The geoR package

July 9, 2012

Version 1.7-4

Date 2012-06-29

Title Analysis of geostatistical data

LazyLoad yes

LazyData yes

Author Paulo J. Ribeiro Jr paulojus@ufpr.br and Peter J. Diggle p.diggle@lancaster.ac.uk

Maintainer Paulo J. Ribeiro Jr paulojus@ufpr.br

Depends R (¿= 2.10), stats, sp, methods, MASS

Imports splancs, RandomFields

Suggests scatterplot3d, geoRglm, lattice, graphics, tcltk

Description Geostatistical analysis including traditional, likelihood-based and Bayesian methods.

License GPL (¿= 2)

URL http://www.leg.ufpr.br/geoR

topics documented: .nlmPAdapts nlm for Constraints in the Parameter Values.nlmP spatial.nlmP optimize.nlmP This function adapts the function nlmnlm to allow for constraints (upper and/or lower bounds) in the values of the parameters.

```
.nlmP(objfunc, params, lower=rep(-Inf, length(params)),
      upper=rep(+Inf, length(params)), ...)
```

the function to be minimized.

starting values for the parameters.

lower bounds for the variables. Defaults to -Inf.

upper bounds for the variables. Defaults to -Inf.

further arguments to be passed to the function nlmnlm. Constraints on the parameter values are internally imposed by using exponential, logarithmic, and logit transformation of the parameter values. The output is the same as for the function nlmnlm. Patrick E. Brown p.brown@lancaster.ac.uk.
Adapted and included in geoR by
Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br
Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. nlmnlm, optimoptim. as.geodataConverts an Object to the Class "geodata"as.geodata as.data.frame.geodataas.geodataas.data.frame.geodata
as.geodata.defaultas.geodataas.geodata.default as.geodata.geodata.frameas.geodataas.geodata.geodata.
as.geodata.SpatialPointsDataFrameas.geodataas.geodata.SpatialPointsDataFrame
geodataas.geodatageodata is.geodataas.geodatais.geodata spatialas.geodata classesas.geodata manipas.geodata The default method converts a matrix or a data-frame to an object of the classclass "geodata".
Objects of the class "geodata" are lists with two obligatory components: coords and data. Optional components are allowed and a typical example is a vector or matrix with covariate(s) values.

```
as.geodata(obj, ...)

## Default S3 method:
as.geodata(obj, coords.col = 1:2, data.col = 3, data.names = NULL,
                   covar.col = NULL, covar.names = "obj.names",
                   units.m.col = NULL, realisations = NULL,
                   na.action = c("ifany", "ifdata", "ifcovar", "none"),
                   rep.data.action, rep.covar.action, rep.units.action,
                   ...)

## S3 method for class 'geodata'
as.data.frame(x, ..., borders = TRUE)

## S3 method for class 'geodata.frame'
as.geodata(obj, ...)

## S3 method for class 'SpatialPointsDataFrame'
as.geodata(obj, data.col = 1, ...)

is.geodata(x)
```

a matrix or data-frame where each line corresponds to one spatial location. It should contain values of 2D coordinates, data and, optionally, covariate(s) value(s) at the locations. A method for SpatialPointsDataFrameSpatialPointsDataFrame is also provided. It can also take an output of the function grfgrf, see DETAILS below.

a vector with the column numbers corresponding to the spatial coordinates.

a scalar or vector with column number(s) corresponding to the data.

optional. A string or vector of strings with names for the data columns. Only valid if there is more than one column of data. By default, takes the names from the original object.

optional. A scalar or numeric vector with the column number(s) corresponding to the covariate(s). Alternativelly can be a character vector with the names of the covariates.

optional. A string or vector of strings with the name(s) of the covariates. By default take the names from the original object.

optional. A scalar with the column number corresponding to the offset variable. Alternativelly can be a character vector with the name of the offset. This option is particularly relevant when using the package geoRglm. All values must be greater then zero.

optional. A vector indicating the realisation number or a number indicating a column in obj with the realisation indicator variable. See DETAILS below.

string defining action to be taken in the presence of NA's. The default option "ifany" excludes all points for which there are NA's in the data or covariates. The option "ifdata" excludes points for which there are NA's in the data. The default option "ifcovar" excludes all points for which there are NA's in the covariates. The option "none" do not exclude points.

a string or a function. Defines action to be taken when there is more than one data at the same location. The default option "none" keeps the repeated locations, if any. The option "first" retains only the first data recorded at each location. Alternativelly a function can be passed and it will be used. For instance if mean is provided, the function will compute and return the average of the data at coincident locations. The non-default options will eliminate the repeated locations.

idem to rep.data.locations, to be applied to the covariates, if any. Defaults to the same option set for rep.data.locations.

an object which is tested for the class geodata.

a string or a function. Defines action to be taken on the element units.m, if present when there is more than one data at the same location. The default option is the same value set for rep.data.action.

logical. If TRUE the element borders in the geodata object is set as an attribute of the data-frame.

values to be passed for the methods. Objects of the class "geodata" contain data for geostatistical analysis using the package geoR. Storing data in this format facilitates the usage of the functions in geoR. However, conversion of objects to this class is not obligatory to carry out the analysis.

