Package 'spdgp'

May 9, 2026

Title Simulate Spatial Data Generation Processes

Version 0.1.0

Description Provides functionality for simulating data generation processes across various spatial regression models, conceptually aligned with the 'dgp' module of the 'Python' library 'spreg' <https://pysal.org/spreg/api.html#dgp>.

License MIT + file LICENSE

Encoding UTF-8

Language en

RoxygenNote 7.3.3

URL https://josiahparry.github.io/spdgp/, https://github.com/josiahparry/spdgp

BugReports https://github.com/josiahparry/spdgp/issues

Imports cli, MASS, Matrix, methods, rlang, sf, smoothmest, spatialreg, spdep, stats, vctrs

NeedsCompilation no

Author Josiah Parry [aut] (ORCID: <https://orcid.org/0000-0001-9910-865X>), Wenbo Lv [aut, cre] (ORCID: <https://orcid.org/0009-0002-6003-3800>), Edzer Pebesma [ctb] (ORCID: <https://orcid.org/0000-0001-8049-7069>)

Maintainer Wenbo Lv <lyu.geosocial@gmail.com>

Repository CRAN

Date/Publication 2025-10-28 12:20:02 UTC

Contents

`make_error` *Simulate an error term*

Description

Simulate an error term

Usage

```
make_error(
  n = 10,
  mu = 0,
  var = 1,
  method = c("normal", "laplace", "cauchy", "lognormal")
)
```

Arguments

`n`	the number of values to simulate.
`mu`	the sample average.
`var`	the sample variance. The `sqrt(var)` is passed to `rnorm()` and `rlnorm()` for normal and laplace distributions. `sqrt(var / 2)` is used for `laplace()` .
`method`	must be one of `"normal"`, `"laplace"`, `"cauchy"`, or `"lognormal"`.

Details

- `"normal"`: fit with `rnorm()`
- `"laplace"`: fit with `smoothmest::rdoublex()`
- `"cauchy"`: fit with `rcauchy()`
- `"lognormal"`: fit with `rlnorm()`

Value

A numeric vector

make_square_grid *Create a square grid*

Description

Creates a square grid with `ncol` and `nrow` dimensions.

Usage

```
make_square_grid(nrow, ncol = nrow)
```

Arguments

`nrow`	the number of rows in the grid.
`ncol`	defaults to `nrow`. The number columns in the grid.

Value

An `sfc` object by `sf` package.

Examples

```
make_square_grid(3, 2)
```

make_wx *Create Spatial Lags of variables*

Description

Given a dataframe of numeric values and a spatial weights matrix, calculate the spatial lag of each variable.

Usage

```
make_wx(x, listw, order = NULL)
```

Arguments

`x`	a `data.frame` of independent variables generated with `make_x()`.
`listw`	a `listw` object generated with `sim_grid_listw()`.
`order`	unused.

Value

A `data.frame` of the spatially lagged variables.

Examples

```
listw <- sim_grid_listw(10, 10)
x_vars <- make_x(100, mu = c(0.5, 1.2), var = c(1, 0.5))
res <- make_wx(x_vars, listw)
head(res)
```

---

make_wxg *Calculate the effect of spatially lagged X variables*

Description

This function computes the contribution of spatially lagged X variables based on provided coefficients. The function takes the spatially lagged variables (wx, see make_wx()) and multiplies them by their corresponding regression coefficients (gamma), returning the predicted influence of the spatial lags. Only spatial lags are considered; the original X variables are not included in this calculation.

Usage

```
make_wxg(wx, gamma)
```

Arguments

wx	a matrix of spatially lagged x variables.
gamma	a vector of coefficients for the spatially lagged x variables. Its length must match the number of columns in wx.

Value

A numeric vector

Examples

```
grid <- make_square_grid(5)
listw <- spdep::nb2listw(spdep::poly2nb(grid))
x <- make_x(25, c(0,1), c(1,4))
wx <- make_wx(x, listw)
gamma <- c(1.75, 0.4)
make_wxg(wx, gamma)
```

make_xb *Calculate predicted X values based on coefficients*

Description

This function calculates predicted x values based on regression coefficients. The results of this function can be passed to other `sim_*()` functions.

Usage

```
make_xb(x, beta)
```

Arguments

`x`	a `data.frame` of independent variables generated with `make_x()`.
`beta`	a vector of the beta coefficients for each of the variables. There must be `ncol(x) + 1` values. The first element of the vector is the intercept.

Value

A numeric vector

Examples

```
x <- make_x(25, c(0,1), c(1,4))
betas <- c(1, 1.5, -2)
make_xb(x, betas)
```

make_x_bivariate *Simulate X variables*

Description

Simulates independent variables.

