Package ‘spsur’

September 17, 2020

Type Package

Title Spatial Seemingly Unrelated Regression Models

Version 1.0.1.6

Date 2020-09-17

Maintainer Roman Minguez <roman.minguez@uclm.es>

Description A collection of functions to test and estimate Seemingly
Unrelated Regression (usually called SUR) models, with spatial structure, by maximum
likelihood and three-stage least squares. The package estimates the
most common spatial specifications, that is, SUR with Spatial Lag of
X regressors (called SUR-SLX), SUR with Spatial Lag Model (called SUR-SLM),
SUR with Spatial Error Model (called SUR-
SEM), SUR with Spatial Durbin Model (called SUR-SDM),
SUR with Spatial Durbin Error Model (called SUR-SDEM),
SUR with Spatial Autoregressive terms and Spatial Autoregressive
Disturbances (called SUR-SARAR), SUR-SARAR with Spatial Lag of X
regressors (called SUR-GNM) and SUR with Spatially Independent Model (called SUR-SIM).
The methodology of these models can be found in next references
Mur, J., Lopez, F., and Herrera, M. (2010) <doi:10.1080/17421772.2010.516443>
Lopez, F.A., Mur, J., and Angulo, A. (2014) <doi:10.1007/s00168-014-0624-2>.

LazyData true

Encoding UTF-8

License GPL-3

Depends R (>= 3.5), methods (>= 3.5), stats (>= 3.5)

Imports car (>= 3.0-8), Formula (>= 1.2-3), ggplot2 (>= 3.3.0),
gmodels (>= 2.18.1), gridExtra (>= 2.3), knitr (>= 1.28),
lmtest (>= 0.9-37), MASS (>= 7.3-51.4), Matrix (>= 1.2-16),
minqa (>= 1.2.4), numDeriv (>= 2016.8-1.1), rlang (>= 0.4.6),
Rdpack (>= 0.11-1), rmarkdown (>= 2.1), sparseMVN (>= 0.2.1.1),
spatialreg (>= 1.1-5), spdep (>= 1.1-3)

Suggests bookdown (>= 0.18), dplyr (>= 0.8.5), sf (>= 0.9-0)

VignetteBuilder knitr

RoxygenNote 7.1.1

RdMacros Rdpack

URL https://github.com/rominsal/spsur

BugReports https://github.com/rominsal/spsur/issues

NeedsCompilation no

Author Ana Angulo [aut],
Fernando A Lopez [aut],
Roman Minguez [aut, cre],
Jesus Mur [aut]

Repository CRAN

Date/Publication 2020-09-17 10:50:02 UTC

R topics documented:

---

spsur-package *Spatial Seemingly Unrelated Regression Models.*

Description

spsur offers the user a collection of functions to estimate Spatial Seemingly Unrelated Regression (SUR) models by maximum likelihood or three-stage least squares, using spatial instrumental variables. Moreover, **spsur** obtains a collection of misspecification tests for omitted or wrongly specified spatial structure. The user will find spatial models more popular in applied research such as the SUR-SLX, SUR-SLM, SUR-SEM, SUR-SDM, SUR-SDEM SUR-SARAR and SUR-GNM plus the spatially independent SUR, or SUR-SIM.

Details

Some functionalities that have been included in **spsur** package are:

1. Testing for spatial effects

The function `lmtestspsur` provides a collection of Lagrange Multipliers, LM, for testing different forms of spatial dependence in SUR models. They are extended versions of the well-known LM tests for omitted lags of the explained variable in the right hand side of the equation, LM-SLM, the LM tests for omitted spatial errors, LM-SEM, the join test of omitted spatial lags and spatial errors, LM-SARAR, and the robust version of the firt two Lagrange Multipliers, LM*-SLM and LM*-SEM.
These tests can be applied to models always with a SUR nature. Roughly, we may distinguish two situations:

- Datasets with a single equation *G=1*, for different time periods *Tm>1* and a certain number of spatial units in the cross-sectional dimension, *N*. This is what we call *spatial panel datasets*. In this case, the SUR structure appears in form of (intra) serial dependence in the errors of each spatial unit.
- Datasets with a several equations *G>1*, different time periods *Tm>1* and a certain number of spatial units, *N*. The SUR structure appears, as usual, because the errors of the spatial units for different equations are contemporaneously correlated.

2. Estimation of the Spatial SUR models

As indicated above, **spsur** package may work with a list of different spatial specifications. They are the following:

- *SUR-SIM*: SUR model without spatial effects

$$y_{tg} = X_{tg}\beta_g + \epsilon_{tg}$$

- *SUR-SLX*: SUR model with spatial lags of the regresors

$$y_{tg} = X_{tg}\beta_g + WX_{tg}\theta_g + \epsilon_{tg}$$

- *SUR-SLM*: SUR model with spatial lags of the endogenous

$$y_{tg} = \rho_g W y_{tg} + X_{tg}\beta_g + \epsilon_{tg}$$

- *SUR-SEM*: SUR model with spatial errors

$$y_{tg} = X_{tg}\beta_g + u_{tg}$$

$$u_{tg} = \lambda_g W u_{tg} + \epsilon_{tg}$$

- *SUR-SDM*: SUR model with spatial lags of the endogenous variable and of the regressors or Spatial Durbin model

$$y_{tg} = \rho_g W y_{tg} + X_{tg}\beta_g + WX_{tg}\theta_g + \epsilon_{tg}$$

- *SUR-SDEM*: SUR model with spatial errors and spatial lags of the endogenous variable and of the regressors

$$y_{tg} = X_{tg}\beta_g + WX_{tg}\theta_g + u_{tg}$$

$$u_{tg} = \lambda_g W u_{tg} + \epsilon_{tg}$$

- *SUR-SARAR*: Spatial lag model with spatial errors

$$y_{tg} = \rho_g W y_{tg} + X_{tg}\beta_g + u_{tg}$$

$$u_{tg} = \lambda_g W u_{tg} + \epsilon_{tg}$$

- *SUR-GNM*: SUR model with spatial lags of the explained variables, regressors and spatial errors

$$y_{tg} = \rho_g W y_{tg} + X_{tg}\beta_g + WX_{tg}\theta_g + u_{tg}$$

$$u_{tg} = \lambda_g W u_{tg} + \epsilon_{tg}$$

where y_{tg}, u_{tg} and ϵ_{tg} are *(Nx1)* vectors; X_{tg} is a matrix of regressors of order *(NxP)*; ρ_g and λ_g are parameters of spatial dependence and *W* is the *(NxN)* spatial weighting matrix.

These specifications can be estimated by maximum-likelihood methods, using the function `spsurml`. Moroever, the models that include spatial lags of the explained variables in the right hand side of the equations, and the errors are assumed to be spatially incorrelated (namely, the SUR-SLM and the SUR-SDM), can also be estimated using three-stage least-squares, `spsur3sls`, using spatial instrumental variable to correct for the problem of endogeneity present in these cases.

3. Diagnostic tests

Testing for inconsistencies or misspecifications in the results of an estimated (SUR) model should be a primary task for the user. **spsur** focuses, especifically, on two main question such as omitted or wrongly specified spatial structure and the existence of structural breaks or relevant restrictions in the parameters of the model. In this sense, the user will find:

1. *Marginal tests*
   The Marginal Multipliers test for omitted or wrongly specified spatial structure in the equations. They are routinely part of the output of the maximum-likelihood estimation, shown by `spsurml`. In particular, the LM($\rho|\lambda$) tests for omitted spatial lags in a model specified with spatial errors (SUR-SEM; SUR-SDEM). The LM($\lambda|\rho$) tests for omitted spatial error in a model specified with spatial lags of the explained variable (SUR-SLM; SUR-SDM).

2. *Coefficients stability tests*
   spsur includes two functions designed to test for linear restrictions on the β coefficients of the models and on the spatial coefficients (ρs and λs terms). The function for the first case is `wald_betas` and `wald_deltas` that of the second case. The user has ample flexibility to define different forms of linear restrictions, so that it is possible, for example, to test for their time constancy to identify structural breaks.

4. Marginal effects

In recent years, since the publication of LeSage and Pace (2009), it has become popular in spatial econometrics to evaluate the multiplier effects that a change in the value of a regressor, in a point in the space, has on the explained variable. **spsur** includes a function, `impacts`, that computes these effects. Specifically, `impacts` obtains the average, over the *N* spatial units and *Tm* time periods, of such a change on the contemporaneous value of the explained variable located in the same point as the modified variable. This is the so-called *Average Direct effect*. The *Average Indirect effect* measure the proportion of the impact that spills-over to other locations. The sum of the two effects is the *Average Total effect*.
These estimates are complemented with a measure of statistical significance, following the randomization approach suggested by LeSage and Pace (2009).

5. Additional functionalities

A particular feature of **spsur** is that the package allows to obtain simulated datasets with a SUR nature and the spatial structure decided by the user. This is the purpose of the function `dgp_spsur`. The function can be inserted into a more general code to solve, for example, Monte Carlo studies related to these type of models or, simply, to show some of the stylized characteristics of a SUR model with certain spatial structure.

Datasets

spsur includes three different datasets: spc, NCOVR and spain.covid. These sets are used to illustrate the capabilities of different functions. Briefly, their main characteristics are the following

- The *spc* dataset (Spatial Phillips-Curve) is a classical dataset taken from Anselin (1988, p. 203), of small dimensions.
- The *NCOVR* dataset comprises Homicides and a list of selected socio-economic variables for continental U.S. counties in four decennial census years: 1960, 1970, 1980 and 1990. It is freely available from `https://geodacenter.github.io/data-and-lab/ncovr/`. *NCOVR* is a typical spatial panel dataset *(G=1)*.
- The *spain.covid* dataset comprises Within and Exit mobility index together with the weeklly incidence COVID-19 at Spain provinces from February 21 to May 21 2020. `https://www.mitma.gob.es/ministerio/covid-19/evolucion-movilidad-big-data`

Author(s)

Fernando López	<fernando.lopez@upct.es>
Román Mínguez	<roman.minguez@uclm.es>
Jesús Mur	<jmur@unizar.es>

References

- Breusch T, Pagan A (1980). The Lagrange multiplier test and its applications to model specification in econometrics. *Review of Economic Studies* 47: 239-254.
- LeSage, J., and Pace, R. K. (2009). *Introduction to spatial econometrics*. Chapman and Hall/CRC.
- Lopez, F.A., Mur, J., and Angulo, A. (2014). Spatial model selection strategies in a SUR framework. The case of regional productivity in EU. *Annals of Regional Science*, 53(1), 197-220.
- López, F.A., Martínez-Ortiz, P.J., and Cegarra-Navarro, J.G. (2017). Spatial spillovers in public expenditure on a municipal level in Spain. *Annals of Regional Science*, 58(1), 39-65.
- Mur, J., López, F., and Herrera, M. (2010). Testing for spatial effects in seemingly unrelated regressions. *Spatial Economic Analysis*, 5(4), 399-440.

---

dgp_spsur — *Generation of a random dataset with a spatial SUR structure.*

Description

The purpose of the function `dgp_spsur` is to generate a random dataset with the dimensions and spatial structure decided by the user. This function may be useful in pure simulation experiments or with the aim of showing specific properties and characteristics of a spatial SUR dataset and inferential procedures related to them.

The user of `dgp_spsur` should think in terms of a Monte Carlo experiment. The arguments of the funtion specify the dimensions of the dataset to be generated, the spatial mechanism underlying the data, the intensity of the SUR structure among the equations and the values of the parameters to be used to obtain the simulated data, which includes the error terms, the regressors and the explained variables.

Usage

```
dgp_spsur (Sigma, Tm = 1, G, N, Betas, Thetas = NULL,
                  rho = NULL,lambda = NULL, p = NULL, listw = NULL,
                  X = NULL, pdfU = "nvrnorm", pdfX = "nvrnorm")
```

Arguments

`Sigma`	Covariance matrix between the G equations of the SUR model. This matrix should be definite positive and the user must check for that.
`Tm`	Number of time periods.Default = `1`
`G`	Number of equations.
`N`	Number of cross-section or spatial units
`Betas`	A row vector of order $(1xP)$ showing the values for the *beta* coefficients. The first P_1 terms correspond to the first equation (where the first element is the intercept), the second P_2 terms to the coefficients of the second equation and so on.

Thetas
: Values for the θ coefficients in the *G* equations of the model, when the type of spatial SUR model to be simulated is a "slx", "sdm" or "sdem". *Thetas* is a row vector of order $1xPTheta$, where $PThetas = p - G$; let us note that the intercept cannot appear among the spatial lags of the regressors. The first $1xKTheta_1$ terms correspond to the first equation, the second $1xPTheta_2$ terms correspond to the second equation, and so on. Default = `NULL`.

rho
: Values of the coefficients $\rho_g; g = 1, 2, ..., G$ related to the spatial lag of the explained variable of the g-th equation. If rho is an scalar and there are *G* equations in the model, the same value will be used for all the equations. If rho is a row vector, of order *(1xG)*, the function `dgp_spsur` will use these values, one for each equation. Default = `NULL`.

lambda
: Values of the coefficients $\lambda_g; g = 1, 2, ..., G$ related to the spatial lag of the errors in the *G* equations. If $lambda$ is an scalar and there are *G* equations in the model, the same value will be used for all the equations. If $lambda$ is a row vector, of order *(1xG)*, the function `dgp_spsur` will use these values, one for each equation of the spatial errors. Default = `NULL`.

p
: Number of regressors by equation, including the intercept. *p* can be a row vector of order *(1xG)*, if the number of regressors is not the same for all the equations, or a scalar, if the *G* equations have the same number of regressors.

listw
: A `listw` object created for example by `nb2listw` from **spatialreg** package; if `nb2listw` not given, set to the same spatial weights as the `listw` argument. It can also be a spatial weighting matrix of order *(NxN)* instead of a `listw` object. Default = `NULL`.

X
: This argument tells the function `dgp_spsur` which *X* matrix should be used to generate the SUR dataset. If *X* is different from `NULL`, `{dgp_spsur}` will upload the *X* matrix selected in this argument. Note that the *X* must be consistent with the dimensions of the model. If *X* is `NULL`, `dgp_spsur` will generate the desired matrix of regressors from a multivariate Normal distribution with mean value zero and identity (PxP) covariance matrix. As an alternative, the user may change this probability distribution function to the uniform case, $U(0, 1)$, through the argument *pdfX*. Default = `NULL`.

pdfU
: Multivariate probability distribution function, mdf, from which the values of the error terms will be drawn. The covariance matrix is the Σ matrix specificied by the user in the argument *Sigma*. The funtion `dgp_spsur` provides two mdf, the multivariate Normal, which is the default, and the log-Normal distribution funtion which means just exponenciate the sampling drawn form a $N(0, \Sigma)$ distribution. Default = `"nvrnorm"`.

pdfX
: Multivariate probability distribution function, mdf, from which the values of the regressors will be drawn. The regressors are assumed to be independent. `dgp_spsur` provides two mdf, the multivariate Normal, which is the default, and the uniform in the interval $U[0, 1]$, using the dunif function, `dunif`, from the **stats** package. Default = `"nvrnorm"`.

