(s)
Gianfranco Piras

References

See Also
sphtest

Examples
```r
data(Produc, package="plm")
data(usaww)
f m <- log(gsp)-log(pcap)+log(pc)+log(emp)+unemp
bsktest(f m, data=Produc, listw = mat2listw(usaww),
       test="LM1")
```

---

**effects.splm**  
*method for extracting fixed effects*

Description
Methods used for extracting fixed effects from objects of class *splm* where type is one of "fixed effects lag" or "fixed effects error"

Usage
```r
## S3 method for class 'splm'
effects(object,...)
```

Arguments
- **object**  
an object of class 'splm'
- **...**  
additional arguments to be passed over

Details
If the argument object is not of class *splm* the function will terminate with an error.
If the argument object is of class *splm* but type is not one of "fixed effects lag" or "fixed effects error", the function will terminate with an error.
Insurance

Value

An object of class effects.splm

res a list whose elements are various type of fixed effects and the intercept (when present)

Author(s)

Gianfranco Piras

References


See Also

spml summary.effects.splm

Examples

data(Produc, package = "plm")
data(usaww)
fm <- log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp
err <- spml(fm, data = Produc, listw = mat2listw(usaww), model="within")
summary(err)
eff <- effects(err)
print(eff)

Insurance

*Insurance consumption across Italian provinces, 1998-2002*

Description

A panel of 103 observations

number of observations : 515

observation : provinces

country : Italy

Usage

data(Insurance)
Format

A dataframe containing:

- **code** the province code according to Istat
- **year** the year of observation
- **ppcd** real per capita premiums in 2000 euros, non-life insurance excluding mandatory motor third-party liability
- **rgdp** real per-capita GDP
- **bank** real per-capita bank deposits
- **den** population density per square Km
- **rirs** real interest rate on lending to families and small enterprises
- **agen** density of insurance agencies per 1000 inhabitants
- **school** share of people with second grade schooling or more
- **vaagr** share of value added, agriculture
- **fam** average number of family members
- **inef** judicial inefficiency index: average years to settle first degree of civil case
- **trust** survey result to the question "do you trust others?"
- **dXX** year dummies
- **NorthWest** macroregional dummy
- **NorthEast** macroregional dummy
- **Centre** macroregional dummy
- **South** macroregional dummy
- **Islands** macroregional dummy (Sicily and Sardinia)

Author(s)

Giovanni Millo

Source

Spatial weights matrix - Italian provinces

Description


Usage

data(itaww)

Format

A matrix with elements different from zero if province i and j are neighbors. Weights are row-standardized. Messina and Reggio Calabria, divided by the Messina Strait, are considered neighbours.

Author(s)

Giovanni Millo

listw2dgCMatrix

Interface between Matrix class objects and weights list

Description

Interface between Matrix class objects and weights list

Usage

listw2dgCMatrix(listw, zero.policy = NULL)

Arguments

listw a listw object created for example by nb2listw

zero.policy See lagsarlm for details

Value

Matrix class object: a sparse Matrix

Author(s)

Gianfranco Piras
print.splm

print method for class splm

Description

Method to print objects of class summary.splm and splm

Usage

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

Arguments

x an object of class splm
digits minimal number of significant digits, see print.default
... additional arguments to be passed

Details

The summary function summary.splm returns an objects of class 'splm' organized in a coefficient matrix.
Also a matrix for the error components, or the spatial coefficients will be generated depending on the estimated model.

