Package ‘spldv’

October 11, 2023

Type Package
Title Spatial Models for Limited Dependent Variables
Version 0.1.3
Date 2023-10-11
Description The current version of this package estimates spatial autoregressive models for binary dependent variables using GMM estimators \(<\text{doi}:10.18637/jss.v107.i08>\). It supports one-step (Pinkse and Slade, 1998) \(<\text{doi}:10.1016/S0304-4076(97)00097-5>\) and two-step GMM estimator along with the linearized GMM estimator proposed by Klier and McMillen (2008) \(<\text{doi}:10.1198/073500107000000188>\). It also allows for either Probit or Logit model and compute the average marginal effects. All these models are presented in Sarrias and Piras (2023) \(<\text{doi}:10.1016/j.jocm.2023.100432>\).

Encoding UTF-8
RoxygenNote 7.2.3
Depends R (>= 4.0)
Imports Formula, Matrix, maxLik, stats, sphet, memisc, car, methods, numDeriv, MASS, spatialreg
Suggests spdep
License GPL (>= 2)
URL https://github.com/gpiras/spldv
BugReports https://github.com/gpiras/spldv/issues
NeedsCompilation no
Author Mauricio Sarrias [aut, cre] (<https://orcid.org/0000-0001-5932-4817>), Gianfranco Piras [aut] (<https://orcid.org/0000-0003-0225-6061>), Daniel McMillen [ctb]
Maintainer Mauricio Sarrias <msarrias86@gmail.com>
Repository CRAN
Date/Publication 2023-10-11 13:30:02 UTC
getSummary.bingmm

R topics documented:

getSummary.bingmm .................................................. 2
getSummary.binlgmm .................................................. 3
impacts ................................................................. 3
impacts.bingmm ....................................................... 4
sbinaryGMM .......................................................... 7
sbinaryLGMM ......................................................... 12

Description

A generic function to collect coefficients and summary statistics from a bingmm object. It is used in mtable.

Usage

## S3 method for class 'bingmm'
getSummary(obj, alpha = 0.05, ...)

Arguments

obj a bingmm object,
alpha level of the confidence intervals,
... further arguments,

Details

For more details see package memisc.

Value

A list with an array with coefficient estimates and a vector containing the model summary statistics.
**getSummary.binlgmm**

Get Model Summaries for use with "mtable" for objects of class binlgmm

**Description**

A generic function to collect coefficients and summary statistics from a binlgmm object. It is used in mtable.

**Usage**

```r
## S3 method for class 'binlgmm'
getSummary(obj, alpha = 0.05, ...)
```

**Arguments**

- `obj` : a binlgmm object,
- `alpha` : level of the confidence intervals,
- `...` : further arguments,

**Details**

For more details see package memisc.

**Value**

A list with an array with coefficient estimates and a vector containing the model summary statistics.

**impacts**

Estimation of the average marginal effects for SARB models.

**Description**

Obtain the average marginal effects from bingmm or binlgmm class model.

**Usage**

```r
impacts(object, ...)
```

**Arguments**

- `object` : an object of class bingmm or binlgmm
- `...` : Additional arguments to be passed.

**Value**

Estimates of the direct, indirect and total effect.
impacts.bingmm

Estimation of the average marginal effects for SARB model estimated using GMM procedures.

Description

Obtain the average marginal effects from bingmm or binlgmm class model.