Details

The purpose of the function `dgp_spsur` is to generate random datasets, of a SUR nature, with the spatial structure decided by the user. The function requires certain information to be supplied externally because, in fact, dgp_spsur constitutes a Data Generation Process, DGP. The following aspects should be addressed:

- The user must define the dimensions of the dataset, that is, number of equations, *G*, number of time periods, *Tm*, and number of cross-sectional units, *N*.#'
- Then, the user must choose the type of spatial structure desired for the model from among the list of candidates of "sim", "slx", "slm", "sem", "sdm", "sdem" or "sarar"; the default is the "sim" specification which does not have spatial structure. The decision is made implicitly, just omiting the specification of the spatial parameters which are not involved in the model (i.e., in a "slm" there are no λ parameters but appear ρ parameters; in a "sdem" model there are λ and θ parameters but no ρ coefficients). Of course, if the user needs a model with spatial structure, a *(nxN)* weighting matrix, *W*, should be chosen.
- The next step builds the equations of the SUR model. In this case, the user must specify the number of regressors that intervene in each equation and the coefficients, β parameters, associated with each regressor. The *first* question is solved through the argument *p* which, if a scalar, indicates that the same number of regressors should appear in all the equations of the model; if the user seeks for a model with different number of regressors in the *G* equations, the argument *p* must be a *(1xG)* row vector with the required information. It must be remembered that dgp_spsur assumes that an intercept appears in all equations of the model.

  The *second* part of the problem posited above is solved through the argument *Betas*, which is a row vector of order *(1xp)* with the information requiered for this set of coeficcients.
- The user must specify, also, the values of the spatial parameters corresponding to the chosen specification; we are refering to the ρ_g, λ_g and θ_g, for $g = 1, ..., G and k = 1, ..., K_g$ parameters. This is done throught the arguments *rho*, *lambda* and *theta*. The firs two, *rho* and *lambda*, work as *K*: if they are scalar, the same value will be used in the *G* equations of the SUR model; if they are *(1xG)* row vectors, a different value will be assigned for each equation.

  Moreover, *theta* works like the argument *beta*. The user must define a row vector of order $1xPTheta$ showing these values. It is worth to remember that in no case the intercept will appear among the lagged regressors.
- Finally, the user must decide which values of the regressors and of the error terms are to be used in the simulation. The regressors can be uploaded from an external matrix generated previously by the user. This is the argument *X*. It is the responsability of the user to check that the dimensions of the external matrix are consistent with the dataset required for the SUR model. A second possibility implies the regressors to be generated randomly by the function dgp_spsur. In this case, the user must select the probability distribution function from which the corresponding data (of the regressors and the error terms) are to be drawn.

  dgp_spsur provides two multivariate distibution functions, namely, the Normal and the log-Normal for the errors (the second should be taken as a clear departure from the standard assumption of normality). In both cases, random matrices of order *(TmNxG)* are obtained from a multivariate normal distribution, with a mean value of zero and the covariance matrix specified in the argument *Sigma*; then, this matrix is exponentiated for the log-Normal case. Roughly, the same procedure applies for drawing the values of the regressor. There are two distribution functions available, the normal and the uniform in the interval $U[0, 1]$; the regressors are always independent.

Value

A list with a vector Y of order *(TmNGx1)* with the values generated for the explained variable in the G equations of the SUR and a matrix XX of order (*(TmNGxsum(p))*, with the values generated for the regressors of the SUR, including an intercept for each equation.

Author(s)

Fernando López	<fernando.lopez@upct.es>
Román Mínguez	<roman.minguez@uclm.es>
Jesús Mur	<jmur@unizar.es>

See Also

spsurml, spsur3sls, spsurtime

Examples

```
## VIP: The output of the whole set of the examples can be examined
## by executing demo(demo_dgp_spsur, package="spsur")

####################################
######## CROSS SECTION DATA ########
####################################

####################################
#### Example 1: DGP SLM model
####################################
rm(list = ls()) # Clean memory
Tm <- 1 # Number of time periods
G <- 3 # Number of equations
N <- 200 # Number of spatial elements
p <- 3 # Number of independent variables
Sigma <- matrix(0.3, ncol = G, nrow = G)
diag(Sigma) <- 1
Betas <- c(1, 2, 3, 1, -1, 0.5, 1, -0.5, 2)
rho <- 0.5 # level of spatial dependence
lambda <- 0.0 # spatial autocorrelation error term = 0
##  random coordinates
co <- cbind(runif(N,0,1),runif(N,0,1))
lw <- spdep::nb2listw(spdep::knn2nb(spdep::knearneigh(co, k = 5,
                                                    longlat = FALSE)))
DGP <- dgp_spsur(Sigma = Sigma, Betas = Betas,
                 rho = rho, lambda = lambda, Tm = Tm,
                 G = G, N = N, p = p, listw = lw)

SLM <- spsurml(X = DGP$X, Y = DGP$Y, Tm = Tm, N = N, G = G,
```

```
                p = c(3, 3, 3), listw = lw, type = "slm",
                method = "LU", control = list(fdHess = TRUE))
summary(SLM)

#####################################
#### Example 2: DGP SEM model with Tm>1; G=1 and
#### different p for each equation
#####################################
rm(list = ls()) # Clean memory
Tm <- 3 # Number of time periods
G <- 1 # Number of equations
N <- 500 # Number of spatial elements
p <- c(2,3,4) # Number of independent variables
Sigma <- matrix(0.8, ncol = Tm, nrow = Tm)
diag(Sigma) <- 1
Betas <- c(1,2,1,2,3,1,2,3,4)
rho <- 0 # level of spatial dependence = 0
lambda <- c(0.2,0.5,0.8)
## spatial autocorrelation error terms for each equation
## random coordinates
co <- cbind(runif(N,0,1),runif(N,0,1))
lw <- spdep::nb2listw(spdep::knn2nb(spdep::knearneigh(co, k = 5,
                                                   longlat = FALSE)))
DGP2 <- dgp_spsur(Sigma = Sigma, Betas = Betas, rho = rho,
                  lambda = lambda, Tm = Tm, G = G, N = N, p = p,
                  listw = lw)
SLM2 <- spsurml(X = DGP2$X, Y = DGP2$Y, Tm = Tm, N = N, G = G,
               p = c(2,3,4), listw = lw, type = "slm",
               control = list(fdHess = TRUE))
summary(SLM2)
SEM2 <- spsurml(X = DGP2$X, Y = DGP2$Y, Tm = Tm, N = N, G = G,
               p = c(2,3,4), listw = lw, type = "sem",
               method = "LU", control = list(fdHess = TRUE))
summary(SEM2)

#####################################
##### Example with G>1 and Tm>>1
#####################################
### It usually requires 1-2 minutes maximum
rm(list = ls()) # Clean memory
Tm <- 10 # Number of time periods
G <- 3 # Number of equations
N <- 100 # Number of spatial elements
p <- 3 # Number of independent variables
Sigma <- matrix(0.5, ncol = G, nrow = G)
diag(Sigma) <- 1
Betas <- rep(1:3, G)
rho <- c(0.5, 0.1, 0.8)
lambda <- 0.0 # spatial autocorrelation error term = 0
## random coordinates
co <- cbind(runif(N,0,1),runif(N,0,1))
lw <- spdep::nb2listw(spdep::knn2nb(spdep::knearneigh(co, k = 5,
                                                   longlat = FALSE)))
```

```
DGP3 <- dgp_spsur(Sigma = Sigma, Betas = Betas, rho = rho,
                  lambda = lambda, Tm = Tm, G = G, N = N, p = p,
                  listw = lw)
SLM3  <-spsurml(Y = DGP3$Y, X = DGP3$X, G = G, N = N, Tm = Tm,
                p = p, listw = lw, type = "slm",
                method = "LU", control = list(fdHess = TRUE))
summary(SLM3)
```

---

impactspsur — *Direct, indirect and total effects estimated for a spatial SUR model*

Description

This function is a wrapper for `impacts` method used in **spatialreg** package. Nevertheless, in this case the same method is used for both `lagsarlm` and `lmSLX` objects. For details of implementation, see the documentation of `impacts` function in **spatialreg** package.
The function obtains the multiplier effects, on the explained variable, of a change in a regressor for the model that has been estimated. For reasons given below, this function only applies to models with an autoregressive structure ("slm", "sdm", "sarar" and "gnm") or with spatial lags of the regressors ("slx", "sdem").
The measurement of the multiplier effects is a bit more complicated than in a pure time series context because, due to the spatial structure of the model, part of the impacts spills over non uniformly over the space. Using the notation introduced by LeSage and Pace (2009) we distinguish between:

- **Average Direct effects**: The average over the *N* spatial units and *Tm* time periods of the effect of a unitary change in the value of a explanatory variable on the contemporaneous value of the corresponding explained variable, located in the same point of the intervened regressor. This calculus is solved for all the regressors that appear in the *G* equations of the model.
- **Average Indirect effects**: The average over the *N* spatial units and *Tm* time periods of the effects of a unitary change in the value of a explanatory variable on the contemporaneous value of the corresponding explained variable, located in a different spatial unit that that of the intervened regressor. This calculus is solved for all the regressors that appear in the *G* equations of the model.
- **Average total effects**: The sum of Direct and Indirect effects.

The information on the three estimated effects is supplement with an indirect measure of statistical significance obtained from the randomization approach introduced in LeSage and Pace (2009).

Usage

```
impactspsur (obj, ..., tr = NULL, R = NULL, listw = NULL,
                        evalues = NULL,tol = 1e-06,
                        empirical = FALSE, Q = NULL)
```

Arguments

`obj`	An `spsur` object created by `spsurml`, `spsur3sls` or `spsurtime`.
`...`	Arguments passed through to methods in the **coda** package
`tr`	A vector of traces of powers of the spatial weights matrix created using `trW`, for approximate impact measures; if not given, `listw` must be given for exact measures (for small to moderate spatial weights matrices); the traces must be for the same spatial weights as were used in fitting the spatial regression, and must be row-standardised
`R`	If given, simulations are used to compute distributions for the impact measures, returned as `mcmc` objects; the objects are used for convenience but are not output by an MCMC process
`listw`	If `tr` is not given, a spatial weights object as created by `nb2listw`; they must be the same spatial weights as were used in fitting the spatial regression, but do not have to be row-standardised
`evalues`	vector of eigenvalues of spatial weights matrix for impacts calculations
`tol`	Argument passed to `mvrnorm`: tolerance (relative to largest variance) for numerical lack of positive-definiteness in the coefficient covariance matrix
`empirical`	Argument passed to `mvrnorm` (default FALSE): if true, the coefficients and their covariance matrix specify the empirical not population mean and covariance matrix
`Q`	default NULL, else an integer number of cumulative power series impacts to calculate if `tr` is given

Details

LeSage and Pace (2009) adapt the classical notion of *'economic multiplier'* to the problem of measuring the impact that a unitary change in the value of a regressor, produced in a certain point in space, has on the explained variable. The question is interesting because, due to the spatial structure of the model, the impacts of such change spill non uniformly over the space. In fact, the reaction of the explained variable depends on its relative location in relation to the point of intervention.
To simplify matters, LeSage and Pace (2009) propose to obtain aggregated multipliers for each regressor, just averaging the N^2 impacts that results from intervening the value of each regressor on each of the N points in Space, on the explained variable, measured also in each of the N points in space. This aggregated average is the so-called *Total effect*.
Part of this impact will be absorbed by the explained variable located in the same point of the regressor whose value has been changed (for example, the k-th regresor in the g-th equation, in the n-th spatial unit) or, in other words, we expect that $[dy_{tgn}]/[dx_{ktgn}]ne0$. The aggregated average for the *N* points in space (n=1,2,...,N) and *Tm* time periods is the so-called *Direct effect*. The difference between the *Total effect* and the *Direct effect* measures the portion of the impact on the explained variable that leakes to other points in space, $[dy_{tgn}]/[dx_{ktgm}]fornnem$; this is the *Indirect effect*.

`impacts` obtains the three multipliers together with an indirect measure of statistical significance, according to the randomization approach described in Lesage and Pace (2009). Briefly, they suggest to obtain a sequence of *nsim* random matrices of order *(NTmxG)* from a multivariate normal distribution N(0; *Sigma*), being *Sigma* the estimated covariance matrix of the *G* equations in the SUR model. These random matrices, combined with the observed values of the regressors and the estimated values of the parameters of the corresponding spatial SUR model, are used to obtain simulated values of the explained variables. Then, for each one of the *nsim* experiments, the SUR model is estimated, and the effects are evaluated. The function `impacts` obtains the standard deviations of the *nsim* estimated effects in the randomization procedure, which are used to test the significance of the estimated effects for the original data.

Finally, let us note that this is a SUR model where the G equations are connected only through the error terms. This means that if we intervene a regressor in equation g, in any point is space, only the explained variable of the same equation g should react. The impacts do not spill over equations. Moreover, the impact of a regressor, intervened in the spatial unit n, will cross the borders of this spatial unit only if in the right hand side of the equation there are spatial lags of the explained variables or of the regressors. In other words, the *Indirect effect* is zero for the "sim" and "sem" models. `impacts` produces no output for these two models. Lastly, it is clear that all the impacts are contemporaneous because the equations in the SUR model have no time dynamics.

Value

A list of G objects either of class `lagImpact` or `WXImpact`.

For each of the G objects of the list, if no simulation is carried out the object returned is a list with:

`direct`	numeric vector
`indirect`	numeric vector
`total`	numeric vector

and a matching `Qres` list attribute if Q was given.

On the other hand, for each of the G objects of the list, if simulation is carried out the object returned is a list with:

`res`	a list with three components as for the non-simulation case, with a matching `Qres` list attribute if `Q` was given.
`sres`	a list with three `mcmc` matrices, for the direct, indirect and total impacts with a matching `Qmcmclist` attribute if `Q` was giv

Author(s)

Fernando López	`<fernando.lopez@upct.es>`
Román Mínguez	`<roman.minguez@uclm.es>`
Jesús Mur	`<jmur@unizar.es>`

References

- Bivand, R.S. and Piras G. (2015). Comparing Implementations of Estimation Methods for Spatial Econometrics. *Journal of Statistical Software*, 63(18), 1-36. https://www.jstatsoft.org/v63/i18/.
- LeSage, J., and Pace, R. K. (2009). *Introduction to spatial econometrics*. Chapman and Hall/CRC.