Author(s)

Giovanni Millo, Gianfranco Piras

See Also

spml, spgm

Examples

data(Produc, package = "plm")
data(usaww)
spremod<-spml(log(gsp)~log(pc)+log(pc)+log(emp)+unemp, data=Produc,
listw = mat2listw(usaww), model="random", lag=TRUE, spatial.error="none")
summary(spremod)
Description

yearly observations of 171 farms

number of observations : 1026
country : Indonesia
economic topic : producer behavior
econometrics topic : error component

Usage

data(RiceFarms)

Format

A dataframe containing :

id the farm identifier
time the growing season
size the total area cultivated with rice, measured in hectares
status land status, on of 'owner' (non sharecroppers, owner operators or leaseholders or both), 'share' (sharecroppers), 'mixed' (mixed of the two previous status)
varieties one of 'trad' (traditional varieties), 'high' (high yielding varieties) and 'mixed' (mixed varieties)
bimas bIMAS is an intensification program ; one of 'no' (non-bimas famer), 'yes' (bimas farmer) or 'mixed' (part but not all of farmer's land was registered to be in the bimas program)
seed seed in kilogram
urea urea in kilogram
phosphate phosphate in kilogram
pesticide pesticide cost in Rupiah
pseed price of seed in Rupiah per kg
purea price of urea in Rupiah per kg
pphosph price of phosphate in Rupiah per kg
hiredlabor hired labor in hours
famlabor family labor in hours
totlabor total labor (excluding harvest labor)
wage labor wage in Rupiah per hour
goutput gross output of rice in kg
noutput net output, gross output minus harvesting cost (paid in terms of rice)
price price of rough rice in Rupiah per kg
region one of 'wargabinangun','langan','gunungwangi','malausma','sukaambit','ciwangi'
Source


References


---

**riceww**

*Spatial weights matrix of Indonesian rice farms*

---

**Description**

Spatial weights matrix of the 171 farms in the Indonesian Rice Farming example. Farms in the same village (out of six) are considered contiguous.

**Usage**

data(riceww)

**Format**

A matrix with elements different from zero if farms i and j are neighbors. Farms are considered neighbors if in the same village. Weights are row-standardized.

**Author(s)**

Giovanni Millo, data provided by Yves Croissant

---

**rwtest**

*Randomization-based test of spatial dependence for panel models*

---

**Description**

Randomization-based test of spatial dependence for panel models, robust to global dependence induced by common factors and to persistence (serial correlation) in the data.
### Usage

```r
rwtest(x, ...)  
## S3 method for class 'formula'

rwtest(x, data, w, index = NULL, model = NULL, 
replications = 99, seed=NULL, order=1, 
mc=1, test = c("rho", "cd", "sclm"), 
alternative=c("twosided", "onesided", 
"symmetric"), ...)

## S3 method for class 'panelmodel'

rwtest(x, w, replications = 99, seed=NULL, 
order=1, mc=1, 
test = c("rho", "cd", "sclm"), 
alternative=c("twosided", "onesided", 
"symmetric"), ...)

## S3 method for class 'pseries'

rwtest(x, w, replications = 99, seed=NULL, 
order=1, mc=1, 
test = c("rho", "cd", "sclm"), 
alternative=c("twosided", "onesided", 
"symmetric"), ...)
```

### Arguments

- **x**
  - An object of class `formula`, `panelmodel`, or `pseries` (depending on the respective interface) describing the model to be tested.

- **data**
  - A data frame.

- **w**
  - An $n \times n$ matrix describing proximity between individuals, 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. Only the lower triangular part (without diagonal) of $w$ after coercing by `as.logical()` is evaluated for neighbouring information (but $w$ can be symmetric). See also `Details` and `Examples`.

- **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.

- **replications**
  - The number of Monte Carlo randomizations of the neighbourhood matrix (default: 99).

- **seed**
  - The optional random seed.

- **order**
  - The order of neighbourhood to test for.

- **mc**
  - The number of parallel threads to execute; defaults to 1 (serial execution); is limited to the number of execution cores actually available, and depends on operating system support.

- **test**
  - The type of test statistic to be returned. One of
    - "rho" for the average correlation coefficient,
• "cd" for Pesaran’s CD statistic, or
• "sc1m" for the scaled version of Breusch and Pagan’s LM statistic,

alternative the alternative hypothesis for the test, defaulting to (asymmetric) twosided.

... further arguments to be passed on to plm, such as e.g. effect or random.method

Details

This test is meant as a generalization of Pesaran’s spatial dependence test "CD(p)" for robustness against global dependence (perhaps of the factor type) and persistence in the data, both of which the original test does not tolerate.

