Package 'EgoCor'

July 12, 2022

Type Package

Title Simple Presentation of Estimated Exponential Semi-Variograms

Version 1.0.0

Description User friendly interface based on the R package 'gstat' to fit exponential parametric models to empirical semi-variograms in order to model the spatial correlation structure of health data. Geo-located health outcomes of survey participants may be used to model spatial effects on health in an ego-centred approach. The package contains a range of functions to help explore the spatial structure of the data as well as visualize the fit of exponential models for various metaparameter combinations with respect to the number of lag intervals and maximal distance. Furthermore, the outcome of interest can be adjusted for covariates by fitting a linear regression in a preliminary step before the semi-variogram fitting process.

License MIT + file LICENSE

Depends R (>= 3.5.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Imports SpatialTools,
stats,
grDevices,
graphics,
shiny,
Rdpack,
gstat,
sp

RdMacros Rdpack

Suggests knitr,
rmarkdown,
lme4

VignetteBuilder knitr

URL https://github.com/julia-dyck/EgoCor

BugReports https://github.com/julia-dyck/EgoCor/issues

R topics documented:

`birth` *Spatially correlated birthweight data with artificial coordinates*

Description

A simulated dataset containing geo-coded birthweight data with covariates. It is provided for exemplary applications of the functions within the EgoCor package.

Usage

```
birth
```

Format

A data frame with 903 rows and 8 variables:

x x-coordinate given in Cartesian format in meters,

y y-coordinate given in Cartesian format in meters,

birthweight weight of the child in gram,

primiparous binary: 1 = first child birth, 0 = not the first child birth,

datediff difference between due date and birth,

bmi BMI (body mass index) of the mother at the beginning of pregnancy,

weight weight of the mother in kg,

inc income quintiles 0 (low), 1, 2, 3, 4 (high).

Details

The dataset is loosely based on the BaBi study dataset referred to in Spallek et al. (2017). The spatial correlation structure was already modeled using an exponential model in Sauzet et al. (2021).

References

Sauzet O, Breiding JH, Zolitschka KA, Breckenkamp J, Razum O (2021). "An ego-centred approach for the evaluation of spatial effects on health in urban areas based on parametric semivariogram models: concept and validation." *BMC medical research methodology*, **21**(1), 1–12.

Spallek J, Grosser A, Höller-Holtrichter C, Doyle I, Breckenkamp J, Razum O (2017). "Early childhood health in Bielefeld, Germany (BaBi study): study protocol of a social-epidemiological birth cohort." *BMJ Open*, **7**(8). ISSN 2044-6055, doi:10.1136/bmjopen2017018398, https://bmjopen.bmj.com/content/7/8/e018398.

coords.plot *Spatial Data Coordinate Plot*

Description

Plot of the Cartesian coordinates of study participant with color and shape coded indication whether the variable of interest is observed at a specific location.

Usage

```
coords.plot(data)
```

Arguments

data	A data frame or matrix containing the x-coordinates in meters in the first column, the y-coordinates in meters in the second column, and the values of the attribute of interest in the third column. Additional columns are ignored.

Value

The function returns a plot showing the points based on Cartesian coordinates. A black circle indicates that the variable of interest is observed at that location. A red cross flags a missing value.

Examples

```
## Example 1
xcoords = rnorm(10, mean = 0, sd = 20)
ycoords = rnorm(10, mean = 0, sd = 20)
value = c(22, 31, 10, NA, NA, 18, 9, NA, 1, 34)
dataset = cbind(xcoords, ycoords, value)
coords.plot(dataset)

## Example 2
coords.plot(birth)
```

distance.info *Pairwise Distances of Spatial Coordinates*

Description

A range of descriptive statistics on the distribution of the pairwise distances between observations.

Usage

```
distance.info(data)
```

Arguments

data	A data frame or matrix containing the x-coordinates in the first column and the y-coordinates in the second column. Additional columns are ignored.

Value

A list containing:

`distmatrix`	A matrix containing the pairwise Euclidean distances between all given locations.
`distset`	A vector containing the pairwise Euclidean distances. The elements equal the upper (or lower) triangle minus the diagonal of the distance matrix. If the input dataset has n rows, `distset` contains $n(n-1)/2$ distances.
`distsummary`	A summary containing the minimum, 1st quartile, median, mean, 3rd quartile and maximum of the `distset`.
`maxdist`	The maximal distance.