- López, F.A., Mur, J., and Angulo, A. (2014). Spatial model selection strategies in a SUR framework. The case of regional productivity in EU. *Annals of Regional Science*, 53(1), 197-220.
- Mur, J., López, F., and Herrera, M. (2010). Testing for spatial effects in seemingly unrelated regressions. *Spatial Economic Analysis*, 5(4), 399-440.

See Also

`impacts`, `spsurml`, `spsur3sls`

Examples

```
## VIP: The output of the whole set of the examples can be examined
## by executing demo(demo_impactspsur, package="spsur")


################################################
### PURE CROSS SECTIONAL DATA(G>1; Tm=1) ######
################################################

#### Example 1: Spatial Phillips-Curve. Anselin (1988, p. 203)
 rm(list = ls()) # Clean memory
 data(spc)
 lwspc <- spdep::mat2listw(Wspc, style = "W")
 Tformula <- WAGE83 | WAGE81 ~ UN83 + NMR83 + SMSA | UN80 + NMR80 + SMSA
## For SLM, SDM and SARAR models the output is a list of "lagImpact" objects
## See spatialreg::impacts for details.
 spcsur_slm <-spsurml(formula = Tformula, data = spc,
                      type = "slm", listw = lwspc)
 summary(spcsur_slm)
 impacts_slm <- impactspsur(spcsur_slm, listw = lwspc, R = 1000)
## Impacts equation 1
 summary(impacts_slm[[1]], zstats = TRUE, short = TRUE)
## Impacts equation 2
 summary(impacts_slm[[2]], zstats = TRUE, short = TRUE)
## For SLX and SDEM models the output is a list of "WXImpact" objects
## See spatialreg::impacts for details.
## A SUR-SLX model
 spcsur_slx <-spsurml(formula = Tformula, data = spc,
                      type = "slx", listw = lwspc)
 summary(spcsur_slx)
 impacts_slx <- impactspsur(spcsur_slx, listw = lwspc)
 summary(impacts_slx[[1]], zstats = TRUE, short = TRUE)
 summary(impacts_slx[[2]], zstats = TRUE, short = TRUE)

## A SUR-SDM model
 spcsur_sdm <-spsurml(formula = Tformula, data = spc,
                      type = "sdm", listw = lwspc)
 impacts_sdm <- impactspsur(spcsur_sdm, listw = lwspc, R = 1000)
 summary(impacts_sdm[[1]], zstats = TRUE, short = TRUE)
 summary(impacts_sdm[[2]], zstats = TRUE, short = TRUE)
```

```
## A SUR-SDM model with different spatial lags in each equation
 TformulaD <- ~ UN83 + NMR83 + SMSA | UN80
 spcsur_sdm2 <-spsurml(formula = Tformula, data = spc, type = "sdm",
                       listw = lwspc, Durbin = TformulaD)
 summary(spcsur_sdm2)
 impacts_sdm2 <- impactspsur(spcsur_sdm2, listw = lwspc, R = 1000)
 summary(impacts_sdm2[[1]], zstats = TRUE, short = TRUE)
 summary(impacts_sdm2[[2]], zstats = TRUE, short = TRUE)
 ## A SUR-SLX model with different spatial lags in each equation
 spcsur_slx2 <-spsurml(formula = Tformula, data = spc,
                      type = "slx", listw = lwspc, Durbin = TformulaD)
 summary(spcsur_slx2)
 impacts_slx2 <- impactspsur(spcsur_slx2, listw = lwspc)
 summary(impacts_slx2[[1]], zstats = TRUE, short = TRUE)
 summary(impacts_slx2[[2]], zstats = TRUE, short = TRUE)
### A SUR-SDEM model
 spcsur_sdem <-spsurml(formula = Tformula, data = spc,
                      type = "sdem", listw = lwspc)
 impacts_sdem <- impactspsur(spcsur_sdem, listw = lwspc)
 summary(impacts_sdem[[1]], zstats = TRUE, short = TRUE)
 summary(impacts_sdem[[2]], zstats = TRUE, short = TRUE)

### A SUR-SARAR model
 spcsur_sarar <-spsurml(formula = Tformula, data = spc,
                      type = "sarar", listw = lwspc,
                      control = list(tol = 0.01))
 impacts_sarar <- impactspsur(spcsur_sarar, listw = lwspc, R = 1000)
 summary(impacts_sarar[[1]], zstats = TRUE, short = TRUE)
 summary(impacts_sarar[[2]], zstats = TRUE, short = TRUE)

## A SUR-GNM model
 spcsur_gnm <-spsurml(formula = Tformula, data = spc,
                      type = "gnm", listw = lwspc,
                      control = list(tol = 0.1))
 impacts_gnm <- impactspsur(spcsur_gnm, listw = lwspc, R = 1000)
 summary(impacts_gnm[[1]], zstats = TRUE, short = TRUE)
 summary(impacts_gnm[[2]], zstats = TRUE, short = TRUE)
## A SUR-GNM model with different spatial lags in each equation
 TformulaD <- ~ UN83 + NMR83 + SMSA | UN80
 spcsur_gnm2 <-spsurml(formula = Tformula, data = spc, type = "gnm",
                       listw = lwspc, Durbin = TformulaD,
                       control = list(tol = 0.1))
 summary(spcsur_gnm2)
 impacts_gnm2 <- impactspsur(spcsur_gnm2, listw = lwspc, R = 1000)
 summary(impacts_gnm2[[1]], zstats = TRUE, short = TRUE)
 summary(impacts_gnm2[[2]], zstats = TRUE, short = TRUE)

# ###################################
# ######## G=1; Tm>1              ###
# ###################################
#
 rm(list = ls()) # Clean memory
 data(NCOVR, package="spsur")
```

```
 nbncovr <- spdep::poly2nb(NCOVR.sf, queen = TRUE)
### Some regions with no links...
 lwncovr <- spdep::nb2listw(nbncovr, style = "W", zero.policy = TRUE)
 Tformula <- HR80  | HR90 ~ PS80 + UE80 | PS90 + UE90
### A SUR-SLM model
 NCOVRSUR_slm <-spsurml(formula = Tformula, data = NCOVR.sf,
                        type = "slm", listw = lwncovr,
                        method = "Matrix", zero.policy = TRUE,
                        control = list(fdHess = TRUE))
 summary(NCOVRSUR_slm)
### Use of trW to compute.
 Wncovr <- as(spdep::listw2mat(lwncovr), "CsparseMatrix")
 trwncovr <- spatialreg::trW(Wncovr, type = "MC")
 impacts_NCOVRSUR_slm <- impactspsur(NCOVRSUR_slm, tr = trwncovr,
                                     R = 1000)
 summary(impacts_NCOVRSUR_slm[[1]], zstats = TRUE, short = TRUE)
 summary(impacts_NCOVRSUR_slm[[2]], zstats = TRUE, short = TRUE)
### A SUR-SDM model
 NCOVRSUR_sdm <-spsurml(formula = Tformula, data = NCOVR.sf,
                        type = "sdm", listw = lwncovr,
                        method = "Matrix", zero.policy = TRUE,
                        control = list(fdHess = TRUE))
 impacts_NCOVRSUR_sdm <- impactspsur(NCOVRSUR_sdm, tr = trwncovr,
                                     R = 1000)
 summary(impacts_NCOVRSUR_sdm[[1]], zstats = TRUE, short = TRUE)
 summary(impacts_NCOVRSUR_sdm[[2]], zstats = TRUE, short = TRUE)
```

`lmtestspsur` *Testing for the presence of spatial effects in Seemingly Unrelated Regressions*

Description

The function spsurml reports a collection of Lagrange Multipliers designed to test for the presence of different forms of spatial dependence in a *SUR* model of the "sim" type. That is, the approach of this function is from *'specific to general'*. As said, the model of the null hypothesis is the "sim" model whereas the model of the alternative depends on the effect whose omission we want to test.

The collection of Lagrange Multipliers obtained by `lmtestspsur` are standard in the literature and take into account the multivariate nature of the *SUR* model. As a limitation, note that each Multiplier tests for the omission of the same spatial effects in all the cross-sections of the *G* equations.

Usage

```
lmtestspsur(...)

## S3 method for class 'formula'
lmtestspsur(
```

```
  formula,
  data,
  listw,
  na.action,
  time = NULL,
  Tm = 1,
  zero.policy = NULL,
  R = NULL,
  b = NULL,
  ...
)

## Default S3 method:
lmtestspsur(Y, X, G, N, Tm = 1, listw, p, R = NULL, b = NULL, ...)
```

Arguments

`...` further arguments passed to the method.

`formula` An object type `Formula` similar to objects created with the package **Formula** describing the equations to be estimated in the model. This model may contain several responses (explained variables) and a varying number of regressors in each equation.

`data` An object of class data.frame or a matrix.

`listw` A `listw` object created for example by `nb2listw` from **spatialreg** package; if `nb2listw` not given, set to the same spatial weights as the `listw` argument. It can also be a spatial weighting matrix of order *(NxN)* instead of a `listw` object. Default = `NULL`.

`na.action` A function (default `options("na.action")`), can also be `na.omit` or `na.exclude` with consequences for residuals and fitted values. It may be necessary to set `zero.policy` to `TRUE` because this subsetting may create no-neighbour observations.

`time` time index for the spatial panel SUR data.

`Tm` Number of time periods.

`zero.policy` Similar to the corresponding parameter of `lagsarlm` function in **spatialreg** package. If `TRUE` assign zero to the lagged value of zones without neighbours, if `FALSE` assign `NA` - causing `spsurml()` to terminate with an error. Default = `NULL`.

`R` A row vector of order *(1xpr)* with the set of *r* linear constraints on the *beta* parameters. The *first* restriction appears in the first *p* terms, the second restriction in the next *p* terms and so on. Default = `NULL`.

`b` A column vector of order *(rx1)* with the values of the linear restrictions on the *beta* parameters. Default = `NULL`.

`Y` A column vector of order *(NTmGx1)*, with the observations of the explained variables. The ordering of the data must be (first) equation, (second) time dimension and (third) cross-sectional/spatial units. The specification of *Y* is only necessary if not available a `Formula` and a data frame. Default = `NULL`.

`X` A data matrix of order *(NTmGxp)* with the observations of the regressors. The number of covariates in the SUR model is p = $sum(p_g)$ where p_g is the number of regressors (including the intercept) in the g-th equation, g = *1,...,G*). The specification of "X" is only necessary if not available a `Formula` and a data frame. Default = `NULL`.

`G` Number of equations.

`N` Number of cross-section or spatial units

`p` Number of regressors by equation, including the intercept. *p* can be a row vector of order *(1xG)*, if the number of regressors is not the same for all the equations, or a scalar, if the *G* equations have the same number of regressors. The specification of *p* is only necessary if not available a `Formula` and a data frame.

Details

`lmtestspsur` tests for the omission of spatial effects in the "sim" version of the *SUR* model:

$$y_{tg} = X_{tg}\beta_g + u_{tg}$$

$$E[u_{tg}u_{th}'] = \sigma_{gh}I_N \quad E[u_{tg}u_{sh}'] = 0 \text{ if } tnes$$

where y_{tg} and u_{tg} are *(Nx1)* vectors, corresponding to the g-th equation and time period t; X_{tg} is the matrix of exogenous variables, of order (Nxp_g). Moreover, β_g is an unknown (p_gx1) vector of coefficients and $\sigma_{gh}I_N$ the covariance between equations *g* and *h*, being σ_{gh} and scalar and I_N the identity matrix of orden N.

The Lagrange Multipliers reported by this function are the followings:

- **LM-SUR-LAG**: Tests for the omission of a spatial lag of the explained variable in the right hand side of the "sim" equation. The model of the alternative is:

  $y_{tg} = \rho_g W y_{tg} + X_{tg}\beta_g + u_{tg}$
  The null and alternative hypotheses are:
  $H_0 : \rho_g = 0 (forallg)$ vs $H_A : \rho_g ne 0 (existg)$
- **LM-SUR-ERR**: Tests for the omission of spatial dependence in the equation of the errors of the "sim" model. The model of the alternative is:
  $y_{tg} = X_{tg}\beta_g + u_{tg}; u_{tg} = \lambda_g W u_{tg} + \epsilon_{tg}$
  The null and alternative hypotheses are:
  $H_0 : \lambda_g = 0 (forallg)$ vs $H_A : \lambda_g ne 0 (existg)$
- **LM-SUR-SARAR**: Tests for the simultaneous omission of a spatial lag of the explained variable in the right hand side of the "sim" equation and spatial dependence in the equation of the errors. The model of the alternative is:
  $y_{tg} = \rho_g W y_{tg} + X_{tg}\beta_g + u_{tg}; u_{tg} = \lambda_g W u_{tg} + \epsilon_{tg}$
  The null and alternative hypotheses are:
  $H_0 : \rho_g = \lambda_g = 0 (forallg)$ vs $H_A : \rho_g ne 0 or \lambda_g ne 0 (existg)$

- **LM*-SUR-SLM** and **LM*-SUR-SEM**: These two test are the robustifyed version of the original, raw Multipliers, **LM-SUR-SLM** and **LM-SUR-SEM**, which can be severely oversized if the respective alternative hypothesis is misspeficied (this would be the case if, for example, we are testing for omitted lags of the explained variable whereas the problem is that there is spatial dependence in the errors, or viceversa). The null and alternative hypotheses of both test are totally analogous to their twin non robust Multipliers.

Value

A list of `htest` objects each one including the Wald statistic, the corresponding p-value and the degrees of freedom.

Author(s)

Fernando López	<fernando.lopez@upct.es>
Román Mínguez	<roman.minguez@uclm.es>
Jesús Mur	<jmur@unizar.es>

References

- Mur, J., López, F., and Herrera, M. (2010). Testing for spatial effects in seemingly unrelated regressions. *Spatial Economic Analysis*, 5(4), 399-440.
- López, F.A., Mur, J., and Angulo, A. (2014). Spatial model selection strategies in a SUR framework. The case of regional productivity in EU. *Annals of Regional Science*, 53(1), 197-220.