The procedure can be applied to model residuals as well as to individual pseries. See the comments in pcdtest as for the different methods.

Space is defined supplying a proximity matrix (elements coercible to logical) with argument \(w\) which provides information on whether any pair of individuals are neighbours or not. If \(order=1\), only first-order neighbouring pairs will be used in computing the test; else, \(w\) will be transformed in the neighbourhood matrix of the appropriate order. 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 spdep’s function nb2mat before using.

Notice that the \(BrhoB\) and \(BcdB\) tests are permutationally equivalent.

The test is suitable also for unbalanced panels.

The test on a pseries is the same as a test on a pooled regression model of that variable on a constant, i.e. \(rwtest(some_pseries)\) is equivalent to \(rwtest(plm(some_var ~ 1, data = some_pdata.frame, model = \text{"pooling"})\) and also equivalent to \(rwtest(some_var ~ 1, data = \text{some_data}), where some_var is the variable name in the data which corresponds to some_pseries.

Value

An object of class "htest".

Author(s)

Giovanni Millo

References


Examples

data(Produc, package = "plm")
data(usaww)
fm <- log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp
## test on heterogeneous model (separate time series regressions)
rwtest(fm, data = Produc, w=usaww, index = c("state", "year"))
## slg

### Spatial lag operator

**Description**

Spatial lagging method for vectors or pseries objects.

**Usage**

```r
# S3 method for class 'pseries'
slg(x, listw, maxlag, ...)
```

**Arguments**

- `x` an object of class `pseries`
- `listw` an object of class `listw`
- `maxlag` the spatial lag order (including lower)
- `...` additional arguments to be passed

**Value**

A `pseries`

**Author(s)**

Giovanni Millo

**Examples**

```r
data(Produc, package="plm")
data(usaaww)
usalw <- mat2listw(usaaww)
fm <- log(gsp)-log(pcap)+log(pc)+log(emp)+unemp+slg(log(pcap),
       listw=usalw)
slxmod <- spreml(fm, data=Produc, w = usaaww,
       model="pooling", lag=FALSE, errors="ols")
```
Locally robust panel LM tests for spatial lag (error) correlation sub spatial error (lag) correlation in panel models

Usage

```r
slmtest(x, ...)
## S3 method for class 'formula'
slmtest(formula, data, listw, model="pooling",
    test=c("lme","lml","rlme","rlml"), index=NULL, ...)
## S3 method for class 'plm'
slmtest(x, listw,
    test=c("lme","lml","rlme","rlml"), ...)
```

Arguments

- `formula` an object of class `formula`
- `data` a `data.frame` or `pdata.frame` containing the variables in the model
- `x` an object of class `plm`
- `listw` either a `matrix` or a `listw` representing the spatial structure
- `model` a character value specifying the transformation to be applied before calculating the test. Defaults to "pooling" (no transformation).
- `test` one of `c("lme","lml","rlme","rlml"),` the test to be performed.
- `index` either NULL (default) or a character vector to identify the indexes among the columns of the `data.frame`
- `...` additional arguments to be passed

Details

This tests are panel versions of the locally robust LM tests of Anselin et al. (1996), based on a pooling assumption: i.e., they do not allow for any kind of individual effect. Therefore it is advisable to employ a within transformation whenever individual effects cannot be ruled out.

It must be kept in mind that these locally robust procedures have been designed for situations in which the "other" effect is not of substantial magnitude, and can behave suboptimally otherwise.

Four tests are available to be chosen through the `test` argument: "lml" for "LM lag" and, respectively, "lme" for "LM error" are the standard, non-robust versions, obtained simply pooling the cross-sectional versions; "rlml" and "rlme" are, respectively, the locally robust test for lag, allowing for a spatial error; and for error, allowing for a spatial lag.