Usage

```r
## S3 method for class 'bingmm'
impacts(
  object,
  vcov = NULL,
  vce = c("robust", "efficient", "ml"),
  het = TRUE,
  atmeans = FALSE,
  type = c("mc", "delta"),
  R = 100,
  approximation = FALSE,
  pw = 5,
  tol = 1e-06,
  empirical = FALSE,
  ...
)

## S3 method for class 'binlgmm'
impacts(
  object,
  vcov = NULL,
  het = TRUE,
  atmeans = FALSE,
  type = c("mc", "delta"),
  R = 100,
  approximation = FALSE,
  pw = 5,
  tol = 1e-06,
  empirical = FALSE,
  ...
)

## S3 method for class 'impacts.bingmm'
print(x, ...)

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

### Arguments

- `object`: an object of class `bingmm`, `binlgmm`, or `impacts.bingmm` for `summary` and `print` method.
- `vcov`: an estimate of the asymptotic variance-covariance matrix of the parameters for a `bingmm` or `binlgmm` object.
- `vce`: string indicating what kind of variance-covariance matrix of the estimate should be computed when using `effect.bingmm`. For the one-step GMM estimator, the options are "robust" and "ml". For the two-step GMM estimator, the options are "robust", "efficient" and "ml". The option "vce = ml" is an exploratory method that evaluates the VC of the RIS estimator using the GMM estimates.
- `het`: logical. If TRUE (the default), then the heteroskedasticity is taken into account when computing the average marginal effects.
- `atmeans`: logical. If FALSE (the default), then the average marginal effects are computed at the unit level.
- `type`: string indicating which method is used to compute the standard errors of the average marginal effects. If "mc", then the Monte Carlo approximation is used. If "delta", then the Delta Method is used.
- `R`: numerical. Indicates the number of draws used in the Monte Carlo approximation if `type = "mc"`.
- `approximation`: logical. If TRUE then \((I - \lambda W)^{-1}\) is approximated as \(I + \lambda W + \lambda^2 W^2 + \lambda^3 W^3 + \ldots + \lambda^q W^q\). The default is FALSE.
- `pw`: numeric. The power used for the approximation \(I + \lambda W + \lambda^2 W^2 + \lambda^3 W^3 + \ldots + \lambda^q W^q\). The default is 5.
- `tol`: Argument passed to `mvrnorm`: tolerance (relative to largest variance) for numerical lack of positive-definiteness in the coefficient covariance matrix.
- `empirical`: logical. Argument passed to `mvrnorm` (default FALSE): if TRUE, the coefficients and their covariance matrix specify the empirical not population mean and covariance matrix.
- `...`: further arguments. Ignored.
- `x`: an object of class `impacts.bingmm`.
- `digits`: the number of digits.

### Details

Let the model be:

\[
y^* = X\beta + WX\gamma + \lambda Wy^* + \epsilon = Z\delta + \lambda Wy^* + \epsilon
\]

where \(y = 1\) if \(y^* > 0\) and 0 otherwise; \(\epsilon \sim N(0, 1)\) if `link = "probit"` or \(\epsilon \sim L(0, \pi^2/3)\) if `link = "logit"`.

The marginal effects respect to variable \(x_r\) can be computed as
\[ \text{diag}(f(a))D^{-1}_\lambda A^{-1}_\lambda (I_n \beta_r + W \gamma_r) = C_r(\theta) \]

where \( f() \) is the pdf, which depends on the assumption of the error terms; \( \text{diag} \) is the operator that creates a \( n \times n \) diagonal matrix; \( A_\lambda = (I - \lambda W) \); and \( D_\lambda \) is a diagonal matrix whose elements represent the square root of the diagonal elements of the variance-covariance matrix of \( u = A_\lambda^{-1} \epsilon \).

We implement these three summary measures: (1) The average total effects, \( ATE_r = n^{-1} i_n' C_r i_n \), (2) The average direct effects, \( ADE_r = n^{-1} \text{tr}(C_r) \), and (3) the average indirect effects, \( ATE_r - ADE_r \).

The standard errors of the average total, direct and indirect effects can be estimated using either Monte Carlo (MC) approximation, which takes into account the sampling distribution of \( \theta \), or Delta Method.

Value

An object of class \textit{impacts.bingmm}.

Author(s)

Mauricio Sarrias and Gianfranco Piras.

See Also

\texttt{sbinaryGMM, sbinaryLGMM}.

Examples

```r
# Data set
data(oldcol, package = "spdep")

# Create dependent (dummy) variable
COL.OLD$CRIMED <- as.numeric(COL.OLD$CRIME > 35)

# Two-step (Probit) GMM estimator
ts <- sbinaryGMM(CRIMED ~ INC + HOVAL | HOVAL,
                link = "probit",
                listw = spdep::nb2listw(COL.nb, style = "W"),
                data = COL.OLD,
                type = "twostep")

# Marginal effects using Delta Method
summary(impacts(ts, type = "delta"))

# Marginal effects using MC with 100 draws
summary(impacts(ts, type = "mc", R = 100))

# Marginal effects using efficient VC matrix
summary(impacts(ts, type = "delta", vce = "efficient"))

# Marginal effects using efficient VC matrix and ignoring the heteroskedasticity
```
summary(impacts(ts, type = "delta", vce = "efficient", het = FALSE))

---

**sbinaryGMM**

**Estimation of SAR for binary dependent models using GMM**

**Description**

Estimation of SAR model for binary dependent variables (either Probit or Logit), using one- or two-step GMM estimator. The type of model supported has the following structure:

\[ y^* = X\beta + WX\gamma + \lambda Wy^* + \epsilon = Z\delta + \lambda Wy^* + \epsilon \]

where \( y = 1 \) if \( y^* > 0 \) and 0 otherwise; \( \epsilon \sim N(0,1) \) if link = "probit" or \( \epsilon \sim L(0,\pi^2/3) \) if link = "logit".

**Usage**