See Also

spsurml, anova

Examples

```
#################################################
######## CROSS SECTION DATA (G>1; Tm=1) # #######
#################################################

#### Example 1: Spatial Phillips-Curve. Anselin (1988, p. 203)
rm(list = ls()) # Clean memory
data("spc")
Tformula <- WAGE83 | WAGE81 ~ UN83 + NMR83 + SMSA | UN80 + NMR80 + SMSA
lwspc <- spdep::mat2listw(Wspc, style = "W")
lmtestspsur(formula = Tformula, data = spc, listw = lwspc)

## VIP: The output of the whole set of the examples can be examined
## by executing demo(demo_lmtestspsur, package="spsur")
```

```
##################################################
######## PANEL DATA (G>1; Tm>1)          ########
##################################################

#### Example 2: Homicides & Socio-Economics (1960-90)
# Homicides and selected socio-economic characteristics for
# continental U.S. counties.
# Data for four decennial census years: 1960, 1970, 1980 and 1990.
# https://geodacenter.github.io/data-and-lab/ncovr/
data(NCOVR, package="spsur")
nbncovr <- spdep::poly2nb(NCOVR.sf, queen = TRUE)
### Some regions with no links...
lwncovr <- spdep::nb2listw(nbncovr, style = "W", zero.policy = TRUE)
### With different number of exogenous variables in each equation
Tformula <- HR70 | HR80  | HR90 ~ PS70 + UE70 | PS80 + UE80 +RD80 |
            PS90 + UE90 + RD90 + PO90
lmtestspsur(formula = Tformula, data = NCOVR.sf,
            listw = lwncovr)

#################################################################
######### PANEL DATA: TEMPORAL CORRELATIONS (G=1; Tm>1) ########
#################################################################

##### Example 3: NCOVR in panel data form
Year <- as.numeric(kronecker(c(1960,1970,1980,1990),
                   matrix(1,nrow = dim(NCOVR.sf)[1])))
HR <- c(NCOVR.sf$HR60,NCOVR.sf$HR70,NCOVR.sf$HR80,NCOVR.sf$HR90)
PS <- c(NCOVR.sf$PS60,NCOVR.sf$PS70,NCOVR.sf$PS80,NCOVR.sf$PS90)
UE <- c(NCOVR.sf$UE60,NCOVR.sf$UE70,NCOVR.sf$UE80,NCOVR.sf$UE90)
NCOVRpanel <- as.data.frame(cbind(Year,HR,PS,UE))
Tformula <- HR ~ PS + UE
lmtestspsur(formula = Tformula, data = NCOVRpanel, time = Year,
listw = lwncovr)
```

`lr_betas` *Likelihood ratio for testing homogeneity constraints on beta coefficients of the SUR equations.*

Description

Function `lr_betas` obtains a Likelihood Ratio test, LR in what follows, with the purpose of testing if some of the β coefficients in the G equations of the *SUR* model are equal. This function has a straightforward application, especially when $G = 1$, to the case of testing for the existence of structural breaks in the β parameters.

The function can test for the homogeneity of only one coefficient, of a few of them or even the homogeneity of all the slope terms. The testing procedure implies, *first*, the estimation of both a constrained and a unconstrained model and, *second*, the comparison of the log-likelihoods to compute the LR statistics.

@usage lr_betas (obj, R, b)

Usage

```
lr_betas(obj, R, b)
```

Arguments

obj	An spsur object created by spsurml, spsur3sls or spsurtime.
R	A row vector of order $(1xPr)$ showing the set of *r* linear constraints on the β parameters. The *first* restriction appears in the first *K* terms in *R*, the *second* restriction in the next *K* terms and so on.
b	A column vector of order *(rx1)* with the values of the linear restrictions on the β parameters.

Value

Object of htest including the LR statistic, the corresponding p-value, the degrees of freedom and the values of the sample estimates.

Author(s)

Fernando López	<fernando.lopez@upct.es>
Román Mínguez	<roman.minguez@uclm.es>
Jesús Mur	<jmur@unizar.es>

References

- Mur, J., López, F., and Herrera, M. (2010). Testing for spatial effects in seemingly unrelated regressions. *Spatial Economic Analysis*, 5(4), 399-440.

See Also

spsurml, spsurtime, wald_betas

Examples

```
## VIP: The output of the whole set of the examples can be examined
## by executing demo(demo_lr_betas, package="spsur")


#' #################################################
######## CROSS SECTION DATA (G>1; Tm=1)  ########
#################################################
```

```
#### Example 1: Spatial Phillips-Curve. Anselin (1988, p. 203)
rm(list = ls()) # Clean memory
data(spc)
lwspc <- spdep::mat2listw(Wspc, style = "W")
Tformula <- WAGE83 | WAGE81 ~ UN83 + NMR83 + SMSA | UN80 + NMR80 + SMSA
### H0: equal beta for SMSA in both equations.
R <- matrix(c(0,0,0,1,0,0,0,-1), nrow=1)
b <- matrix(0, ncol=1)
spcsur.slm <- spsurml(formula = Tformula, data = spc,
                      type = "slm", listw = lwspc)
summary(spcsur.slm)
lr_betas(spcsur.slm, R = R, b = b)

### Estimate restricted SUR-SLM model
spcsur.slmr <- spsurml(formula = Tformula, data = spc,
                      type = "slm", listw = lwspc,
                      R = R, b = b)
summary(spcsur.slmr)
#################################################
######## PANEL DATA (G>1; Tm>1)          ########
#################################################

##### Example 2: Homicides + Socio-Economics (1960-90)
## Homicides and selected socio-economic characteristics for continental
## U.S. counties.
## Data for four decennial census years: 1960, 1970, 1980 and 1990.
## \url{https://geodacenter.github.io/data-and-lab/ncovr/}
rm(list = ls()) # Clean memory
data(NCOVR, package="spsur")
nbncovr <- spdep::poly2nb(NCOVR.sf, queen = TRUE)
### Some regions with no links...
lwncovr <- spdep::nb2listw(nbncovr, style = "W", zero.policy = TRUE)
Tformula <- HR80  | HR90 ~ PS80 + UE80 | PS90 + UE90
### H0: equal beta for PS80 and PS90 in both equations.
NCOVRSUR.slm <-spsurml(formula = Tformula, data = NCOVR.sf,
                       type = "slm", listw = lwncovr,
                       method = "LU", zero.policy = TRUE,
                       control = list(fdHess = TRUE))
summary(NCOVRSUR.slm)
R <- matrix(c(0, 1, 0, 0, -1, 0), nrow = 1)
b <- matrix(0, ncol = 1)
lr_betas(NCOVRSUR.slm, R = R, b = b)
### Restricted model
NCOVRSUR.slmr <-spsurml(formula = Tformula, data = NCOVR.sf,
                       type = "slm", listw = lwncovr,
                       method = "LU", zero.policy = TRUE,
                       control = list(fdHess = TRUE),
                       R = R, b = b)
summary(NCOVRSUR.slmr)
#################################################################
######### PANEL DATA: TEMPORAL CORRELATIONS (nG=1; nT>1) ########
#################################################################
### Example 3: with classical panel data set. Database is
```

```
###           a spatio-temporal panel

N <- nrow(NCOVR.sf)
Tm <- 4
index_time <- rep(1:Tm, each = N)
index_indiv <- rep(1:N, Tm)
pHR <- c(NCOVR.sf$HR60, NCOVR.sf$HR70, NCOVR.sf$HR80, NCOVR.sf$HR90)
pPS <- c(NCOVR.sf$PS60, NCOVR.sf$PS70, NCOVR.sf$PS80, NCOVR.sf$PS90)
pUE <- c(NCOVR.sf$UE60, NCOVR.sf$UE70, NCOVR.sf$UE80, NCOVR.sf$UE90)
pNCOVR <- data.frame(indiv = index_indiv, time = index_time,
                     HR = pHR, PS = pPS, UE = pUE)
form_pHR <- HR ~ PS + UE
## H0: equal PS beta coefficient in equations 1, 3, and 4
## Fit with spsurtime and fit_method = "ml"...
pHR.slm <-spsurtime(formula = form_pHR, data = pNCOVR,
                    time = pNCOVR$time,
                    type = "slm", listw = lwncovr,
                    zero.policy = TRUE,
                    fit_method = "ml", method = "LU",
                    control = list(fdHess = TRUE))
summary(pHR.slm)
### H0: equal betas for PS in equations 1, 3 and 4.
R <- matrix(0, nrow = 2, ncol = 12)
## nrow = number of restrictions
## ncol = number of beta parameters
R[1, 2] <- 1; R[1, 8] <- -1 # PS beta coefficient in equations 1 equal to 3
R[2, 2] <- 1; R[2, 11] <- -1 # PS beta coefficient in equations 1 equal to 4
b <- matrix(0, nrow = 2, ncol = 1)
lr_betas(pHR.slm, R = R, b = b)
```

`methods_spsur` *Methods for class spsur*

Description

The `anova()` function provides tables of fitted spsur models including information criteria (AIC and BIC), log-likelihood and degrees of freedom of each fitted model. The argument `lrtest` allows to perform LR tests between nested models. The `plot()` function allows the user to plot both beta and spatial coefficients for all equations of the spsur model. The argument `viewplot` is used to choose between interactive or non-interactive plots. The `print()` function is used to print short tables including the values of beta and spatial coefficients as well as p-values of significance test for each coefficient. This can be used as an alternative to `summary.spsur` when a brief output is needed. The rest of methods works in the usual way.

Usage

```
## S3 method for class 'spsur'
anova(object, ..., lrtest = TRUE)
```

```
## S3 method for class 'spsur'
coef(object, ...)

## S3 method for class 'spsur'
fitted(object, ...)

## S3 method for class 'spsur'
logLik(object, ...)

## S3 method for class 'spsur'
residuals(object, ...)

## S3 method for class 'spsur'
vcov(object, ...)

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

## S3 method for class 'spsur'
plot(x, ci = 0.95, viewplot = TRUE, ...)
```

Arguments

`object`	a `spsur` object created by `spsurml`, `spsur3sls` or `spsurtime`.
`...`	further arguments passed to or from other methods.
`lrtest`	logical value to compute likelihood ratio test for nested models in 'anova' method. Default = `TRUE`
`x`	similar to `object` argument for `print()` and `plot` functions.
`digits`	number of digits to show in printed tables. Default: max(3L, getOption("digits") - 3L).
`ci`	confidence level for the intervals in 'plot' method. Default `ci = 0.95`
`viewplot`	logical value to show interactively the plots. Default = `TRUE`

Author(s)

Fernando López	`<fernando.lopez@upct.es>`
Román Mínguez	`<roman.minguez@uclm.es>`
Jesús Mur	`<jmur@unizar.es>`

Examples

```
rm(list = ls()) # Clean memory
```

```
data(spc)
Tformula <- WAGE83 | WAGE81 ~ UN83 + NMR83 + SMSA | UN80 + NMR80 + SMSA
spcsur.sim <-spsurml(formula = Tformula, data = spc, type = "sim")
## Print Table
print(spcsur.sim)

spcsur.slm <-spsurml(formula = Tformula, data = spc, type = "slm",
                     listw = Wspc)
# ANOVA table and LR test for nested models:
anova(spcsur.sim, spcsur.slm)
## Plot spatial and beta coefficients
# Interactive plot
plot(spcsur.slm)
# Non-interactive plot
if (require(gridExtra)) {
  pl <- plot(spcsur.slm, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}
```

`NCOVR.sf` *Homicides in U.S. counties*

Description

Homicides and selected socio-economic characteristics for continental U.S. counties. Data for four decennial census years: 1960, 1970, 1980 and 1990.

Usage

```
NCOVR.sf
```

Format

An spatial feature (sf) object with 3085 rows and 41 variables:

NAME County coded as a name (factor)

STATE_NAME state fips code (factor)

FIPS state fips code (factor)

SOUTH dummy variable for Southern counties (South = 1)

HR60, HR70, HR80, HR90 homicide rate per 100,000 (1960, 1970, 1980, 1990)

HC60, HC70, HC80, HC90 homicide count, three year average centered on 1960, 1970, 1980, 1990

PO60, PO70, PO80, PO90 county population, 1960, 1970, 1980, 1990

RD60, RD70, RD80, RD90 resource deprivation 1960, 1970, 1980, 1990 (principal component, see Codebook for details)

PS60, PS70, PS80, PS90 population structure 1960, 1970, 1980, 1990 (principal component, see Codebook for details)

UE60, UE70, UE80, UE90 unemployment rate 1960, 1970, 1980, 1990

DV60, DV70, DV80, DV90 divorce rate 1960, 1970, 1980, 1990 (% males over 14 divorced)

MA60, MA70, MA80, MA90 median age 1960, 1970, 1980, 1990

FP59, FP69, FP79, FP89 % families below poverty 1960, 1970, 1980, 1990 (see Codebook for details)

geometry Multipolygon geometry of the spatial feature object

Source

S. Messner, L. Anselin, D. Hawkins, G. Deane, S. Tolnay, R. Baller (2000). An Atlas of the Spatial Patterning of County-Level Homicide, 1960-1990. Pittsburgh, PA, National Consortium on Violence Research (NCOVR) https://geodacenter.github.io/data-and-lab/ncovr/

References

- Baller, R., L. Anselin, S. Messner, G. Deane and D. Hawkins (2001). Structural covariates of US county homicide rates: incorporating spatial effects. *Criminology* 39, 561-590.

`print.summary.spsur` *Print method for objects of class summary.spsur.*

Description

Print method for objects of class summary.spsur.

Usage

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

Arguments

`x`	object of class *summary.spsur*.
`digits`	number of digits to show in printed tables. Default: max(3L, getOption("digits") - 3L).
`...`	further arguments passed to or from other methods.

Author(s)

Fernando López	`<fernando.lopez@upct.es>`
Román Mínguez	`<roman.minguez@uclm.es>`
Jesús Mur	`<jmur@unizar.es>`

See Also

summary.spsur.

Examples

```
# See examples for \code{\link{spsurml}} or
# \code{\link{spsur3sls}} functions.
```

---

spain.covid *Within/Exit mobility index and incidence COVID-19 at Spain provinces*

Description

Weekly within/exit mobility indices and COVID-19 incidence in Spain at provincial level. A total of 17 weeks from February 21 to May 21. Every week starts on a Friday and ends on the following Thursday. All travels are expressed with respect to the pre-COVID week of February 14th-20th, 2020 (week0). A value lower than 1 indicate a reduction of the mobility. Upper values than 1 indicate an increase in the mobility with respect to the reference week (week0).

Usage

```
spain.covid
```

Format

A data frame with 850 rows and 11 variables:

province Province name coded as a factor

indiv Province coded as a number

time Number of week afther pre-COVID (week0: February 14th-20th, 2020). time=1 (week1, February 21th-27th); time=2 (week2,February 28th-March 5th);....