The `model` argument, specified according to the standards of `plm`, is passed on internally and employed to determine the panel data transformation to be applied before calculating the test. Defaults to "pooling" (no transformation).
**Value**

an object of class `htest`

**Author(s)**

Giovanni Millo

**References**


**Examples**

```r
data(Produc, package="plm")
data(usaww)
fm <- log(gsp)-log(pcap)+log(pc)+log(emp)+unemp
## robust LM test for spatial error sub spatial lag
## model on original data, pooling hypothesis
slmtest(fm, data=Produc, listw = usaww, test="rlme")
## model on within-transformed (time-demeaned) data,
## eliminates individual effects
slmtest(fm, data=Produc, listw = usaww, test="rlme",
model="within")
```

---

**Description**

GM estimation of panel data models with spatially correlated errors components of the form:

\[
y_N(t) = \lambda W y + X_N(t)\beta + u_N(t)
\]

\[
u_N(t) = \rho W u_N(t) + \epsilon(t)
\]

\[
\epsilon_N = (\epsilon T \otimes I_N)\mu_N + \nu_N
\]

where \(\rho\), and the variance components \(\sigma^2_\mu\) and \(\sigma^2_\nu\) are estimated by GM, and the model coefficients by a feasible GLS estimator. The model can also include additional (other than the spatial lag) endogenous variables.
Usage

```r
spgm(formula, data=list(), index=NULL, listw =NULL, listw2 = NULL, model=c("within","random"), lag = FALSE, spatial.error=TRUE, moments = c("initial", "weights", "fullweights"), endog = NULL, instruments= NULL, lag.instruments = FALSE, verbose = FALSE, method = c("w2sls", "b2sls", "g2sls", "ec2sls"), control = list(), optim.method = "nlminb", pars = NULL)
```

Arguments

- **formula**: a description of the model to be fit. The details of model specification are given for `lm`
- **data**: an object of class `data.frame` or `pdata.frame`. An optional data frame containing the variables in the model. When the object is a `data.frame`, the first two columns may contain the indexes. See `index`
- **index**: if not NULL (default), a character vector to identify the indexes among the columns of the `data.frame`
- **listw**: an object of class `listw`, `matrix`, or `Matrix`
- **listw2**: an object of class `listw`, `matrix`, or `Matrix`. Only if both `lag` and `spatial.error` are both `TRUE`
- **model**: One of "within" or "random". The assumption made on the individual effects
- **lag**: if `TRUE` a spatial lag of the dependent variable is added to the regression equation
- **spatial.error**: a logic vector. If `TRUE` the spatial autoregressive error term is added to the model and an estimate for $\rho$ is produced
- **moments**: "initial" (default) defines the set of GM estimator to be used. Alternatives are "weights" and "fullweights" (See Details)
- **endog**: additional endogenous variables. Default NULL. If not NULL should be specified as a formula with no dependent variable (endog = ~ x1 + x2). Note the ~ before the expression.
- **instruments**: external instruments. Default NULL. If not NULL should be specified as a formula with no dependent variable (instruments = ~ x1 + x2). Note the ~ before the expression.
- **lag.instruments**: should the external instruments be spatially lagged?
- **verbose**: default `FALSE`, If `TRUE` reports function values during optimization
- **method**: One of "w2sls", "b2sls", "g2sls", "ec2sls". (See Details)
- **control**: a list of control parameters for the optimization
- **optim.method**: default set to "nlminb". or optionally a method passed to `optim` to use an alternative optimizer.
- **pars**: initial values of the parameter `rho` and `sigmav`. The default for `rho` is to start from a regression of the spatially lagged residuals on the residuals (depending on the model). For `sigmav` the starting value is the variance of the residuals (again this depends on the model).
Details

The function is a very general interface to estimate various nested specifications of the general model including additional endogenous variables described above. When both spatial.error and lag are FALSE the model reduces to a panel data model with an additional endogeneous variable. The function then uses ivsplm to perform the Instrumental Variables and two-stage least squares for panel data model. method = "w2s1s" corresponds to the fixed effects estimator, method = "b2s1s" to the between effects model, method = "g2s1s" to the GLS random effects model, and method = "ec2s1s" to the Baltagi’s EC2SLS.