Moreover, the function prints a histogram of the pairwise distances saved in the list entry `distset`.

Examples

```
## Example 1
x = c(1,3,7,10,15)
y = c(5,19,8,3,11)
z = rnorm(5, mean = 20, sd = 40)
dataset = as.data.frame(cbind(x,y,z))
distance.info(data = dataset)

## Example 2
distance.info(birth)
```

`par.uncertainty` *Semi-variogram parameter uncertainty*

Description

Standard error estimates for the nugget effect c_0, partial sill σ_0^2 and shape parameter ϕ of a fitted exponential semi-variogram model.

Usage

```
par.uncertainty(
  vario.mod.output,
  mod.nr,
  par.est = NULL,
  data = NULL,
  max.dist = NULL,
  nbins = NULL,
  B = 1000,
  threshold.factor = 1.2
)
```

Arguments

`vario.mod.output`
: An output of the `vario.mod` function containing the information of the estimated exponential semi-variogram model of interest.

`mod.nr`
: The index number specifiying one of the exponential semi-variogram models listed in the `vario.mod.ouput`.

`par.est`
: A vector of length three containing the estimated parameters: the nugget effect, the partial sill and the shape parameter of the estimated exponential semi-variogram model. It is automatically extracted from the `vario.mod.output`, if provided.

`data`
: The data frame or matrix used to estimate the exponential semi-variogram of interest containing the x-coordinates in meters in the first column, the y-coordinates in meters in the second column and the data values in the third column. It is automatically extracted from the vario.mod.output, if provided.

`max.dist`
: The maximal distance used for the estimation of the exponential semi-variogram model of interest. It is automatically extracted from the vario.mod.output, if provided.

`nbins`
: The number of bins used for the estimation of the exponential semi-variogram model of interest. It is automatically extracted from the vario.mod.output, if provided.

`B`
: The number of bootstrap repetitions to generate a set of re-estimates of each parameter.

`threshold.factor`
: The threshold factor specifies the filter within the filtered bootstrap method (see details). If not specified, a default value of 1.2 is used.

Details

Two alternative approaches for the input of the arguments:

1. Provide the arguments vario.mod.output (output object from vario.mod function) and mod.nr (number of the model in the infotable).

2. Provide the necessary information manually, namely `par.est` (vector with estimated nugget, partial sill and shape parameters), `data` (used to estimate the semi-variogram model parameters), `max.dist` (semi-variogram parameter, numeric of length 1) and `nbins` (semi-variogram parameter, numeric of length 1).

Filtered bootstrap method:

For the semi-variogram model parameter estimation, the weighted least squares method is used in order to make the numerical calculation possible for large sample sizes. A filter is set up within the bootstrapping process to remove all bootstrap estimates for which the estimation algorithm for the semi-variogram model parameters did not converge.

The parameter standard errors are estimated using the generalized bootstrap method with check-based filtering. The semi-variogram structure from the given model is used to remove the spatial correlation structure within the original dataset. Then, classical bootstrap sampling with replacement is used to generate B bootstrap samples from the uncorrelated data. Each bootstrap sample inherits the correlation structure back and is used to estimate the nugget effect, partial sill and shape parameter for an exponential model. Within the bootstrap repetitions, a test is performed to check whether the estimated parameters lie within a probable range. If the total variance of the bootstrap model exceeds the empirical variance of the data times the treshold factor τ, ie.

$$c_{0b}^{*} + \sigma_{0b}^{2*} > \tau \widehat{Var(\mathbf{z})}$$

for the bth bootstrap estimate, it is discarded. Otherwise, it is saved. This procedure is performed until B bootstrap estimates have aggregated. The empirical standard deviation calculated from the bootstrap estimates provides the uncertainty estimate for each parameter.

Details about the algorithm used to obtain standard errors for the parameters of the exponential semi-variogram model are provided in Dyck and Sauzet (2022).

Reproducibility:

In order to generate reproducible bootstrap results, set a random seed with the command `set.seed()` before using the `par.uncertainty` function.