Within Mobility index within of the province. See details for a formal definition

Exits Mobility index of exits of the province. See details for a formal definition

Emergence Dummy variable. 1 if the Emergence State ("Estado de Alarma") is active in the week "time" but economic activity of essential services are allowed. 0 in anoter case. The Emergence State was active in Spain from February 14th, 2020 to May 21th, 2020

EmergenceTotal Dummy variable. 1 if the Emergence State ("Estado de Alarma") is active in the week "time" and economic activity of essential services are not allowed. 0 in anoter case

Old65 Percentage of population aged 65 and older in the province

Density Inhabitants (in thousands) per km^2 in the province

Essential Percentage of firms in the province "indiv" with essential activities (food, health) over the total of firms in the province "indiv" in the province (e.g., food, heath and some economic subsectors of industry and construction)

Incidence Weekly incidence in the week "time-1" in logs

Details

- Mobility

  The mobility indices Within and Exits has been obtain as ratio of the total number of weekly travels in reference to the total number of travels in the week of reference (week0).
  In particular,

  $$Within = \frac{\text{Number of travels within the province 'indiv' in the week 'time'}}{\text{Number of travels within the province 'indiv' in the reference week (week0)}}$$

  $$Exits = \frac{\text{Number of travels with origin in the province 'indiv' and arrival to another province in the week 'time'}}{\text{Number of travels with origin in the province 'indiv' and arrival to another province in the reference week (week0)}}$$

  A 'travel' is a displacement from an origin to a destination of at least 500m. A travel can have several stages. These stages of the same travel are calculated based on the duration of the intermediate stop. For example , if I move from origin A to destination B with a stop at a point C of long duration, it is considered two travels, but if the stop is short, it is considered a single travel. For example, I can go by train from Madrid to Alicante, and there takes a bus to Benidorm. The travel will be one (Madrid-Benidorm). If I do the same but the stop in Alicante is long (or an overnight stay, for example), two travels will be considered (Madrid-Alicante, and Alicante-Benidorm). Similarly, if I go by car from Madrid to Alicante and stop for 15 minutes to take a coffee, it is also considered only one travel and not two. The travels considered in this study are always from 500m, due to the limitation of source data that is based on mobile telephony and its antennas. But one travel can be 600 meters or 600km.

- Incidence

  $$Incidence = \log\left(\frac{\text{total diagnostic cases of COVID-19, PCR test in the week 'time' at the province 'indiv'}}{\text{total population in the province 'indiv'}}\right)$$

- Essential activities

  Essential activities. Economic activities whose activities are essential for the population. By example, essential activities are healthcare, food supply, State security, media and communication, refuse collection, management and public transport, etc. Non essential activities are by example, restaurants, hotels, hairdressers, etc. A full list in the Spanish official bulletin

  https://www.boe.es/boe/dias/2020/03/29/pdfs/BOE-A-2020-4166.pdf

Source

Ministerio de Transportes, Movilidad y Agenda Urbana. Spain Government.
https://www.mitma.gob.es/ministerio/covid-19/evolucion-movilidad-big-data
The National Statistics Institute

https://www.ine.es/en/index.htm
Instituto de Salud Carlos III
https://cnecovid.isciii.es/covid19/

`spain.covid.sf` *Spain geometry*

Description

A sf object with the Spanish geometry

Usage

```
spain.covid.sf
```

Format

A sf object with the Spanish geometry

PROVINCIA province name coded as a factor

ID_INE province coded as a number

`spc` *A classical Spatial Phillips-Curve*

Description

A data set from Anselin (1988, p. 203-2011) used to estimate a Spatial Phillips-Curve for 25 counties in South-West Ohio for two time periods (1981 and 1983).

Usage

```
spc
```

Format

A data frame with 25 rows and 10 variables:

COUNTY County coded as a name.

WAGE83 Changes in wage rates for 1983.

UN83 Inverse unemployment rate in 1983.

NMR83 Net migration rate 1983.

SMSA Dummy variable to identify counties defined as Standard Metropolitan Statistical Areas (SMSA = 1).

WAGE82 Changes in wage rates for 1982.

WAGE81 Changes in wage rates for 1981.

UN80 Inverse unemployment rate in 1980.

NMR80 Net migration rate 1983.

WAGE80 changes in wage rates.

Source

Anselin (1988, p. 203-211)

References

- Anselin, L. (1988). *Spatial Econometrics: Methods and Models*. Springer Science & Business Media.

`spsur3sls` *Three Stages Least Squares estimation,3sls, of spatial SUR models.*

Description

The function estimates spatial SUR models using three stages least squares, where the instruments are obtained from the spatial lags of the X variables, assumed to be exogenous. The number of equations, time periods and spatial units is not restricted. The user can choose between a Spatial Durbin Model or a Spatial Lag Model, as described below. The estimation procedure allows for the introduction of linear restrictions on the β parameters associated to the regressors.

Usage

```
spsur3sls (formula = NULL, data = NULL, na.action,
                  R = NULL, b = NULL, listw = NULL,
                  zero.policy = NULL, X= NULL, Y = NULL, G = NULL,
                  N = NULL, Tm = NULL, p = NULL,
                  type = "slm", Durbin = NULL, maxlagW = NULL,
                  trace = TRUE)
```

Arguments

`formula`	An object type `Formula` similar to objects created with the package **Formula** describing the equations to be estimated in the model. This model may contain several responses (explained variables) and a varying number of regressors in each equation.
`data`	An object of class data.frame or a matrix.
`na.action`	A function (default `options("na.action")`), can also be `na.omit` or `na.exclude` with consequences for residuals and fitted values. It may be necessary to set `zero.policy` to `TRUE` because this subsetting may create no-neighbour observations.

R
: A row vector of order *(1xpr)* with the set of *r* linear constraints on the *beta* parameters. The *first* restriction appears in the first *p* terms, the second restriction in the next *p* terms and so on. Default = `NULL`.

b
: A column vector of order *(rx1)* with the values of the linear restrictions on the *beta* parameters. Default = `NULL`.

listw
: A `listw` object created for example by `nb2listw` from **spatialreg** package; if `nb2listw` not given, set to the same spatial weights as the `listw` argument. It can also be a spatial weighting matrix of order *(NxN)* instead of a `listw` object. Default = `NULL`.

zero.policy
: Similar to the corresponding parameter of `lagsarlm` function in **spatialreg** package. If `TRUE` assign zero to the lagged value of zones without neighbours, if `FALSE` assign `NA` - causing `spsurml()` to terminate with an error. Default = `NULL`.

X
: A data matrix of order *(NTmGxp)* with the observations of the regressors. The number of covariates in the SUR model is *p* = $sum(p_g)$ where p_g is the number of regressors (including the intercept) in the g-th equation, *g* = *1,...,G*). The specification of "X" is only necessary if not available a `Formula` and a data frame. Default = `NULL`.

Y
: A column vector of order *(NTmGx1)*, with the observations of the explained variables. The ordering of the data must be (first) equation, (second) time dimension and (third) cross-sectional/spatial units. The specification of *Y* is only necessary if not available a `Formula` and a data frame. Default = `NULL`.

G
: Number of equations.

N
: Number of cross-section or spatial units

Tm
: Number of time periods.

p
: Number of regressors by equation, including the intercept. *p* can be a row vector of order *(1xG)*, if the number of regressors is not the same for all the equations, or a scalar, if the *G* equations have the same number of regressors. The specification of *p* is only necessary if not available a `Formula` and a data frame.

type
: Type of spatial model, restricted to cases where lags of the explainded variable appear in the righ hand side of the equations. There are two possibilities: "slm" or "sdm". Default = "slm".

Durbin
: If a formula object and model is type "sdm" the subset of explanatory variables to lag for each equation.

maxlagW
: Maximum spatial lag order of the regressors employed to produce spatial instruments for the spatial lags of the explained variables. Default = 2. Note that in case of `type="sdm"`, the default value for `maxlagW` is set to 3 because the first lag of the regressors, WX_{tg}, can not be used as spatial instruments.

trace
: A logical value to show intermediate results during the estimation process. Default = `TRUE`.

Details

spsur3sls can be used to estimate two groups of spatial models:

- "slm": SUR model with spatial lags of the endogenous in the right hand side of the equations

$$y_{tg} = \rho_g W y_{tg} + X_{tg}\beta_g + \epsilon_{tg}$$

- "sdm": SUR model of the Spatial Durbin type

$$y_{tg} = \rho_g W y_{tg} + X_{tg}\beta_g + W X_{tg}\theta_g + \epsilon_{tg}$$

where y_{tg} and ϵ_{tg} are *(Nx1)* vectors, corresponding to the g-th equation and time period t; X_{tg} is the matrix of regressors, of order *(Nxp_g)*. Moreover, ρ_g is a spatial coefficient and *W* is a *(NxN)* spatial weighting matrix.

By default, the input of this function is an object created with `Formula` and a data frame. However, *spsur3sls* also allows for the direct especification of vector *Y* and matrix *X*, with the explained variables and regressors respectively, as inputs (these terms may be the result, for example, of `dgp_spsur`).

`spsur3sls` is a Least-Squares procedure in three-stages designed to circumvent the endogeneity problems due to the presence of spatial lags of the explained variable in the right hand side of the equations do the SUR. The instruments are produced internally by `spsur3sls` using a sequence of spatial lags of the *X* variables, which are assumed to be exogenous. The user must define the number of (spatial) instruments to be used in the procedure, through the argument `maxlagW` (i.e. maxlagW=3). Then, the collection of instruments generated is $[WX_{tg}; W*WX_{tg}; W*W*WX_{tg}]$. In the case of a *SDM*, the first lag of the *X* matrix already is in the equation and cannot be used as instrument. In the example above, the list of instruments for a *SDM* model would be $[W^2X_{tg}; W^3X_{tg}]$.

The *first* stage of the procedure consists in the least squares of the *Y* variables on the set of instruments. From this estimation, the procedure retains the estimates of *Y* in the so-called *Yls* variables. In the *second* stage, the *Y* variables that appear in the right hand side of the equation are substituted by *Yls* and the SUR model is estimated by Least Squares. The *third* stage improves the estimates of the second stage through a Feasible Generalized Least Squares estimation of the parameters of the model, using the residuals of the *second* stage to estimate the *Sigma* matrix.

The arguments *R* and *b* allows to introduce linear restrictions on the *beta* coefficients of the *G* equations. `spsur3sls`, first, introduces the linear restrictions in the SUR model and builds, internally, the corresponding constrained SUR model. Then, the function estimates the restricted model which is shown in the output. The function does not compute the unconstrained model nor test for the linear restrictions. The user may ask for the unconstrained estimation using another `spsurml` estimation. Moreover, the function `wald_betas` obtains the Wald test of a set of linear restrictions for an object created previously by `spsurml` or `spsur3sls`.

Value

Object of `spsur` class with the output of the three-stages least-squares estimation of the specified spatial model. A list with:

`call`	Matched call.
`type`	Type of model specified.
`Durbin`	Value of `Durbin` argument.
`coefficients`	Estimated coefficients for the regressors.

`deltas`	Estimated spatial coefficients.
`rest.se`	Estimated standard errors for the estimates of β coefficients.
`deltas.se`	Estimated standard errors for the estimates of the spatial coefficients.
`resvar`	Estimated covariance matrix for the estimates of *beta's* and spatial coefficients.
`R2`	Coefficient of determination for each equation, obtained as the squared of the correlation coefficient betwee
`Sigma`	Estimated covariance matrix for the residuals of the G equations.
`residuals`	Residuals of the model.
`df.residuals`	Degrees of freedom for the residuals.
`fitted.values`	Estimated values for the dependent variables.
`G`	Number of equations.
`N`	Number of cross-sections or spatial units.
`Tm`	Number of time periods.
`p`	Number of regressors by equation (including intercepts).
`Y`	Vector Y of the explained variables of the SUR model.
`X`	Matrix X of the regressors of the SUR model.
`W`	Spatial weighting matrix.
`zero.policy`	Logical value of `zero.policy` .
`listw_style`	Style of neighborhood matrix `W`.

Author(s)

Fernando López	`<fernando.lopez@upct.es>`
Román Mínguez	`<roman.minguez@uclm.es>`
Jesús Mur	`<jmur@unizar.es>`

References

- Mur, J., López, F., and Herrera, M. (2010). Testing for spatial effects in seemingly unrelated regressions. *Spatial Economic Analysis*, 5(4), 399-440.
- López, F.A., Mur, J., and Angulo, A. (2014). Spatial model selection strategies in a SUR framework. The case of regional productivity in EU. *Annals of Regional Science*, 53(1), 197-220.

See Also

`spsurml`, `stsls`, `wald_betas`

Examples

```
#################################################
######## CROSS SECTION DATA (G=1; Tm>1) ########
#################################################
```

```
#### Example 1: Spatial Phillips-Curve. Anselin (1988, p. 203)
## A SUR model without spatial effects
rm(list = ls()) # Clean memory
data(spc)
lwspc <- spdep::mat2listw(Wspc, style = "W")
Tformula <- WAGE83 | WAGE81 ~ UN83 + NMR83 + SMSA | UN80 + NMR80 + SMSA

## A SUR-SLM model (3SLS Estimation)
spcsur_slm_3sls <-spsur3sls(formula = Tformula, data = spc,
                            type = "slm", listw = lwspc)
summary(spcsur_slm_3sls)
print(spcsur_slm_3sls)

if (require(gridExtra)) {
  pl <- plot(spcsur_slm_3sls, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}

## VIP: The output of the whole set of the examples can be examined
## by executing demo(demo_spsur3sls, package="spsur")


## A SUR-SDM model (3SLS Estimation)
 spcsur_sdm_3sls <-spsur3sls(formula = Tformula, data = spc,
                             type = "sdm", listw = lwspc)
 summary(spcsur_sdm_3sls)
if (require(gridExtra)) {
  pl <- plot(spcsur_sdm_3sls, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}
rm(spcsur_sdm_3sls)

# A SUR-SDM model with different spatial lags in each equation
 TformulaD <-  ~ UN83 + NMR83 + SMSA | UN80 + NMR80
 spcsur_sdm2_3sls <-spsur3sls(formula = Tformula, data = spc,
                             type = "sdm", listw = lwspc,
                             Durbin = TformulaD)
 summary(spcsur_sdm2_3sls)
if (require(gridExtra)) {
  pl <- plot(spcsur_sdm2_3sls, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}
rm(spcsur_sdm2_3sls)


#################################################
######## PANEL DATA (G>1; Tm>1)         #########
#################################################

#### Example 2: Homicides + Socio-Economics (1960-90)
```

```
# Homicides and selected socio-economic characteristics for continental
# U.S. counties.
# Data for four decennial census years: 1960, 1970, 1980 and 1990.
# https://geodacenter.github.io/data-and-lab/ncovr/


rm(list = ls()) # Clean memory
## Read NCOVR.sf object
 data(NCOVR, package = "spsur")
 nbncovr <- spdep::poly2nb(NCOVR.sf, queen = TRUE)
## Some regions with no links...
 lwncovr <- spdep::nb2listw(nbncovr, style = "W", zero.policy = TRUE)
 Tformula <- HR80  | HR90 ~ PS80 + UE80 | PS90 + UE90
## A SUR-SLM model
 NCOVRSUR_slm_3sls <-spsur3sls(formula = Tformula, data = NCOVR.sf,
                              type = "slm", zero.policy = TRUE,
                              listw = lwncovr, maxlagW = 2,
                              trace = FALSE)
 summary(NCOVRSUR_slm_3sls)
if (require(gridExtra)) {
  pl <- plot(NCOVRSUR_slm_3sls, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}
rm(NCOVRSUR_slm_3sls)
```

---

spsurml *Maximum likelihood estimation of spatial SUR model.*

Description

This function estimates spatial SUR models using maximum-likelihood methods.The number of equations, time periods and cross-sectional units is not restricted.The user can choose between different spatial specifications as described below. The estimation procedure allows for the introduction of linear restrictions on the β parameters associated to the regressors.