When spatial.error is TRUE and lag is FALSE the model is one with spatially autocorrelated error components. If effects is "random", the Kapoor et al. (2007) GM estimator is performed and the residuals in the first step come from an OLS regression. When moments is "initial", the initial estimator is calculated. This first set of GM estimators is based only on a subset of the moments conditions and assigns equal weights to each of them. When moments is "fullweights", the second set of GM estimators is calculated. This estimator is based on the full set of moments conditions. It also involves the expression for the variance covariance matrix of the sample moments calculated under the assumption of normally distributed innovations. The calculation of the trace terms in the expression of the variance covariance matrix of the sample moments uses codes from the Matrix package. When moments is "weights", the third set of GM estimator is used. This is motivated by computational issues. The procedure is analogous to the second one but uses a simplified expression for the variance covariance matrix of the sample moments. If effects is "fixed", the initial estimator is a within estimator and the moments conditions of Kapoor et al. (2007) are modified accordingly.

Finally, when both spatial.error and lag are TRUE the complete model is estimated (with or without additional endogenous variables). OLS residuals are no longer consistent because of the spatially lagged dependent variable. If effects is "random", two initial estimators are computed: a within two-stage least squares and a between two stage least squares. The two sets of corresponding residuals are used in the spatial generalized moments estimator (GM) where the moments conditions of Kapoor et al. (2007) are again modified accordingly. If effects is "fixed", the initial estimator is a within two stage least squares estimator and the moments conditions of Kapoor et al. (2007) are modified accordingly.

Note that for the random effects models, $\sigma_2^2$ is not reported. $\sigma_1^2$ is reported instead. However, a value for $\sigma_2^2$ can easily be obtained from:

$$\sigma_1^2 = \sigma_2^2 + T\sigma_\mu^2$$

The function also produces an estimate for $\theta$ which is a function of the variance components.

Value

An object of class "splm".

coefficients GLS coefficients estimate of the model parameters
vcov the variance covariance matrix of the estimated coefficients
residuals the GLS residuals
fitted.values difference between response variable and residuals
sigma2 GLS residuals variance
type 'random effect GM'
rho  a vector including the spatial parameter and the variance components (see Details)
model  the matrix of the data used
call  the call used to create the object

Author(s)
Gianfranco Piras

References

Examples
data(Produc, package = "plm")
data(usaww)
GM <- spgm(log(gsp)-log(pc)+log(pc)+log(emp)+unemp, data=Produc, listw = usaww, moments="fullweights", spatial.error = TRUE)
summary(GM)

sphtest  

sphtest  *Hausman test for spatial panel data models*

Description
Hausman specification test for spatial panel data models

Usage
sphtest(x, ...)
## S3 method for class 'formula'
sphtest(x, data, index = NULL, listw, spatial.model = c("lag", "error", "sarar"), method = c("ML", "GM"), errors = c("KKP", "BSK"), ...)
## S3 method for class 'splm'
sphtest(x, x2, ...)

rho  a vector including the spatial parameter and the variance components (see Details)
model  the matrix of the data used
call  the call used to create the object

Author(s)
Gianfranco Piras

References

Examples
data(Produc, package = "plm")
data(usaww)
GM <- spgm(log(gsp)-log(pc)+log(pc)+log(emp)+unemp, data=Produc, listw = usaww, moments="fullweights", spatial.error = TRUE)
summary(GM)

sphtest  

sphtest  *Hausman test for spatial panel data models*

Description
Hausman specification test for spatial panel data models

Usage
sphtest(x, ...)
## S3 method for class 'formula'
sphtest(x, data, index = NULL, listw, spatial.model = c("lag", "error", "sarar"), method = c("ML", "GM"), errors = c("KKP", "BSK"), ...)
## S3 method for class 'splm'
sphtest(x, x2, ...)
Arguments