Value

The function returns parameter estimates and corresponding standard error estimates together and provides a list with the following objects:

`se`	A vector of length 3 containing the estimated standard errors of the nugget effect, the partial sill and the shape parameter.
`unc.table`	A matrix containing the parameter estimates and the corresponding standard errors.
`re_estimates`	A matrix with B rows containing the set of bootstrap re-estimates for each parameter.
`re_estimate.mean`	A vector containing the mean parameter estimates based on the set of bootstrap re-estimates for each parameter.
`call`	The function call.

References

Dyck J, Sauzet O (2022). "Parameter uncertainty estimation for exponential semivariogram models: Two generalized bootstrap methods with check- and quantile-based filtering." Preprint. https://doi.org/10.48550/arXiv.2202.05752.

Examples

```
## Example 1
# Estimate semi-variogram models:
mods = vario.mod(data = birth, max.dist = c(1000,600), nbins = 13,
                 shinyresults = FALSE, windowplots = FALSE)
print(mods$infotable)

# Estimate the parameter standard errors:

se.mod1 = par.uncertainty(vario.mod.output = mods, mod.nr = 1, B = 1000)
se.mod2 = par.uncertainty(vario.mod.output = mods, mod.nr = 2, B = 1000)


## Example 2
# Type in the specifications of the estimated exponential semi-variogram manually:

se.mod1.man = par.uncertainty(par.est = c(1021.812, 225440.3, 0),
              data = birth, max.dist = 1000, nbins = 13, B = 1000)

se.mod2.man = par.uncertainty(par.est = c(121895.486, 107232.6, 63.68720),
              data = birth, max.dist = 600, nbins = 13, B = 1000)
```

vario.mod *Semi-variogram modeling function*

Description

The function allows for user friendly exponential semi-variogram model fitting to data. Based on the gstat function variogram, vgm and fit.variogram, the function fits one or multiple exponential semi-variogram models given one or multiple maximal distances and number of bins. All estimated model parameters are summarized in an information table. Graphics of all models can be observed in a shiny application output or in several plot windows, one for each empirical semi-variogram. Additionally, a pdf file including all the figures can be saved in a specified working directory.

Usage

```
vario.mod(
  data,
  max.dist = c(2000, 1500, 1000, 750, 500, 250),
  nbins = 13,
  shinyresults = TRUE,
  windowplots = FALSE,
  pdf = FALSE,
  pdf.directory = getwd(),
  pdf.name = "Semivariograms"
)
```

Arguments

data	A data frame or matrix containing the x-coordinates in the first column, the y-coordinates in the second column (by default in meters) and the data values in the third column. The dataset may contain more attributes in further columns. In this case, a warning is provided. All columns beyond the third one are ignored.
max.dist	An optional numeric argument; the default is the vector c(2000,1500,1000,750,500,250). Either a scalar or vector containing the maximal distances can be inserted. If a vector is provided, the nbins argument must be either a scalar or a vector of the same length.
nbins	An optional argument; the default is 13 bins for all empirical semi-variograms to be estimated. Either a scalar or vector containing the number of bins can be inserted. If a vector is provided, the max.dist argument must be either a scalar or a vector of the same length.
shinyresults	A logical argument; by default TRUE. If shinyresults = T, the information table and graphics of all estimated semi-variogram models can be observed in an automatically generated shiny application.
windowplots	A logical argument; by default FALSE. If windowplots = T, all graphics are opened in new windows. They can be observed and saved manually in a wished format.

pdf	A logical argument; by default FALSE. If pdf = T, all graphics are saved in a pdf file. The file path and the name of the pdf file can be specified by the following two arguments pdf.directory and pdf.name.
pdf.directory	A character argument to specify the folder in which the pdf file is saved. If no file path is given, the pdf file is saved in the current working directory identified by getwd().
pdf.name	A character argument to specify the name of the pdf file. If no name is provided, the file is saved as 'Semivariograms.pdf'.

Details

Prespecification and Interpretation of max.dist and nbins arguments:

max.dist: only data pairs with a separation smaller than the prespecified maximal distance are included in the semi-variogram estimation. Data pairs that are separated by a higher distance are excluded.

nbins: the interval (0, max.dist] is separated into nbins equidistant lag bins or intervals, respectively. Each pairwise distance is then assigned to one of the bins. The point pair subsets $N(h_k) := \{(\mathbf{s_i}, \mathbf{s_j}) \in D| \quad ||\mathbf{s_i} - \mathbf{s_j}|| \in Bin_k\}$ are defined and a point estimate of the semi-variogram is estimated for each Bin_k for $k = 1, ...$nbins.