Usage

```
spsurml(formula = NULL, data = NULL, na.action,
                listw = NULL, type = "sim", Durbin = NULL,
                method = "eigen", zero.policy = NULL, interval = NULL,
                trs = NULL, R = NULL, b = NULL, X = NULL, Y = NULL,
                G = NULL, N = NULL, Tm = NULL,p = NULL,
                control = list() )
```

Arguments

- `formula` — An object type `Formula` similar to objects created with the package **Formula** describing the equations to be estimated in the model. This model may contain several responses (explained variables) and a varying number of regressors in each equation.
- `data` — An object of class data.frame or a matrix.
- `na.action` — A function (default `options("na.action")`), can also be `na.omit` or `na.exclude` with consequences for residuals and fitted values. It may be necessary to set `zero.policy` to `TRUE` because this subsetting may create no-neighbour observations.
- `listw` — A `listw` object created for example by `nb2listw` from **spatialreg** package; if `nb2listw` not given, set to the same spatial weights as the `listw` argument. It can also be a spatial weighting matrix of order *(NxN)* instead of a `listw` object. Default = `NULL`.
- `type` — Type of spatial model specification: "sim", "slx", "slm", "sem", "sdm", "sdem", "sarar" or "gnm". Default = "sim".
- `Durbin` — If a formula object and model is type "sdm", "sdem" or "slx" the subset of explanatory variables to lag for each equation.
- `method` — Similar to the corresponding parameter of `lagsarlm` function in **spatialreg** package. "eigen" (default) - the Jacobian is computed as the product of (1 - rho*eigenvalue) using `eigenw`, and "spam" or "Matrix_J" for strictly symmetric weights lists of styles "B" and "C", or made symmetric by similarity (Ord, 1975, Appendix C) if possible for styles "W" and "S", using code from the spam or Matrix packages to calculate the determinant; "Matrix" and "spam_update" provide updating Cholesky decomposition methods; "LU" provides an alternative sparse matrix decomposition approach. In addition, there are "Chebyshev" and Monte Carlo "MC" approximate log-determinant methods; the Smirnov/Anselin (2009) trace approximation is available as "moments". Three methods: "SE_classic", "SE_whichMin", and "SE_interp" are provided experimentally, the first to attempt to emulate the behaviour of Spatial Econometrics toolbox ML fitting functions. All use grids of log determinant values, and the latter two attempt to ameliorate some features of "SE_classic".
- `zero.policy` — Similar to the corresponding parameter of `lagsarlm` function in **spatialreg** package. If `TRUE` assign zero to the lagged value of zones without neighbours, if `FALSE` assign `NA` - causing `spsurml()` to terminate with an error. Default = `NULL`.
- `interval` — Search interval for autoregressive parameter. Default = `NULL`.
- `trs` — Similar to the corresponding parameter of `lagsarlm` function in **spatialreg** package. Default `NULL`, if given, a vector of powered spatial weights matrix traces output by `trW`.
- `R` — A row vector of order *(1xpr)* with the set of *r* linear constraints on the *beta* parameters. The *first* restriction appears in the first *p* terms, the second restriction in the next *p* terms and so on. Default = `NULL`.
- `b` — A column vector of order *(rx1)* with the values of the linear restrictions on the *beta* parameters. Default = `NULL`.

`X`	A data matrix of order *(NTmGxp)* with the observations of the regressors. The number of covariates in the SUR model is *p* = $sum(p_g)$ where p_g is the number of regressors (including the intercept) in the g-th equation, g = *1,...,G*). The specification of "X" is only necessary if not available a `Formula` and a data frame. Default = `NULL`.
`Y`	A column vector of order *(NTmGx1)*, with the observations of the explained variables. The ordering of the data must be (first) equation, (second) time dimension and (third) cross-sectional/spatial units. The specification of *Y* is only necessary if not available a `Formula` and a data frame. Default = `NULL`.
`G`	Number of equations.
`N`	Number of cross-section or spatial units
`Tm`	Number of time periods.
`p`	Number of regressors by equation, including the intercept. *p* can be a row vector of order *(1xG)*, if the number of regressors is not the same for all the equations, or a scalar, if the *G* equations have the same number of regressors. The specification of *p* is only necessary if not available a `Formula` and a data frame.
`control`	List of additional control arguments.

Details

The list of (spatial) models that can be estimated with the *spsurml* function are:

- "sim": SUR model with no spatial effects

$$y_{tg} = X_{tg}\beta_g + \epsilon_{tg}$$

- "slx": SUR model with spatial lags of the regressors

$$y_{tg} = X_{tg}\beta_g + WX_{tg}\theta_g + \epsilon_{tg}$$

- "slm": SUR model with spatial lags of the explained variables

$$y_{tg} = \rho_g Wy_{tg} + X_{tg}\beta_g + \epsilon_{tg}$$

- "sem": SUR model with spatial errors

$$y_{tg} = X_{tg}\beta_g + u_{tg}$$

$$u_{tg} = \lambda_g Wu_{tg} + \epsilon_{tg}$$

- "sdm": SUR model of the Spatial Durbin type

$$y_{tg} = \rho_g Wy_{tg} + X_{tt}\beta_g + WX_{tg}\theta_g + \epsilon_{tg}$$

- "sdem": SUR model with spatial lags of the regressors and spatial errors

$$y_{tg} = X_{tg}\beta_g + WX_{tg}\theta_g + u_{tg}$$

$$u_{tg} = \lambda_g Wu_{tg} + \epsilon_{tg}$$

- "sarar": SUR model with spatial lags of the explained variables and spatial errors

$$y_{tg} = \rho_g W y_{tg} + X_{tg}\beta_g + u_{tg}$$

$$u_{tg} = \lambda_g W u_{tg} + \epsilon_{tg}$$

- "gnm": SUR model with spatial lags of the explained variables, regressors and spatial errors

$$y_{tg} = \rho_g W y_{tg} + X_{tg}\beta_g + W X_{tg}\theta_g + u_{tg}$$

$$u_{tg} = \lambda_g W u_{tg} + \epsilon_{tg}$$

Value

Object of `spsur` class with the output of the maximum-likelihood estimation of the specified spatial SUR model. A list with:

`call`	Matched call.
`type`	Type of model specified.
`method`	Value of `method` argument to compute the Jacobian
`Durbin`	Value of `Durbin` argument.
`coefficients`	Estimated coefficients for the regressors.
`deltas`	Estimated spatial coefficients.
`rest.se`	Estimated standard errors for the estimates of *beta*.
`deltas.se`	Estimated standard errors for the estimates of the spatial coefficients (`deltas`).
`resvar`	Estimated covariance matrix for the estimates of *beta's* and spatial coefficients (`deltas`).
`LL`	Value of the likelihood function at the maximum-likelihood estimates.
`R2`	Coefficient of determination for each equation, obtained as the squared of the correlation coefficient betwee
`Sigma`	Estimated covariance matrix for the residuals of the *G* equations.
`fdHess`	Logical value of `fdHess` argument when computing numerical covariances.
`residuals`	Residuals of the model.
`df.residuals`	Degrees of freedom for the residuals.
`fitted.values`	Estimated values for the dependent variables.
`BP`	Value of the Breusch-Pagan statistic to test the null hypothesis of diagonality among the errors of the *G* equ
`LMM`	Marginal Lagrange Multipliers, LM($\rho|\lambda$) and LM($\lambda|\rho$), to test for omitted spatial effects in the specification
`G`	Number of equations.
`N`	Number of cross-sections or spatial units.
`Tm`	Number of time periods.
`p`	Number of regressors by equation (including intercepts).
`Y`	Vector *Y* of the explained variables of the SUR model.
`X`	Matrix *X* of the regressors of the SUR model.
`W`	Spatial weighting matrix.
`zero.policy`	Logical value of `zero.policy` .
`interval`	Search interval for spatial parameter.
`listw_style`	Style of neighborhood matrix `W`.
`trs`	Either `NULL` or vector of powered spatial weights matrix traces output by `trW`.
`insert`	Logical value to check if `is.null(trs)`.

Control arguments

tol	Numerical value for the tolerance for the estimation algorithm until convergence. Default = 1e-3.
maxit	Maximum number of iterations until convergence; it must be an integer value. Default = 200.
trace	A logical value to show intermediate results during the estimation process. Default = TRUE.
fdHess	Compute variance-covariance matrix using the numerical hessian. Suited for large samples. Default = FALSE
Imult	default 2; used for preparing the Cholesky decompositions for updating in the Jacobian function
super	if NULL (default), set to FALSE to use a simplicial decomposition for the sparse Cholesky decomposition and met
cheb_q	default 5; highest power of the approximating polynomial for the Chebyshev approximation
MC_p	default 16; number of random variates
MC_m	default 30; number of products of random variates matrix and spatial weights matrix
spamPivot	default "MMD", alternative "RCM"
in_coef	default 0.1, coefficient value for initial Cholesky decomposition in "spam_update"
type	default "MC", used with method "moments"; alternatives "mult" and "moments", for use if trs is missing
correct	default TRUE, used with method "moments" to compute the Smirnov/Anselin correction term
trunc	default TRUE, used with method "moments" to truncate the Smirnov/Anselin correction term
SE_method	default "LU", may be "MC"
nrho	default 200, as in SE toolbox; the size of the first stage lndet grid; it may be reduced to for example 40
interpn	default 2000, as in SE toolbox; the size of the second stage lndet grid
SElndet	default NULL, may be used to pass a pre-computed SE toolbox style matrix of coefficients and their lndet values t
LU_order	default FALSE; used in "LU_prepermutate", note warnings given for lu method
pre_eig	default NULL; may be used to pass a pre-computed vector of eigenvalues

Author(s)

Fernando López	<fernando.lopez@upct.es>
Román Mínguez	<roman.minguez@uclm.es>
Jesús Mur	<jmur@unizar.es>

References

- Anselin, L. (1988). *Spatial econometrics: methods and models.* Dordrecht: Kluwer
- Bivand, R.S. and Piras G. (2015). Comparing Implementations of Estimation Methods for Spatial Econometrics. *Journal of Statistical Software*, 63(18), 1-36. https://www.jstatsoft.org/v63/i18/.
- Bivand, R. S., Hauke, J., and Kossowski, T. (2013). Computing the Jacobian in Gaussian spatial autoregressive models: An illustrated comparison of available methods. *Geographical Analysis*, 45(2), 150-179.
- Breusch T, Pagan A (1980). The Lagrange multiplier test and its applications to model specification in econometrics. *Rev Econ Stud* 47: 239-254
- Cliff, A. D. and Ord, J. K. (1981). *Spatial processes: Models and applications*, Pion.

- LeSage J and Pace, R.K. (2009). *Introduction to Spatial Econometrics.* CRC Press, Boca Raton.
- López, F.A., Mur, J., and Angulo, A. (2014). Spatial model selection strategies in a SUR framework. The case of regional productivity in EU. *Annals of Regional Science*, 53(1), 197-220.
- Mur, J., López, F., and Herrera, M. (2010). Testing for spatial effects in seemingly unrelated regressions. *Spatial Economic Analysis*, 5(4), 399-440.
- Ord, J. K. (1975). Estimation methods for models of spatial interaction, *Journal of the American Statistical Association*, 70, 120-126;