- **x**: an object of class `formula` or `splm`
- **x2**: an object of class `splm`
- **data**: an object of class `data.frame` or `pdata.frame`. An optional data frame containing the variables in the model. When the object is a `data.frame`, the first two columns may contain the indexes. See `index`
- **index**: if not NULL (default), a character vector to identify the indexes among the columns of the `data.frame`
- **listw**: an object of class `listw` created for example by `nb2listw`
- **spatial.model**: one of `c("lag","error","sarar")`, the model to be estimated (only lag, only error, both lag and error dependence)
- **method**: one of `c("ML","GM")`
- **errors**: one of `c("BSK","KPP")`. When method is "ML" defines the specification of the innovations
- **...**: additional arguments to be passed

Value

an object of class `htest`

Author(s)

Gianfranco Piras

References


See Also

`spgm`

Examples

data(Produc, package="plm")
data(usaww)
fm <- log(gsp)+log(pcap)+log(pc)+log(emp)+unemp
test1 <- sphtest(fm, data=Produc, listw = mat2listw(usaww),
spatial.model = "error", method="GM")
test1
mod1 <- spgm(fm, data=Produc, listw = usaww, model = "random",
spatial.error = TRUE, moments="fullweights")
mod2 <- spgm(fm, data=Produc, listw = usaww, model = "within",
spatial.error = TRUE)
test2 <- sphtest(mod1, mod2)
test2
spml  

*Spatial Panel Model by Maximum Likelihood*

**Description**

Maximum likelihood (ML) estimation of spatial panel models, possibly with fixed or random effects.

**Usage**

```r
spml(formula, data, index=NULL, listw, listw2=listw, na.action, 
    model=c("within","random","pooling"),
    effect=c("individual","time","twoways"),
    lag=FALSE, spatial.error=c("b","kkp","none"),
    ...
```

**Arguments**

- `formula`  
  a symbolic description of the model to be estimated
- `data`  
  an object of class `data.frame` or `pdata.frame`. A data frame containing the variables in the model. When the object is a `data.frame`, the first two columns shall contain the indexes, unless otherwise specified. See `index`  
- `index`  
  if not NULL (default), a character vector to identify the indexes among the columns of the `data.frame`  
- `listw`  
  an object of class `listw` or a matrix. It represents the spatial weights to be used in estimation.  
- `listw2`  
  an object of class `listw` or a matrix. Second of set spatial weights for estimation, if different from the first (e.g., in a 'sarar' model).  
- `na.action`  
  see `spdep` for more details.  
- `model`  
  one of c("within", "random", "pooling").  
- `effect`  
  one of c("individual","time","twoways"); the effects introduced in the model.  
- `lag`  
  default=FALSE. If TRUE, a spatial lag of the dependent variable is added.  
- `spatial.error`  
  one of c("b","kkp","none"). The type of spatial error in the specification, if any. See details.  
- `...`  
  additional argument to pass over to other functions

**Details**

The models are estimated by two-step Maximum Likelihood. Further optional parameters to be passed on to the estimator may be: `pvar`: if TRUE the `pvar` function is called `hess`: if TRUE use numerical Hessian instead of GLS for the standard errors of the estimates `quiet`: if FALSE report function and parameters values during optimization `initval`: one of c("zeros", "estimate"). the initial values for the parameters. If "zeros" a vector of zeros is used. If "estimate" the initial
values are retrieved from the estimation of the nested specifications. Alternatively, a numeric vector can be specified. x.tol: Tolerance. See `nlminb` for details. rel.tol: Relative tolerance. See `nlminb` for details.

Value

An object of class "splm".