```r
sbinaryGMM(
  formula,
  data,
  listw = NULL,
  nins = 2,
  link = c("probit", "logit"),
  winitial = c("optimal", "identity"),
  s.matrix = c("robust", "iid"),
  type = c("onestep", "twostep"),
  gradient = TRUE,
  start = NULL,
  cons.opt = FALSE,
  approximation = FALSE,
  verbose = TRUE,
  print.init = FALSE,
  pw = 5,
  tol.solve = .Machine$double.eps,
  ... )
```

## S3 method for class 'bingmm'

```r
coeff(object, ...)
```

## S3 method for class 'bingmm'

```r
cov( object,
  vce = c("robust", "efficient", "ml"),
  method = "bhhh",
  R = 1000,
  ... )
```
tol.solve = .Machine$double.eps, 
... 
}

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

## S3 method for class 'bingmm'
summary(
  object,
  vce = c("robust", "efficient", "ml"),
  method = "bhhh",
  R = 1000,
  tol.solve = .Machine$double.eps,
  ...
)

## S3 method for class 'summary.bingmm'
print(x, digits = max(5,getOption("digits") - 3), ...)

Arguments

- **formula**: a symbolic description of the model of the form y ~ x | wx where y is the binary dependent variable, x are the independent variables. The variables after | are those variables that enter spatially lagged: WX. The variables in the second part of formula must also appear in the first part. This rules out situations in which one of the regressors can be specified only in lagged form.
- **data**: the data of class data.frame.
- **listw**: object. An object of class listw, matrix, or Matrix.
- **nins**: numerical. Order of instrumental-variable approximation; as default nins = 2, such that $H = (Z, WZ, W^2Z)$ are used as instruments.
- **link**: string. The assumption of the distribution of the error term; it can be either link = "probit" (the default) or link = "logit".
- **winitial**: string. A string indicating the initial moment-weighting matrix $\Psi$; it can be either winitial = "optimal" (the default) or winitial = "identity".
- **s.matrix**: string. Only valid of type = "twostep" is used. This is a string indicating the type of variance-covariance matrix $S$ to be used in the second-step procedure; it can be s.matrix = "robust" (the default) or s.matrix = "iid".
- **type**: string. A string indicating whether the one-step (type = "onestep"), or two-step GMM (type = "twostep") should be computed.
- **gradient**: logical. Only for testing procedures. Should the analytic gradient be used in the GMM optimization procedure? TRUE as default. If FALSE, then the numerical gradient is used.
- **start**: if not NULL, the user must provide a vector of initial parameters for the optimization procedure. When start = NULL, sbinaryGMM uses the traditional Probit or Logit estimates as initial values for the parameters, and the correlation between $y$ and $Wy$ as initial value for $\lambda$. 


cons.opt  logical. Should a constrained optimization procedure for $\lambda$ be used? FALSE as default.

approximation  logical. If TRUE then $\left(I - \lambda W\right)^{-1}$ is approximated as $I + \lambda W + \lambda^2 W^2 + \lambda^3 W^3 + \ldots + \lambda^q W^q$. The default is FALSE.

verbose  logical. If TRUE, the code reports messages and some values during optimization.

print.init  logical. If TRUE the initial parameters used in the optimization of the first step are printed.

pw  numeric. The power used for the approximation $I + \lambda W + \lambda^2 W^2 + \lambda^3 W^3 + \ldots + \lambda^q W^q$. The default is 5.

tol.solve  Tolerance for solve().

... additional arguments passed to maxLik.

vce  string. A string indicating what kind of standard errors should be computed when using summary. For the one-step GMM estimator, the options are "robust" and "ml". For the two-step GMM estimator, the options are "robust", "efficient" and "ml". The option "vce = ml" is an exploratory method that evaluates the VC of the RIS estimator using the GMM estimates.

method  string. Only valid if vce = "ml". It indicates the algorithm used to compute the Hessian matrix of the RIS estimator. The default is "bhhh".

R  numeric. Only valid if vce = "ml". It indicates the number of draws used to compute the simulated probability in the RIS estimator.

x, object,  an object of class binaryGMM

digits  the number of digits

Details

The data generating process is:

$$y^* = X\beta + WX\gamma + \lambda W y^* + \epsilon = Z\delta + \lambda W y^* + \epsilon$$

where $y = 1$ if $y^* > 0$ and 0 otherwise; $\epsilon \sim N(0, 1)$ if link = "probit" or $\epsilon \sim L(0, \pi^2/3)$ if link = "logit". The general GMM estimator minimizes

$$J(\theta) = g'(\theta)\hat{\Psi}g(\theta)$$

where $\theta = (\beta, \gamma, \lambda)$ and

$$g = n^{-1}H'v$$

where $v$ is the generalized residuals. Let $Z = (X, WX)$, then the instrument matrix $H$ contains the linearly independent columns of $H = (Z, WZ, \ldots, W^q Z)$. The one-step GMM estimator minimizes $J(\theta)$ setting either $\hat{\Psi} = I_p$ if winitial = "identity" or $\hat{\Psi} = (H' H / n)^{-1}$ if winitial = "optimal". The two-step GMM estimator uses an additional step to achieve higher efficiency by computing the variance-covariance matrix of the moments $\hat{S}$ to weight the sample moments. This matrix is computed using the residuals or generalized residuals from the first-step, which are consistent. This matrix is computed as $\hat{S} = n^{-1} \sum_{i=1}^n h_i (f^2/(F(1-F))) h_i'$ or $\hat{S} = n^{-1} \sum_{i=1}^n \hat{v}_i h_i h_i'$, where $\hat{v}$ are the first-step generalized residuals.
Value

An object of class "bingmm", a list with elements:

coefficients the estimated coefficients,
call the matched call,
callF the full matched call,
X the X matrix, which contains also WX if the second part of the formula is used,
H the H matrix of instruments used,
y the dependent variable,
listw the spatial weight matrix,
link the string indicating the distribution of the error term,
Psi the moment-weighting matrix used in the last round,
type type of model that was fitted,
s.matrix the type of S matrix used in the second round,
winitial the moment-weighting matrix used for the first step procedure
opt object of class maxLik,
approximation a logical value indicating whether approximation was used to compute the inverse matrix,
pw the powers for the approximation,
formula the formula.

Author(s)
Mauricio Sarrias and Gianfranco Piras.

References


#### Examples