Empirical semi-variogram estimator:

Using the gstat function variogram an empirical semi-variogram according to Matheron's semi-variogram estimator (Matheron 1962)

$$\hat{\gamma}(h) = \frac{1}{2 \cdot |N(h)|} \sum_{(\mathbf{s_i}, \mathbf{s_j}) \in N(h)} \{Z(\mathbf{s_i}) - Z(\mathbf{s_j})\}^2$$

with $N(h)$ defined as above is obtained.

Exponential semi-variogram model:

Based on the empirical semi-variogram an exponential semi-variogram model of the form (Cressie et al. 1993)

$$\gamma_{exp}(h) = c_0 + \sigma_0^2 \Big\{ 1 - \exp\big(-\frac{h}{\phi}\big) \Big\}$$

for $h > 0$ is fitted using the vgm and fit.variogram function from package gstat via weighted least squares estimation. The weights have the form $w_k = |N(h_k)|/h_k^2$ specified by the fit.method = 7 argument within the fit.variogram function.

For the numerical optimization, starting values for the model parameters have to be provided. The initial value for the partial sill σ_0^2 equals the empirical variance of the observations. The starting value for the nugget effect c_0 is set to zero. The initial value for the shape parameter ϕ is set as max.dist divided by 3.

Result statistics:

The results for all models are automatically printed when running the function and can be found under function.output$infotable. Part of the table contains a repetition of the specified max.dist and nbins parameters as well as the estimated model parameters. The additional statistics within the infotable output are the following:

Practical range: In case of the exponential semi-variogram model, the sill $\sigma^2 = c_0 + \sigma_0^2$ is only reached asymptotically. The distance H at which $\gamma(h^* = 0.95 \cdot \sigma^2)$ is called the practical range. Formally, the practical range is defined as

$$prac.range = \frac{1}{\phi} log\Big(\frac{\sigma_0^2}{0.05(c_0 + \sigma_0^2)}\Big).$$

Relative Structural Variability (RSV): The relative structural variability is a measure of the proportion of the total variance with a spatial structure and defined as

$$RSV = \frac{\sigma_0^2}{c_0 + \sigma_0^2}.$$

Relative Bias: The relative bias describes the proportion of the total variance according to the semi-variogram model to the true total variance. It is estimated as

$$rel.bias = \frac{c_0 + \sigma_0^2}{\widehat{Var(Z)}},$$

where $\widehat{Var(Z)}$ is the sample variance or empirical variance of the attribute of interest of the dataset at hand. A relative bias of 1 indicates equality of sample variance and variance according to the semi-variogram model.

For more details, see Schabenberger and Gotway (2017).

Value

A list containing the following arguments:

`infotable`	A table containing the statistics of all estimated exponential semi-variogram models. Each row corresponds to one model. Shown are the prespecified `max.dist` and `nbins` values, the parameter estimates for the nugget effect, partial sill and shape, the resulting estimated practical range, the relative structured variability (RSV) and the relative bias.
`variog.list`	A list: each list entry contains the `variog` output with further information on the estimated empirical semi-variogram.
`vmod.list`	A list: each list entry contains the `variofit` output with further information on the fitted parametric semi-variogram model.
`input.arguments`	A list containing the evaluated input arguments, namely the `$data` used to fit the exponential semi-variogram, the `$max.dist` and `$nbins` specifications and the specifications for the pdf-output, `$pdf`, `$pdf.directory` and `$pdf.name`.
`call`	Contains the call of the function.

The models are visualized in an automatically opened shiny application if `shinyresults = T`. Beware that in this case the output of the `vario.mod` function is not saved in the environment, even with a variable name assigned. In order to save the output, set `shinyresults = F`.

If the argument `windowplots = T`, one or multiple graphics of the estimated empirical semi-variograms and semi-variogram models are plotted in the R environment. If the argument `pdf = T`, a pdf file containing the same figures is saved in the manually specified or current working directory.

References

Cressie N, Ribeiro PJ, Diggle PJ (1993). *Statistics for spatial data*, Rev. ed. edition. Wiley, New York. ISBN 9781119115151, https://onlinelibrary.wiley.com/doi/book/10.1002/9781119115151.

Matheron G (1962). *Traité de géostatistique appliquée. 1 (1962)*, volume 1. Editions Technip.