See Also

spsur3sls, lagsarlm, lmtestspsur, wald_betas, lrtest

Examples

```
#################################################
######## CROSS SECTION DATA (G>1; Tm=1) ########
#################################################

#### Example 1: Spatial Phillips-Curve. Anselin (1988, p. 203)
rm(list = ls()) # Clean memory
data(spc)
Tformula <- WAGE83 | WAGE81 ~ UN83 + NMR83 + SMSA | UN80 + NMR80 + SMSA
spcsur.sim <-spsurml(formula = Tformula, data = spc, type = "sim")
summary(spcsur.sim)
# All the coefficients in a single table.
print(spcsur.sim)
# Plot of the coefficients of each equation in different graphs
plot(spcsur.sim)

## A SUR-SLX model
## (listw argument can be either a matrix or a listw object )
spcsur.slx <-spsurml(formula = Tformula, data = spc, type = "slx",
  listw = Wspc)
summary(spcsur.slx)
# All the coefficients in a single table.
print(spcsur.slx)
# Plot of the coefficients in a single graph
if (require(gridExtra)) {
  pl <- plot(spcsur.slx, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
                        nrow = 2)
}

## VIP: The output of the whole set of the examples can be examined
## by executing demo(demo_spsurml, package="spsur")


### A SUR-SLM model
```

```
spcsur.slm <-spsurml(formula = Tformula, data = spc, type = "slm",
                     listw = Wspc)
summary(spcsur.slm)
if (require(gridExtra)) {
  pl <- plot(spcsur.slm, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}

#' ### A SUR-SEM model
spcsur.sem <-spsurml(formula = Tformula, data = spc, type = "sem",
                     listw = Wspc)
summary(spcsur.sem)
print(spcsur.sem)
if (require(gridExtra)) {
  pl <- plot(spcsur.sem, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}

### A SUR-SDM model
spcsur.sdm <-spsurml(formula = Tformula, data = spc, type = "sdm",
                     listw = Wspc)
summary(spcsur.sdm)
print(spcsur.sdm)
if (require(gridExtra)) {
  pl <- plot(spcsur.sdm, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}

## A SUR-SDM model with different spatial lags in each equation
TformulaD <- ~ UN83 + NMR83 + SMSA | UN80
spcsur.sdm2 <-spsurml(formula = Tformula, data = spc, type = "sdm",
                      listw = Wspc, Durbin = TformulaD)
summary(spcsur.sdm2)
if (require(gridExtra)) {
  pl <- plot(spcsur.sdm2, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}
### A SUR-SDEM model
spcsur.sdem <-spsurml(formula = Tformula, data = spc, type = "sdem",
                      listw = Wspc)
print(spcsur.sdem)
if (require(gridExtra)) {
  pl <- plot(spcsur.sdem, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}

### A SUR-SARAR model
spcsur.sarar <-spsurml(formula = Tformula, data = spc, type = "sarar",
```

```
                       listw = Wspc, control = list(tol = 0.1))
print(spcsur.sarar)
if (require(gridExtra)) {
  pl <- plot(spcsur.sarar, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}

### A SUR-GNM model
spcsur.gnm <-spsurml(formula = Tformula, data = spc, type = "gnm",
                       listw = Wspc, control = list(tol = 0.1))
print(spcsur.gnm)
if (require(gridExtra)) {
  pl <- plot(spcsur.gnm, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}

## A A SUR-GNM model model with different spatial lags in each equation
TformulaD <- ~ UN83 + NMR83 + SMSA | UN80
spcsur.gnm2 <-spsurml(formula = Tformula, data = spc, type = "gnm",
                       listw = Wspc, Durbin = TformulaD,
                       control = list(tol = 0.1))
print(spcsur.gnm2)
if (require(gridExtra)) {
  pl <- plot(spcsur.gnm2, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}



##################################################
#########  G=1; Tm>1         ########
##################################################
#
##### Example 2: Homicides + Socio-Economics (1960-90)
## Homicides and selected socio-economic characteristics for continental
## U.S. counties.
## Data for four decennial census years: 1960, 1970, 1980 and 1990.
## \url{https://geodacenter.github.io/data-and-lab/ncovr/}


### It usually requires 1-2 minutes maximum...
rm(list = ls()) # Clean memory
### Read NCOVR.sf object
data(NCOVR, package = "spsur")
nbncovr <- spdep::poly2nb(NCOVR.sf, queen = TRUE)
### Some regions with no links...
lwncovr <- spdep::nb2listw(nbncovr, style = "W", zero.policy = TRUE)
Tformula <- HR80  | HR90 ~ PS80 + UE80 | PS90 + UE90
### A SUR-SIM model
NCOVRSUR.sim <-spsurml(formula = Tformula, data = NCOVR.sf, type = "sim")
```

```
summary(NCOVRSUR.sim)
if (require(gridExtra)) {
  pl <- plot(NCOVRSUR.sim, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]], nrow = 3)
}
### A SUR-SLX model
NCOVRSUR.slx <-spsurml(formula = Tformula, data = NCOVR.sf, type = "slx",
                       listw = lwncovr, zero.policy = TRUE)
print(NCOVRSUR.slx)
if (require(gridExtra)) {
  pl <- plot(NCOVRSUR.slx, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]], nrow = 2)
}

### A SUR-SLM model
### method = "Matrix" (Cholesky) instead of "eigen"
### (fdHess = TRUE to compute numerical covariances )
NCOVRSUR.slm <-spsurml(formula = Tformula, data = NCOVR.sf,
                       type = "slm", listw = lwncovr, method = "Matrix",
                       zero.policy = TRUE, control = list(fdHess = TRUE))
summary(NCOVRSUR.slm)

if (require(gridExtra)) {
  pl <- plot(NCOVRSUR.slm, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}

# LR test for nested models
#lrtest(NCOVRSUR.sim, NCOVRSUR.slm)

### A SUR-SDM model with different spatial lags in each equation
### Analytical covariances (default)
TformulaD <- ~ PS80 + UE80 | PS90
NCOVRSUR.sdm <-spsurml(formula = Tformula, data = NCOVR.sf,
                       type = "sdm", listw = lwncovr, method = "Matrix",
                       Durbin = TformulaD, zero.policy = TRUE)
print(NCOVRSUR.sdm)
if (require(gridExtra)) {
  pl <- plot(NCOVRSUR.sdm, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}
### A SUR-SEM model
NCOVRSUR.sem <- spsurml(formula = Tformula, data = NCOVR.sf,
                        type = "sem", listw = lwncovr, method = "Matrix",
                        zero.policy = TRUE, control = list(fdHess = TRUE))
print(NCOVRSUR.sem)
if (require(gridExtra)) {
  pl <- plot(NCOVRSUR.sem, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}
```

```
### A SUR-SDEM model
NCOVRSUR.sdem <-spsurml(formula = Tformula, data = NCOVR.sf,
                        type = "sdem", listw = lwncovr, method = "Matrix",
                        zero.policy = TRUE, control = list(fdHess = TRUE))
print(NCOVRSUR.sdem)
if (require(gridExtra)) {
  pl <- plot(NCOVRSUR.sdem, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$pldeltas, nrow = 3)
}


################################################
######### SUR with G>1; Tm>1  ###
################################################
##### Reshape NCOVR in panel format

N <- nrow(NCOVR.sf)
Tm <- 4
index_time <- rep(1:Tm, each = N)
index_indiv <- rep(1:N, Tm)
pHR <- c(NCOVR.sf$HR60, NCOVR.sf$HR70, NCOVR.sf$HR80, NCOVR.sf$HR90)
pPS <- c(NCOVR.sf$PS60, NCOVR.sf$PS70, NCOVR.sf$PS80, NCOVR.sf$PS90)
pUE <- c(NCOVR.sf$UE60, NCOVR.sf$UE70, NCOVR.sf$UE80, NCOVR.sf$UE90)
pDV <- c(NCOVR.sf$DV60, NCOVR.sf$DV70, NCOVR.sf$DV80, NCOVR.sf$DV90)
pFP <- c(NCOVR.sf$FP59, NCOVR.sf$FP70, NCOVR.sf$FP80, NCOVR.sf$FP90)
pSOUTH <- rep(NCOVR.sf$SOUTH, Tm)
pNCOVR <- data.frame(indiv = index_indiv, time = index_time,
                     HR = pHR, PS = pPS, UE = pUE, DV = pDV,
                     FP = pFP, SOUTH = pSOUTH)
pform <- HR | DV | FP ~ PS + UE | PS + UE + SOUTH | PS
### SIM
### Remark: It is necessary to provide Tm value as argument
### when G>1 && Tm>1
pNCOVRSUR.sim <- spsurml(formula = pform, data = pNCOVR,
                         type = "sim", Tm = Tm)
print(pNCOVRSUR.sim)
if (require(gridExtra)) {
  pl <- plot(pNCOVRSUR.sim, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$lplbetas[[3]], nrow = 3)
}
# SLM
pNCOVRSUR.slm <- spsurml(formula = pform, data = pNCOVR,
                         listw = lwncovr, type = "slm", method = "Matrix", Tm = Tm,
                         zero.policy = TRUE, control= list(fdHess = TRUE))
print(pNCOVRSUR.slm)
if (require(gridExtra)) {
  pl <- plot(pNCOVRSUR.slm, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$lplbetas[[3]], pl$pldeltas, nrow = 4)
}
```

```
pNCOVRSUR.sem <- spsurml(formula = pform, data = pNCOVR,
                         listw = lwncovr, type = "sem", method = "Matrix", Tm = Tm,
                         zero.policy = TRUE, control= list(fdHess = TRUE))
print(pNCOVRSUR.sem)
if (require(gridExtra)) {
  pl <- plot(pNCOVRSUR.sem, viewplot = FALSE)
  grid.arrange(pl$lplbetas[[1]], pl$lplbetas[[2]],
               pl$lplbetas[[3]], pl$pldeltas, nrow = 4)
}
```

---

spsurtime *Estimation of SUR models for simple spatial panels (G=1).*

Description

This function estimates SUR models for simple spatial panel datasets. `spsurtime` is restricted, specifically, to cases where there is only one equation, *G=1*, and a varying number of spatial units, *N*, and time periods, *Tm*. The SUR structure appears in form of serial dependence among the error terms corresponding to the same spatial unit. Note that it is assumed that all spatial units share a common pattern of serial dependence.

The user can choose between different types of spatial specifications, as described below, and the estimation algorithms allow for the introduction of linear restrictions on the β parameters associated to the regressors. The spatial panels with SUR structure can be estimated by maximum-likelihood methods or three-stages least squares procedures, using spatial instrumental variables.

Usage

```
spsurtime (formula, data, time, na.action,
                    listw = NULL, type = "sim", Durbin = NULL,
                    method = "eigen", fit_method = "ml", maxlagW = NULL,
                    zero.policy = NULL, interval = NULL, trs = NULL,
                    R = NULL, b = NULL, demean = FALSE, control = list() )
```

Arguments

`formula` An object type `Formula` similar to objects created with the package **Formula** describing the equations to be estimated in the model. This model may contain several responses (explained variables) and a varying number of regressors in each equation.

`data` An object of class data.frame or a matrix.

`time` Time variable.

`na.action` A function (default `options("na.action")`), can also be `na.omit` or `na.exclude` with consequences for residuals and fitted values. It may be necessary to set `zero.policy` to `TRUE` because this subsetting may create no-neighbour observations.

`listw`	A `listw` object created for example by `nb2listw` from **spatialreg** package; if `nb2listw` not given, set to the same spatial weights as the `listw` argument. It can also be a spatial weighting matrix of order *(NxN)* instead of a `listw` object. Default = `NULL`.
`type`	Type of spatial model specification: "sim", "slx", "slm", "sem", "sdm", "sdem", "sarar" or "gnm". Default = "sim".
`Durbin`	If a formula object and model is type "sdm", "sdem" or "slx" the subset of explanatory variables to lag for each equation.
`method`	Similar to the corresponding parameter of `lagsarlm` function in **spatialreg** package. "eigen" (default) - the Jacobian is computed as the product of (1 - rho*eigenvalue) using `eigenw`, and "spam" or "Matrix_J" for strictly symmetric weights lists of styles "B" and "C", or made symmetric by similarity (Ord, 1975, Appendix C) if possible for styles "W" and "S", using code from the spam or Matrix packages to calculate the determinant; "Matrix" and "spam_update" provide updating Cholesky decomposition methods; "LU" provides an alternative sparse matrix decomposition approach. In addition, there are "Chebyshev" and Monte Carlo "MC" approximate log-determinant methods; the Smirnov/Anselin (2009) trace approximation is available as "moments". Three methods: "SE_classic", "SE_whichMin", and "SE_interp" are provided experimentally, the first to attempt to emulate the behaviour of Spatial Econometrics toolbox ML fitting functions. All use grids of log determinant values, and the latter two attempt to ameliorate some features of "SE_classic".
`fit_method`	Method of estimation for the spatial panel SUR model, either *ml* or *3sls*. Default = *ml*.
`maxlagW`	Maximum spatial lag order of the regressors employed to produce spatial instruments for the spatial lags of the explained variables. Default = 2. Note that in case of `type="sdm"`, the default value for `maxlagW` is set to 3 because the first lag of the regressors, WX_{tg}, can not be used as spatial instruments.
`zero.policy`	Similar to the corresponding parameter of `lagsarlm` function in **spatialreg** package. If `TRUE` assign zero to the lagged value of zones without neighbours, if `FALSE` assign `NA` - causing `spsurml()` to terminate with an error. Default = `NULL`.
`interval`	Search interval for autoregressive parameter. Default = `NULL`.
`trs`	Either `NULL` or vector of powered spatial weights matrix traces output by `trW`. Default = `NULL`.
`R`	A row vector of order *(1xpr)* with the set of *r* linear constraints on the *beta* parameters. The *first* restriction appears in the first *p* terms, the second restriction in the next *p* terms and so on. Default = `NULL`.
`b`	A column vector of order *(rx1)* with the values of the linear restrictions on the *beta* parameters. Default = `NULL`.
`demean`	Logical value to allow for the demeaning of panel data. In this case, `spsurml` substracts the individual mean to each spatial or cross-sectional unit. Default = `FALSE`.
`control`	List of additional control arguments.

Details

Function spsurtime only admits a formula, created with Formula and a dataset of class `data.frame` or `matrix`. That is, the data cannot be uploaded using data matrices Y and X provided for other functions in this package.
The argument `time` selects the variable, in the `data.frame`, associated to the time dimension in the panel dataset. Then spsurtime operates as in Anselin (1988), that is, each cross-section is treated as if it were an equation in a SUR model, which now has *Tm* 'equations' and *N* individuals.
The SUR structure appears because there is serial dependence in the errors of each individual in the panel. The serial dependence in the errors is not parameterized, but estimated non-parametrically in the $Sigma$ covariance matrix returned by the function. An important constraint to mention is that the serial dependence assumed to be the same for all individuals in the sample. Serial dependence among individuals is excluded from Anselin approach.

Value

An `spsur` object with the output of the maximum-likelihood or three-stages least-squares estimation of the spatial panel SUR model.

Author(s)

Fernando López	`<fernando.lopez@upct.es>`
Román Mínguez	`<roman.minguez@uclm.es>`
Jesús Mur	`<jmur@unizar.es>`

References

- Anselin, L. (1988). Spatial econometrics: methods and models. Dordrecht, Kluwer Academic Publishers.
- López, F.A., Mur, J., and Angulo, A. (2014). Spatial model selection strategies in a SUR framework. The case of regional productivity in EU. *Annals of Regional Science*, 53(1), 197-220.
- López, F.A., Martínez-Ortiz, P.J., & Cegarra-Navarro, J.G. (2017). Spatial spillovers in public expenditure on a municipal level in Spain. *Annals of Regional Science*, 58(1), 39-65.
- Mur, J., López, F., and Herrera, M. (2010). Testing for spatial effects in seemingly unrelated regressions. *Spatial Economic Analysis*, 5(4), 399-440.