- **coefficients**: coefficients estimate of the model parameters
- **arcoef**: the coefficient for the spatial lag on \( y \)
- **errcomp**: the estimates of the error variance components
- **vcov**: the asymptotic variance covariance matrix of the estimated coefficients
- **vcov.arcoef**: the asymptotic variance of the estimated spatial lag parameter
- **vcov.errcomp**: the asymptotic variance covariance matrix of the estimated error covariance parameters
- **type**: 'random effects ML'
- **residuals**: the model residuals
- **fitted.values**: the fitted values, calculated as \( \hat{y} = X \hat{\beta} \)
- **sigma2**: GLS residuals variance
- **model**: the matrix of the data used
- **call**: the call used to create the object
- **logLik**: the value of the log likelihood function at the optimum
- **errors**: the value of the errors argument

Author(s)

Giovanni Millo

References


See Also

*spgm*
Examples

```r
data(Produc, package = "plm")
data(usaww)
fm <- log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp
## the two standard specifications (SEM and SAR) one with FE
## and the other with RE:
## fixed effects panel with spatial errors
fespaterr <- spml(fm, data = Produc, listw = mat2listw(usaww),
    model="within", spatial.error="b", Hess = FALSE)
summary(fespaterr)
## random effects panel with spatial lag
respatlag <- spml(fm, data = Produc, listw = mat2listw(usaww),
    model="random", spatial.error="none", lag=TRUE)
summary(respatlag)
## calculate impact measures
impac1 <- impacts(respatlag, listw = mat2listw(usaww, style = "W"), time = 17)
summary(impac1, zstats=TRUE, short=TRUE)
```

spreml

Spatial Panel Model with Random Effects by Maximum Likelihood

Description

Maximum likelihood (ML) estimation of spatial panel models with random effects and serial error correlation.

Usage

```r
spreml(formula, data, index = NULL, w, w2=w, lag = FALSE,
    errors = c("semsrre", "semsr", "srre", "semre",
         "re", "sr", "sem", "ols", "sem2srre", "sem2re"),
    pvar = FALSE, hess = FALSE, quiet = TRUE,
    initval = c("zeros", "estimate"),
    x.tol = 1.5e-18, rel.tol = 1e-15, ...)
```

Arguments

- **formula**: a symbolic description of the model to be estimated
- **data**: an object of class `data.frame` or `pdata.frame`. A data frame containing the variables in the model. When the object is a `data.frame`, the first two columns shall contain the indexes, unless otherwise specified. See index
- **index**: if not NULL (default), a character vector to identify the indexes among the columns of the `data.frame`
- **w**: an object of class `listw` or a matrix. It represents the spatial weights to be used in estimation.
- **w2**: an object of class `listw` or a matrix. Second set of spatial weights for estimation, if different from the first (e.g., in a 'sar' model).
### Details

Second-level wrapper for estimation of random effects models with serial and spatial correlation. The specifications without serial correlation (no "sr" in `errors`) can be called through `spml`, the extended ones only through `spreml`. The models are estimated by two-step Maximum Likelihood. Abbreviations in `errors` correspond to: "sem" Anselin-Baltagi type spatial autoregressive error: if present, random effects are not spatially correlated; "sem2" Kapoor, Kelejian and Prucha-type spatial autoregressive error model with spatially correlated random effects; "sr" serially correlated remainder errors; "re" random effects; "ols" spherical errors (usually combined with `lag=t`). The optimization method can be passed on as optional parameter. Default is "n1mib"; all constrained optimization methods from `maxLik` are allowed ("BFGS", "NM", "SANN") but the latter two are still experimental.

### Value

An object of class "splm".

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>coefficients</td>
<td>coefficients estimate of the model parameters</td>
</tr>
<tr>
<td>arcoef</td>
<td>the coefficient for the spatial lag on $y$</td>
</tr>
<tr>
<td>errcomp</td>
<td>the estimates of the error variance components</td>
</tr>
<tr>
<td>vcov</td>
<td>the asymptotic variance covariance matrix of the estimated coefficients</td>
</tr>
<tr>
<td>vcov.arcoef</td>
<td>the asymptotic variance of the estimated spatial lag parameter</td>
</tr>
<tr>
<td>vcov.errcomp</td>
<td>the asymptotic variance covariance matrix of the estimated error covariance parameters</td>
</tr>
<tr>
<td>type</td>
<td>'random effects ML'</td>
</tr>
<tr>
<td>residuals</td>
<td>the model residuals</td>
</tr>
<tr>
<td>fitted.values</td>
<td>the fitted values, calculated as $\hat{y} = X \hat{\beta}$</td>
</tr>
<tr>
<td>sigma2</td>
<td>GLS residuals variance</td>
</tr>
<tr>
<td>model</td>
<td>the matrix of the data used</td>
</tr>
</tbody>
</table>
Method for summarizing the results of objects of class 'splm'