```r
# Data set
data(oldcol, package = "spdep")

# Create dependent (dummy) variable
COL.OLD$CRIMED <- as.numeric(COL.OLD$CRIME > 35)

# Two-step (Probit) GMM estimator
ts <- sbinaryGMM(CRIMED ~ INC + HOVAL, 
                 link = "probit", 
                 listw = spdep::nb2listw(COL.nb, style = "W"), 
                 data = COL.OLD, 
                 type = "twostep", 
                 verbose = TRUE)

# Robust standard errors
summary(ts)

# Efficient standard errors
summary(ts, vce = "efficient")

# One-step (Probit) GMM estimator
os <- sbinaryGMM(CRIMED ~ INC + HOVAL, 
                 link = "probit", 
                 listw = spdep::nb2listw(COL.nb, style = "W"), 
                 data = COL.OLD, 
                 type = "onestep", 
                 verbose = TRUE)

summary(os)

# One-step (Logit) GMM estimator with identity matrix as initial weight matrix
os_l <- sbinaryGMM(CRIMED ~ INC + HOVAL, 
                   link = "logit", 
                   listw = spdep::nb2listw(COL.nb, style = "W"), 
                   data = COL.OLD, 
                   type = "onestep", 
                   winitial = "identity", 
                   verbose = TRUE)

summary(os_l)

# Two-step (Probit) GMM estimator with WX
ts_wx <- sbinaryGMM(CRIMED ~ INC + HOVAL| INC + HOVAL, 
                    link = "probit", 
                    listw = spdep::nb2listw(COL.nb, style = "W"), 
                    data = COL.OLD, 
                    type = "twostep", 
                    verbose = FALSE)