Usage

```
make_x_bivariate(n = 5, mu = 1, cor = 0.25, var = c(1, 1))

make_x_uniform(n = 5, var = 1)

make_x_normal(n = 5, mu = 0, var = 1)

make_x(
  n = 5,
  mu = 0,
```

```
  var = 1,
  cor = 0,
  method = c("uniform", "normal", "bivnormal")
)
```

Arguments

n	the number of values to simulate.
mu	the sample average.
cor	correlation between bivariate normal
var	the sample variance. The sqrt(var) is passed to rnorm() and rlnorm() for normal and laplace distributions. sqrt(var / 2) is used for laplace() .
method	must be one of "uniform" (default), "normal", or "bivnormal" (bivariate normal).

Value

A data.frame of the simulated independent variables.

Examples

```
make_x(10, mu = c(0.5, 1.2), var = c(1, 0.5))
```

---

sim_durbin *Simulate the Spatial Durbin Model*

Description

Simulate the Spatial Durbin Model

Usage

```
sim_durbin(u, xb, wxg, listw, rho = 0.5)
```

Arguments

u	an error vector
xb	predicted x values as calculated by make_xb()
wxg	predicted spatial lag effect as calculated by make_wxg()
listw	a listw object generated with sim_grid_listw().
rho	the spatial autoregressive coefficient for the spatially lagged dependent variable.

Value

A numeric vector

References

spreg.dgp.dgp_spdurbin

Examples

```
ncol <- 20
n <- ncol^2
listw <- sim_grid_listw(ncol, ncol)  # Create spatial weights for a grid
u <- make_error(n)  # Simulate random errors
x <- make_x(
  n,
  mu = c(0.25, 5),
  var = c(1, 0.75),
  method = "normal"
)  # Generate x variables

# create xb with intercept = 1, beta1 = 2, beta2 = -3
xb <- make_xb(x, c(1, 2, -3))
wx <- make_wx(x, listw)
wxg <- make_wxg(wx, c(-2, 1.5))
y <- sim_durbin(u, xb, wxg, listw, rho = 0.5)

# combine data
df <- cbind(y = y, x)

# fit SDM
spatialreg::lagsarlm(y ~ ., df, listw, Durbin = TRUE)
```

---

sim_error *Simulate Spatial Error Process*

Description

This function generates a pure spatial error process, which is useful when you only want to simulate the error structure without including any deterministic part (i.e., no xb term). This can be used to analyze or simulate the behavior of spatially dependent errors in isolation.

Usage

```
sim_error(u, listw, lambda = 0.5, model = c("sar", "ma"))
```

Arguments

u	an error vector
listw	a listw object generated with sim_grid_listw().
lambda	a value value between -1 and 1. The spatial autoregressive coefficient for the error term.
model	default "sar". Which model should be simulated. Provide "ma" for the moving average.

Value

A numeric vector

References

See spreg.dgp.dgp_errproc

Examples

```
listw <- sim_grid_listw(5)
u <- make_error(25)
sim_error(u, listw)
```

---

sim_gns *Simulate General Nested Model*

Description

Simulate General Nested Model

Usage

```
sim_gns(u, xb, wxg, listw, rho = 0.5, lambda = 0.2, model = c("sar", "ma"))
```

Arguments

u	an error vector
xb	predicted x values as calculated by make_xb()
wxg	predicted spatial lag effect as calculated by make_wxg()
listw	a listw object generated with sim_grid_listw().
rho	the spatial autoregressive coefficient for the spatially lagged dependent variable.
lambda	a value value between -1 and 1. The spatial autoregressive coefficient for the error term.
model	default "sar". Which model should be simulated. Provide "ma" for the moving average.

Value

A numeric vector

References

spreg.dgp.dgp_gns

sim_grid_listw *Generate spatial weights matrix for a grid*

Description

Create a spatial weights matrix based on a square grid structure.

Usage

```
sim_grid_listw(nrow, ncol = nrow, style = "W", type = c("queen", "rook"))
```

Arguments

nrow	the number of rows in the grid.
ncol	defaults to nrow. The number columns in the grid.
style	the spatial weights style. Defaults to row standardized. See spdep::nb2listw() for more.
type	default "queen". Can also be "rook".

Value

A listw object by spdep package.

Examples

```
sim_grid_listw(10, 5)
```

sim_mess *Simiulate Matrix Exponential Spatial Lag Model*

Description

Simiulate Matrix Exponential Spatial Lag Model

Usage

```
sim_mess(u, xb, listw, rho = 0.5)
```

Arguments

u	an error vector
xb	predicted x values as calculated by make_xb()
listw	a listw object generated with sim_grid_listw().
rho	the spatial autoregressive coefficient for the spatially lagged dependent variable.

Value

A numeric vector

References

dgp_mess

`sim_ols` *Simulate OLS*

Description

Simulate a y variable for an Ordinary Least Squares (OLS) regression.

Usage

```
sim_ols(u, xb)
```

Arguments

u	an error vector
xb	predicted x values as calculated by `make_xb()`

Value

A numeric vector

References

spreg.dgp.dgp_ols

Examples

```
u <- make_error(50, method = "normal")
x <- make_x(50)
xb <- make_xb(x, c(1,2))
y <- sim_ols(u, xb)
lm(y ~ x[[1]])
```

sim_sar *Simulate Spatial Lag Model (SAR)*

Description

Simulate y for a SAR model.