Schabenberger O, Gotway CA (2017). *Statistical methods for spatial data analysis*. CRC press.

See Also

`variogram` in the `gstat` package for further information on the estimation of the empirical semi-variogram;

`fit.variogram` and `vgm` in the `gstat` package for further information on the default settings when fitting an exponential semi-variogram model to an empirical semi-variogram.

Examples

```
if(interactive()){

## Example 1
# Default options:
vario.mod(data = birth)

# This is equal to
vario.mod(data = birth, max.dist = c(2000,1500,1000,750,500,250), nbins = 13,
          shinyresults = TRUE, windowplots = FALSE,
          pdf = FALSE, pdf.directory = getwd(), pdf.name = "Semivariograms")

## Example 2
# Open graphics in regular windows and not in shiny application:
vario.mod(data = birth, max.dist = c(2000,1500,1000,750,500,250), nbins = 15:10,
          shinyresults = FALSE, windowplots = TRUE)

## Example 3
# Generate a pdf with the following command:
vario.mod(data = birth, shinyresults = FALSE, windowplots = FALSE,
          pdf = TRUE, pdf.directory = getwd())
# You find a pdf file in your current working directory.

}
```

`vario.reg.prep` *Adjustment for covariates*

Description

Given a linear regression output of class 'lm' or 'lmerMod' with the attribute of interest as dependent variable, the function provides a dataset containing the coordinates of the original observations and the studentized residuals of the regression model.

Usage

```
vario.reg.prep(reg, data = NULL)
```

Arguments

`reg`	An object of class 'lm' obtained as result of a linear regression using the function `lm` from the package `stats` or an object of class 'lmerMod' obtained as result of a linear mixed model regression using the function `lmer` from the package `lme4`.

data	Only needed if the data argument within the regression function lm/lmer is not provided: A data frame containing the geo-coded dataset containing the Cartesian x- and y-coordinates in the first and second column, the outcome of interest in the third column and all covariates used for the regression in further columns.

Details

The adjusted outcome is defined as the student residuals of the linear or linear mixed regression model. They are calculated using the rstudent function from package stats. In case of a mixed model, the adjusted variable vector resembles the conditional studentized residuals.

The geo-coded dataset used for the regression is extracted from the current environment. In order to work, the dataset has to be loaded into the environment prior to the use of vario.reg.prep.

If the data argument was specified within the regression function lm/lmer, vario.reg.prep automatically extracts the name of the dataset used for regression and calls it from the current environment. Otherwise, the dataset has to be provided manually as input argument within vario.reg.prep.

Value

A data frame with three columns:

x	x-coordinate in the first column.
y	y-coordinate in the second column.
adj	Studentized residuals to be used as new variable adjusted for covariates.

See Also

lm in the stats package for information on the fitting of a linear regression model;

lmer in the lme4 package for information on the fitting of a linear mixed regression model;

rstudent in the stats package for information on how the attribute of interest is adjusted for covariates.

Examples

```
## Example 1
head(birth) #geo-coded dataset
hist(birth$birthweight) # attribute of interest

# Linear regression model
mod1 = lm(birthweight ~ primiparous + datediff + bmi
+ factor(inc), data = birth)
summary(mod1)
data.adj1 = vario.reg.prep(mod1)

head(data.adj1)
hist(data.adj1$adj) # adjusted attribute of interest
# The data frame can be used as input for the vario.mod function.


## Example 2
# Data argument within lm not provided (not recommended, but possible):
mod2 = lm(birth$birthweight ~ birth$primiparous + birth$datediff + birth$bmi
+ factor(birth$inc))
summary(mod2)
# In this case, make sure to provide the data argument here:
```

```
data.adj2 = vario.reg.prep(reg = mod2, data = birth)


## Example 3
# Linear mixed regression model
library(lme4)
mod3 = lmer(birthweight ~ primiparous + datediff
                + bmi + (1|inc), data = birth)
summary(mod3)
data.adj3 = vario.reg.prep(mod3)


## Example 4
# Data argument within lmer not provided (not recommended, but possible):
mod4 = lmer(birth$birthweight ~ birth$primiparous + birth$datediff
            + birth$bmi + (1|birth$inc))
summary(mod4)
# In this case, make sure to provide the data argument here:
data.adj4 = vario.reg.prep(reg = mod4, data = birth)
```

Index