Usage

```r
## S3 method for class 'splm'
summary(object,...)
```

Arguments

- **object**: an object of class 'splm'
- **...**: additional arguments to be passed

Description

Method for summarizing the results of objects of class 'splm'

References


Examples

```r
data(Produc, package = "plm")
data(usaww)
fm <- log(gsp) - log(pcap) + log(pc) + log(emp) + unemp
## random effects panel with spatial lag and serial error correlation
## optimization method set to "BFGS"
sarsrmod <- spreml(fm, data = Produc, w = usaww, errors="sr", lag=TRUE, method="BFGS")
summary(sarsrmod)
```
Details

The summary function `summary.splm` returns objects of class 'splm' organized in a coefficient matrix.

Also a matrix for the error components, or the spatial coefficients will be generated depending on the estimated model.

When the 'splm' is produced by the function 'spsegm', the summary will be generated looping over the number of equations in the system.

Value

An object of class 'summary.splm'

Author(s)

Giovanni Millo, Gianfranco Piras

See Also

spml, spgm

Examples

```r
data(Produc, package = "plm")
data(usaww)
GM <- spgm(log(gsp)-log(pcap)+log(pc)+log(emp)+unemp, data=Produc,
  listw=usaww, moments = "fullweights", spatial.error = TRUE)
summary(GM)
```

usaww  Spatial weights matrix - US states

Description

Spatial weights matrix of the 48 continental US States based on the queen contiguity criterium.

Usage

data(usaww)

Format

A matrix with elements different from zero if state i and j are neighbors. Weights are row standardized. According to the queen contiguity criterium, Arizona and Colorado are considered neighbours.

Author(s)

Giovanni Millo
Covariance extractor method for `splm` objects. Seldom used as such but needed, e.g., for interoperability with testing functions in `lmtest` and `car`.

**Usage**

```r
## S3 method for class 'splm'
vcov(object, ...)
```

**Arguments**

- `object`: an object of class `splm`
- `...`: additional arguments to be passed; currently not used

**Value**

A covariance matrix of beta coefficients

**Author(s)**

Giovanni Millo

**References**


**Examples**

```r
## not run:
## data(Produc, package="plm")
## data(usaww)
## fm <- log(gsp)-log(pcap)+log(pc)+log(emp)+unemp
## sarremod <- spml(fm, data=Produc, listw = mat2listw(usaww),
##     model="random", lag=TRUE, spatial.error="none")
## ## compact representation of betas
## library(lmtest)
## coeftest(sarremod)
## ## linear hypothesis test
## library(car)
## lht(sarremod, "log(pcap)=log(pc)"
```
Index

*Topic **datasets**
  Insurance, 5
  itaww, 7
  RiceFarms, 9
  riceww, 10
  usaww, 25
*Topic **htest**
  bsjktest, 2
  bsktest, 3
  rwtest, 10
  slmtest, 14
  sphtest, 18
*Topic **slag**
  slag, 13
*Topic **spatial**
  effects.splm, 4
  listw2dgCMatrix, 7
  print.splm, 8
  spgm, 15
  spml, 20
  spreml, 22
  summary.splm, 24
*Topic **vcov**
  vcov.splm, 26

bsjktest, 2
bsktest, 3
effects.splm, 4

Insurance, 5
itaww, 7
ivsplm(spgm), 15

listw2dgCMatrix, 7

print.effects.splm(effects.splm), 4
print.splm, 8

RiceFarms, 9
riceww, 10

rwtest, 10
slag, 13
slmtest, 14
sperrorgm(spgm), 15
spgm, 15
sphtest, 18
spml, 20
spreml, 22
spssarargm(spgm), 15
summary.splm, 24
usaww, 25
vcov.splm, 26