See Also

spsurml, spsur3sls, wald_betas, wald_deltas, lmtestspsur, lrtest

Examples

```
####################################
```

```
######## PANEL DATA (G=1; Tm>1) ###
####################################

## Example 1:
rm(list = ls()) # Clean memory
data(spc)
lwspc <- spdep::mat2listw(Wspc, style = "W")
N <- nrow(spc)
Tm <- 2
index_time <- rep(1:Tm, each = N)
index_indiv <- rep(1:N, Tm)
WAGE <- c(spc$WAGE83, spc$WAGE81)
UN <- c(spc$UN83, spc$UN80)
NMR <- c(spc$NMR83, spc$NMR80)
SMSA <- c(spc$SMSA, spc$SMSA)
pspc <- data.frame(index_indiv, index_time, WAGE, UN,
                    NMR, SMSA)
form_pspc <- WAGE ~ UN + NMR + SMSA

# SLM
pspc_slm <- spsurtime(formula = form_pspc, data = pspc,
                      listw =lwspc,
                      time = pspc$index_time,
                      type = "slm", fit_method = "ml")
summary(pspc_slm)

## VIP: The output of the whole set of the examples can be examined
## by executing demo(demo_spsurtime, package="spsur")


### Example 2:
rm(list = ls()) # Clean memory
### Read NCOVR.sf object
data(NCOVR, package="spsur")
nbncovr <- spdep::poly2nb(NCOVR.sf, queen = TRUE)
### Some regions with no links...
lwncovr <- spdep::nb2listw(nbncovr, style = "W", zero.policy = TRUE)
N <- nrow(NCOVR.sf)
Tm <- 4
index_time <- rep(1:Tm, each = N)
index_indiv <- rep(1:N, Tm)
pHR <- c(NCOVR.sf$HR60, NCOVR.sf$HR70, NCOVR.sf$HR80, NCOVR.sf$HR90)
pPS <- c(NCOVR.sf$PS60, NCOVR.sf$PS70, NCOVR.sf$PS80, NCOVR.sf$PS90)
pUE <- c(NCOVR.sf$UE60, NCOVR.sf$UE70, NCOVR.sf$UE80, NCOVR.sf$UE90)
pNCOVR <- data.frame(indiv = index_indiv, time = index_time,
                     HR = pHR, PS = pPS, UE = pUE)
form_pHR <- HR ~ PS + UE

## SIM

pHR_sim <- spsurtime(formula = form_pHR, data = pNCOVR,
                    time = pNCOVR$time, type = "sim", fit_method = "ml")
summary(pHR_sim)
```

```
## SLM by 3SLS.

pHR_slm <- spsurtime(formula = form_pHR, data = pNCOVR, listw = lwncovr,
                     time = pNCOVR$time, type = "slm",
                     fit_method = "3sls")
summary(pHR_slm)

############################# Wald tests about betas in spatio-temporal models
### H0: equal betas for PS in equations 1, 3 and 4.
R <- matrix(0, nrow = 2, ncol = 12)
## nrow = number of restrictions
## ncol = number of beta parameters
R[1, 2] <- 1; R[1, 8] <- -1 # PS beta coefficient in equations 1 equal to 3
R[2, 2] <- 1; R[2, 11] <- -1 # PS beta coefficient in equations 1 equal to 4
b <- matrix(0, nrow=2, ncol=1)
wald_betas(pHR_sim , R = R , b = b) # SIM model
wald_betas(pHR_slm , R = R , b = b) # SLM model
############################# Wald tests about spatial-parameters in
############################# spatio-temporal models
### H0: equal rhos in slm model for equations 1 and 2.
R2 <- matrix(0, nrow = 1, ncol = 4)
R2[1, 1] <- 1; R2[1, 2] <- -1
b2 <- matrix(0, nrow = 1, ncol = 1)
wald_deltas(pHR_slm, R = R2, b = b2)
```

`summary.spsur` *Summary of estimated objects of class* spsur.

Description

This function summarizes estimated *spsur* objects. The tables in the output include basic information for each equation. The report also shows other complementary results corresponding to the SUR model like the *(GxG)* covariance matrix of the residuals of the equations of the SUR, the estimated log-likelihood, the Breusch-Pagan diagonality test or the Marginal Lagrange Multiplier, LMM, tests of spatial dependence.

Usage

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

Arguments

`object`	An *spsur* object estimated using `spsurml`, `spsur3sls` or `spsurtime` functions.
`...`	further arguments passed to or from other methods.

Value

An object of class *summary.spsur*

Author(s)

Fernando López	<fernando.lopez@upct.es>
Román Mínguez	<roman.minguez@uclm.es>
Jesús Mur	<jmur@unizar.es>

See Also

`print.summary.spsur`; `spsurml`; `spsur3sls`.

Examples

```
# See examples for \code{\link{spsurml}} or
# \code{\link{spsur3sls}} functions.
```

---

wald_betas — *Wald tests on the* beta *coefficients of the equation of the SUR model*

Description

The function `wald_betas` can be seen as a complement to the restricted estimation procedures included in the functions `spsurml` and `spsur3sls`. `wald_betas` obtains Wald tests for sets of linear restrictions on the coefficients β of the SUR model. The restrictions may involve coefficients of the same equation or coefficients from different equations. The function has great flexibility in this respect. Note that `wald_betas` is more general than `lr_betas` in the sense that the last function only allows to test for restrictions of homogeneity of subsets of β coefficients among the different equations in the SUR model, and in a maximum-likelihood framework.

In order to work with `wald_betas`, the model on which the linear restrictions are to be tested needs to exists as an *spsur* object. Using the information contained in the object, `wald_betas` obtains the corresponding Wald estatistic for the null hypotheses specified by the user through the *R* row vector and *b* column vector, used also in `spsurml` and `spsur3sls`. The function shows the value of the Wald test statistics and its associated p-values.

Usage

```
wald_betas (obj , R , b)
```

Arguments

obj	An spsur object created by spsurml, spsur3sls or spsurtime.
R	A row vector of order $(1xPr)$ showing the set of *r* linear constraints on the β parameters. The *first* restriction appears in the first *K* terms in *R*, the *second* restriction in the next *K* terms and so on.
b	A column vector of order *(rx1)* with the values of the linear restrictions on the β parameters.

Value

Object of htest class including the Wald statistic, the corresponding p-value, the degrees of freedom and the values of the sample estimates.

Author(s)

Fernando López	<fernando.lopez@upct.es>
Román Mínguez	<roman.minguez@uclm.es>
Jesús Mur	<jmur@unizar.es>

References

- López, F.A., Mur, J., and Angulo, A. (2014). Spatial model selection strategies in a SUR framework. The case of regional productivity in EU. *Annals of Regional Science*, 53(1), 197-220.
- López, F.A., Martínez-Ortiz, P.J., & Cegarra-Navarro, J.G. (2017). Spatial spillovers in public expenditure on a municipal level in Spain. *Annals of Regional Science*, 58(1), 39-65.
- Mur, J., López, F., and Herrera, M. (2010). Testing for spatial effects in seemingly unrelated regressions. *Spatial Economic Analysis*, 5(4), 399-440.

See Also

spsurml, spsur3sls, lr_betas

Examples

```
## VIP: The output of the whole set of the examples can be examined
## by executing demo(demo_wald_betas, package="spsur")

##################################################
######## CROSS SECTION DATA (G=1; Tm>1) ########
##################################################

##### Example 1: Spatial Phillips-Curve. Anselin (1988, p. 203)
rm(list = ls()) # Clean memory
data(spc)
```

```
lwspc <- spdep::mat2listw(Wspc, style = "W")
Tformula <- WAGE83 | WAGE81 ~ UN83 + NMR83 + SMSA | UN80 + NMR80 + SMSA
### Estimate SUR-SLM model
spcsur.slm <- spsurml(formula = Tformula, data = spc,
                      type = "slm", listw = lwspc)
summary(spcsur.slm)
### H_0: equality between SMSA coefficients in both equations.
R1 <- matrix(c(0,0,0,1,0,0,0,-1), nrow=1)
b1 <- matrix(0, ncol=1)

wald_betas(spcsur.slm, R = R1, b = b1)


### Estimate restricted SUR-SLM model
spcsur.slmr <- spsurml(formula = Tformula, data = spc,
                      type = "slm", listw = lwspc,
                      R = R1, b = b1)
summary(spcsur.slmr)

### H_0: equality between intercepts and SMSA coefficients in both equations.
R2 <- matrix(c(1,0,0,0,-1,0,0,0,0,0,0,1,0,0,0,-1),
             nrow = 2, ncol = 8, byrow = TRUE)
b2 <- matrix(c(0,0),ncol=1)
wald_betas(spcsur.slm, R = R2, b = b2)
### Estimate restricted SUR-SLM model
spcsur.slmr2 <- spsurml(formula = Tformula, data = spc,
                      type = "slm", listw = lwspc,
                      R = R2, b = b2)

#####################################
#########  G=1; Tm>1         ########
#####################################

##### Example 2: Homicides + Socio-Economics (1960-90)
#
rm(list = ls()) # Clean memory
### Read NCOVR.sf object
data(NCOVR, package = "spsur")
nbncovr <- spdep::poly2nb(NCOVR.sf, queen = TRUE)
### Some regions with no links...
lwncovr <- spdep::nb2listw(nbncovr, style = "W", zero.policy = TRUE)
Tformula <- HR80  | HR90 ~ PS80 + UE80 | PS90 + UE90
##################################
### A SUR-SLM model
NCOVRSUR.slm <-spsurml(formula = Tformula, data = NCOVR.sf,
                       type = "slm", listw = lwncovr,
                       method = "Matrix", zero.policy = TRUE,
                       control = list(fdHess = TRUE))
summary(NCOVRSUR.slm)
R1 <- matrix(c(0,1,0,0,-1,0), nrow=1)
b1 <- matrix(0, ncol=1)
wald_betas(NCOVRSUR.slm, R = R1, b = b1)
NCOVRSUR.slmr <-spsurml(formula = Tformula, data = NCOVR.sf,
```

```
                        type = "slm", listw = lwncovr,
                        method = "Matrix", zero.policy = TRUE,
                        control = list(fdHess = TRUE),
                        R = R1, b = b1)
summary(NCOVRSUR.slmr)
```

`wald_deltas` *Wald tests for spatial parameters coefficients.*

Description

Function wald_deltas obtains Wald tests for linear restrictions on the spatial coefficients of a SUR model that has been estimated previously through the function spsurml. The restrictions can affect to coefficients of the same equation (i.e., $\lambda_g = \rho_g forallg$) or can involve coefficients from different equations (i.e., $\lambda_g = \lambda_h$). The function has great flexibility in this respect. Note that wald_deltas only works in a maximum-likelihood framework.

In order to work with wald_betas, the model on which the linear restrictions are to be tested needs to exists as an *spsur* object. Using the information contained in the object, wald_deltas obtains the corresponding Wald estatistic for the null hypotheses specified by the user through the *R* row vector and *b* column vector discussed, used also in spsurml. The function shows the resulting Wald test statistics and their corresponding p-values.

Usage

```
wald_deltas (obj , R , b)
```

Arguments

obj	An `spsur` object created by spsurml, spsur3sls or spsurtime.
R	A row vector of order *(1xGr)* or *(1x2Gr)* showing the set of *r* linear constraints on the spatial parameters. The last case is reserved to "sarar" models where there appear *G* parameters λ_g and *G* parameters ρ_g, *2G* spatial in total. The *first* restriction appears in the first *G* terms in *R* (*2G* for the "sarar" case), the second restriction in the next *G* terms (*2G* for the "sarar" case) and so on.
b	A column vector of order *(rx1)* with the values of the linear restrictions on the β parameters.

Value

Object of `htest` including the Wald statistic, the corresponding p-value, the degrees of freedom and the values of the sample estimates.

Author(s)

Fernando López	<fernando.lopez@upct.es>
Román Mínguez	<roman.minguez@uclm.es>
Jesús Mur	<jmur@unizar.es>

See Also

spsurml, spsur3sls

Examples

```
##################################################
######## CROSS SECTION DATA (G>1; Tm=1) ########
##################################################
rm(list = ls()) # Clean memory
data(spc, package = "spsur")
lwspc <- spdep::mat2listw(Wspc, style = "W")
Tformula <- WAGE83 | WAGE81 ~ UN83 + NMR83 + SMSA | UN80 + NMR80 + SMSA

#################################
## Estimate SUR-SLM model
spcsur.slm <-spsurml(formula = Tformula, data = spc,
                       type = "slm", listw = lwspc)
summary(spcsur.slm)
## H_0: equality of the lambda parameters of both equations.
R1 <- matrix(c(1,-1), nrow=1)
b1 <- matrix(0, ncol=1)
wald_deltas(spcsur.slm, R = R1, b = b1)


## VIP: The output of the whole set of the examples can be examined
## by executing demo(demo_wald_deltas, package="spsur")


#################################
### Estimate SUR-SEM model
spcsur.sem <-spsurml(form = Tformula, data = spc,
                     type = "sem", listw = lwspc)
summary(spcsur.sem)
### H_0: equality of the rho parameters of both equations.
R2 <- matrix(c(1,-1), nrow=1)
b2 <- matrix(0, ncol=1)
wald_deltas(spcsur.sem, R = R2, b = b2)

##################################
### Estimate SUR-SARAR model
### It usually requires 2-3 minutes maximum
spcsur.sarar <-spsurml(formula = Tformula, data = spc,
                       type = "sarar", listw = lwspc,
                       control = list(tol=0.1))
summary(spcsur.sarar)
### H_0: equality of the lambda and rho parameters of both equations.
R3 <- matrix(c(1,-1,0,0,0,0,1,-1), nrow=2, ncol=4, byrow=TRUE)
b3 <- matrix(c(0,0), ncol=1)
wald_deltas(spcsur.sarar, R = R3, b = b3)
```

```
#####################################
#########  G=1; Tm>1         ########
#####################################

##' ##### Example 2: Homicides + Socio-Economics (1960-90)
rm(list = ls()) # Clean memory
### Read NCOVR.sf object
data(NCOVR, package = "spsur")
nbncovr <- spdep::poly2nb(NCOVR.sf, queen = TRUE)
### Some regions with no links...
lwncovr <- spdep::nb2listw(nbncovr, style = "W", zero.policy = TRUE)
Tformula <- HR80  | HR90 ~ PS80 + UE80 | PS90 + UE90

##################################
### A SUR-SLM model
NCOVRSUR.slm <-spsurml(formula = Tformula, data = NCOVR.sf,
                       type = "slm", listw = lwncovr,
                       method = "Matrix", zero.policy = TRUE,
                       control = list(fdHess = TRUE))
summary(NCOVRSUR.slm)
### H_0: equality of the lambda parameters of both equations.
R1 <- matrix(c(1,-1), nrow=1)
b1 <- matrix(0, ncol=1)
wald_deltas( NCOVRSUR.slm, R = R1, b = b1)

##################################
### Estimate SUR-SEM model
NCOVRSUR.sem <-spsurml(formula = Tformula, data = NCOVR.sf,
                       type = "sem", listw = lwncovr,
                       method = "Matrix", zero.policy = TRUE,
                       control = list(fdHess = TRUE))
summary(NCOVRSUR.sem)
### H_0: equality of the rho parameters of both equations.
R2 <- matrix(c(1,-1), nrow=1)
b2 <- matrix(0, ncol=1)
wald_deltas(NCOVRSUR.sem, R = R2, b = b2)
```

`Wspc` *Spatial weight matrix for South-West Ohio Counties to estimate Spatial Phillips-Curve*

Description

A spatial weight matrix row-standardized based on first order contiguity criterium.

Usage

```
Wspc
```

Format

A row-standardized squared matrix with 25 rows and columns. The rows and columns follow the same order than provinces included in *spc* data frame.

Source

Anselin (1988, p. 207)

References

- Anselin, L. (1988). *Spatial Econometrics: Methods and Models*. Springer Science & Business Media.

Index