Usage

```
sim_sar(u, xb, listw, rho = 0.5)
```

Arguments

u	an error vector
xb	predicted x values as calculated by make_xb()
listw	a listw object generated with sim_grid_listw().
rho	the spatial autoregressive coefficient for the spatially lagged dependent variable.

Value

A numeric vector

References

spreg.dgp.dgp_lag

Examples

```
ncol <- 20
n <- ncol^2
listw <- sim_grid_listw(ncol, ncol)  # Create spatial weights for a grid
u <- make_error(n)  # Simulate random errors
x <- make_x(
  n,
  mu = c(0.25, 5),
  var = c(1, 0.75),
  method = "normal"
)  # Generate x variables

# create xb with intercept = 1, beta1 = 2, beta2 = -3
xb <- make_xb(x, c(1, 2, -3))
y <- sim_sar(u, xb, listw)

# combine data
df <- cbind(y = y, x)

# fit SAR model
# Note lambda, x_1, and x_2 estimates.
spatialreg::stsls(y ~ ., df, listw)
```

sim_sarar *Simulate the Spatial Autoregressive Model with Autoregressive Errors*

Description

Generate y values for the "combo" / SARAR / SAC model.

Usage

```
sim_sarar(u, xb, listw, rho = 0.5, lambda = 0.2, model = c("sar", "ma"))
```

Arguments

u	an error vector
xb	predicted x values as calculated by make_xb()
listw	a listw object generated with sim_grid_listw().
rho	the spatial autoregressive coefficient for the spatially lagged dependent variable.
lambda	a value value between -1 and 1. The spatial autoregressive coefficient for the error term.
model	default "sar". Which model should be simulated. Provide "ma" for the moving average.

Value

A numeric vector

References

spreg.dgp.dgp_lagerr

sim_sem *Simulate Spatial Error Model (SEM)*

Description

Simulate the y values for an SEM model.

Usage

```
sim_sem(u, xb, listw, lambda = 0.5, model = c("sar", "ma"))
```

Arguments

u	an error vector
xb	predicted x values as calculated by make_xb()
listw	a listw object generated with sim_grid_listw().
lambda	a value value between -1 and 1. The spatial autoregressive coefficient for the error term.
model	default "sar". Which model should be simulated. Provide "ma" for the moving average.

Value

A numeric vector

References

spreg.dgp.dgp_sperror

Examples

```
ncol <- 10
n <- ncol^2
listw <- sim_grid_listw(ncol, ncol)  # Create spatial weights for a grid
u <- make_error(n)  # Simulate random errors
x <- make_x(
  n,
  mu = c(0.25, 5),
  var = c(1, 0.75),
  method = "normal"
)  # Generate x variables

# create xb with intercept = 1, beta1 = 2, beta2 = -3
xb <- make_xb(x, c(1, 2, -3))
y <- sim_sem(u, xb, listw)

# combine data
df <- cbind(y = y, x)

# fit SEM model
# Note lambda, x_1, and x_2 estimates.
spatialreg::errorsarlm(y ~ ., df, listw)
```

---

sim_slx *Simulate Spatially Lagged X (SLX) model*

Description

This function simulates the y values of an SLX model, where the dependent variable is influenced by both the original and spatially lagged x variables.

Usage

```
sim_slx(u, xb, wxg)
```

Arguments

u	an error vector
xb	predicted x values as calculated by `make_xb()`
wxg	predicted spatial lag effect as calculated by `make_wxg()`

Value

A numeric vector

References

spreg.dgp.dgp_slx

Examples

```
ncol <- 20
n <- ncol^2
listw <- sim_grid_listw(ncol, ncol)  # Create spatial weights for a grid
u <- make_error(n, method = "normal")  # Simulate random errors
x <- make_x(n, method = "uniform")  # Generate x variables
xb <- make_xb(x, c(1, 2))  # Calculate xb using the original x and coefficients
wx <- make_wx(x, listw)  # Generate spatially lagged x variables
wxg <- make_wxg(wx, 0.5)  # Calculate the effect of the spatial lags
y <- sim_slx(u, xb, wxg)  # Simulate the SLX model outcome
df <- data.frame(y, x)
spatialreg::lmSLX(y ~ ., data = df, listw = listw)  # Estimate the SLX model
```

---

sim_slx_error *Simulate Spatially Lagged X Error Model*

Description

Simulate Spatially Lagged X Error Model

Usage

```
sim_slx_error(u, xb, wxg, listw, lambda = 0.5, model = c("sar", "ma"))
```

Arguments

u	an error vector
xb	predicted x values as calculated by make_xb()
wxg	predicted spatial lag effect as calculated by make_wxg()
listw	a listw object generated with sim_grid_listw().
lambda	a value value between -1 and 1. The spatial autoregressive coefficient for the error term.
model	default "sar". Which model should be simulated. Provide "ma" for the moving average.

Value

A numeric vector

References

spreg.dgp.dgp_slxerror

Index