summary(ts_wx)
```
# Constrained two-step (Probit) GMM estimator

ts_c <- sbinaryGMM(CRIMED ~ INC + HOVAL,
                  link = "probit",
                  listw = spdep::nb2listw(COL.nb, style = "W"),
                  data = COL.OLD,
                  type = "twostep",
                  verbose = TRUE,
                  cons.opt = TRUE)

summary(ts_c)

## Description

Estimation of SAR model for binary dependent variables (either Probit or Logit), using Linearized
GMM estimator suggested by Klier and McMillen (2008). The model is:

\[ y^* = X\beta + WX\gamma + \lambda Wy^* + \epsilon = Z\delta + \lambda Wy^* + \epsilon \]

where \( y = 1 \) if \( y^* > 0 \) and 0 otherwise; \( \epsilon \) \( \sim N(0,1) \) if \( \text{link} = "probit" \) or \( \epsilon \) \( \sim L(0, \pi^2/3) \) \( \text{link} = "logit" \).

## Usage

sbinaryLGMM(
  formula,
  data,
  listw = NULL,
  nins = 2,
  link = c("logit", "probit"),
  ...
)

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

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

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

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

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

formula a symbolic description of the model of the form \( y \sim x \mid wx \) where \( y \) is the binary dependent variable, \( x \) are the independent variables. The variables after \( \mid \) are those variables that enter spatially lagged: \( WX \). The variables in the second part of formula must also appear in the first part.

data the data of class data.frame.
listw object. An object of class listw, matrix, or Matrix.
nins numerical. Order of instrumental-variable approximation; as default nins = 2, such that \( H = (Z, WZ, W^2Z) \) are used as instruments.
link string. The assumption of the distribution of the error term; it can be either link = "probit" (the default) or link = "logit".
... additional arguments.
x, object, an object of class binlgmm.
digits the number of digits

Details

The steps for the linearized spatial Probit/Logit model are the following:

1. Estimate the model by standard Probit/Logit model, in which spatial autocorrelation and heteroskedasticity are ignored. The estimated values are \( \beta_0 \). Calculate the generalized residuals assuming that \( \lambda = 0 \) and the gradient terms \( G_\beta \) and \( G_\lambda \).

2. The second step is a two-stage least squares estimator of the linearized model. Thus regress \( G_\beta \) and \( G_\lambda \) on \( H = (Z, WZ, W^2Z, \ldots, W^qZ) \) and obtain the predicted values \( \hat{G} \). Then regress \( u_0 + G_\beta' \hat{G}_0 \) on \( \hat{G} \). The coefficients are the estimated values of \( \beta \) and \( \lambda \).

The variance-covariance matrix can be computed using the traditional White-corrected coefficient covariance matrix from the last two-stage least squares estimator of the linearized model.

Value

An object of class "binlgmm", a list with elements:

coefficients the estimated coefficients,
call the matched call,
X the X matrix, which contains also WX if the second part of the formula is used,
H the H matrix of instruments used,
y the dependent variable,
listw the spatial weight matrix,
link the string indicating the distribution of the error term,
fit an object of lm representing the T2SLS,
formula the formula.

Author(s)

Mauricio Sarrias and Gianfranco Piras.
References


See Also

sbinaryGMM, impacts.bingmm.

Examples

# Data set
data(oldcol, package = "spdep")

# Create dependent (dummy) variable
COL.OLD$CRIMED <- as.numeric(COL.OLD$CRIME > 35)

# LGMM for probit using q = 3 for instruments
lgmm <- sbinaryLGMM(CRIMED ~ INC + HOVAL | INC,
                     listw = spdep::nb2listw(COL.nb, style = "W"),
                     nins = 3,
                     data = COL.OLD)

summary(lgmm)
Index

* models
  sbinaryGMM, 7
  sbinaryLGMM, 12

coef.bingmm (sbinaryGMM), 7
coef.binlgmm (sbinaryLGMM), 12

getSummary.bingmm, 2
getSummary.binlgmm, 3

impacts, 3
impacts.bingmm, 4, 11, 14
impacts.binlgmm (impacts.bingmm), 4

print.bingmm (sbinaryGMM), 7
print.binlgmm (sbinaryLGMM), 12
print.impacts.bingmm (impacts.bingmm), 4
print.summary.bingmm (sbinaryGMM), 7
print.summary.binlgmm (sbinaryLGMM), 12
print.summary.impacts.bingmm (impacts.bingmm), 4

sbinaryGMM, 6, 7, 14
sbinaryLGMM, 6, 11, 12
summary.bingmm (sbinaryGMM), 7
summary.binlgmm (sbinaryLGMM), 12
summary.impacts.bingmm (impacts.bingmm), 4

vcov.bingmm (sbinaryGMM), 7
vcov.binlgmm (sbinaryLGMM), 12