NA's are not allowed in the coordinates. By default the respective rows will not be included in the output.

Realisations

Tipically geostatistical data correspond to a unique realisation of the spatial process. However, sometimes different "realisations" are possible. For instance, if data are collected in the same area at different points in time and independence between time points is assumed, each time can be considered a different "replicate" or "realisation" of the same process. The argument realisations takes a vector indication the replication number and can be passed to other geoR functions as, for instance, likfitlikfit.

The data format is similar to the usual geodata format in geoR. Suppose there are realisations (times) 1, ..., J and for each realisations $n_1, ..., n_j observations are available. The coordinates for different realisations should be combined in a sing$

grf objects

If an object of the class grf is provided the functions just extracts the elements coords and data of this object. An object of the classclass "geodata" which is a list with two obligatory components (coords and data) and other optional components:

an n $\times 2 matrix where n is the number of spatial locations. a vector of length n, for the univariate case or, an n$ $frame for the multivariate case, where v is the number of variables.$

a vector of length n or an n $\times p matrix with covariate(s) values, where p is the number of covariates. Only re$

Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,

Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:

http://www.leg.ufpr.br/geoR. read.geodataread.geodata for reading data from an *ASCII* file and listlist for general information on lists. Not run: converting the data-set "topo" from the package MASS (VR's bundle) to the geodata format: if(require(MASS)) topo topogeo ¡- as.geodata(topo) names(topogeo) topogeo

End(Not run) boxcoxThe Box-Cox Transformed Normal Distributionbox-cox dboxcoxboxcoxdboxcox rboxcoxboxcoxrboxcox distributionboxcox Functions related with the Box-Cox family of transformations. Density and random generation for the Box-Cox transformed normal distribution with mean equal to mean and standard deviation equal to sd, *in the normal scale.*

```
rboxcox(n, lambda, lambda2 = NULL, mean = 0, sd = 1)

dboxcox(x, lambda, lambda2 = NULL, mean = 0, sd = 1)
```

numerical value(s) for the transformation parameter $\lambda. logical or numerical value(s) of the additional transformation (see DETAILS below). Defaults to NUL$

number of observations to be generated.

a vector of quantiles (dboxcox) or an output of boxcoxfit (print, plot, lines).

a vector of mean values at the normal scale.

a vector of standard deviations at the normal scale. Denote Y the variable at the original scale and Y' the transformed variable. The Box-Cox transformation is defined by:

$$Y' = \begin{cases} log(Y) \text{ , if } \lambda = 0 \\ \frac{Y^\lambda - 1}{\lambda} \text{ , otherwise} \end{cases} .$$

An additional shifting parameter $\lambda_2 can be included in which case the transformation is given by :$

$$Y' = \begin{cases} log(Y + \lambda_2) \text{ , } \lambda = 0 \\ \frac{(Y+\lambda_2)^\lambda - 1}{\lambda} \text{ , otherwise} \end{cases} .$$

The function rboxcox samples Y' from the normal distribution using the function rnormrnorm and backtransform the values according to the equations above to obtain values of Y. If necessary the back-transformation truncates the values such that Y' $\geq \frac{1}{\lambda} results in Y = 0 in the original scale. Increasing the value of the mean and/or reducing the variance might$

The functions returns the following results:

a vector of random deviates.

a vector of densities. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br, Peter J. Diggle p.diggle@lancaster.ac.uk. Box, G.E.P. and Cox, D.R.(1964) An analysis of transformations. JRSS B 26:211–246. The parameter estimation function boxcoxfitboxcoxfit, the function boxcoxboxcox in the package MASS and the function box.coxbox.cox in the package car. Simulating data simul ¡- rboxcox(100, lambda=0.5, mean=10, sd=2) Comparing models with different lambdas, zero means and unit variances curve(dboxcox(x, lambda=-1), 0, 8) for(lambda in seq(-.5, 1.5, by=0.5)) curve(dboxcox(x, lambda), 0, 8, add = TRUE) boxcox.geodataBox-Cox transformation for geodata objectsboxcox.geodata regressionboxcox.geodata modelsboxcox.geodata hplotboxcox.geodata Method for Box-Cox transformation for objects of the class geodata assuming the data are independent. Computes and optionally plots profile log-likelihoods for the parameter of the Box-Cox simple power transformation y$^l ambda$.

```
## S3 method for class 'geodata'
boxcox(object, trend = "cte", ...)
```

an object of the class geodata. See as.geodataas.geodata.

specifies the mean part of the model. See trend.spatialtrend.spatial for further details. Defaults to "cte".

arguments to be passed for the function boxcoxboxcox. This is just a wrapper for the function boxcoxboxcox facilitating its usage with geodata objects.

Notice this assume independent observations which is typically not the case for geodata objects. A list of the lambda vector and the computed profile log-likelihood vector, invisibly if the result is plotted. boxcoxboxcox for parameter estimation results for independent data and likfitlikfit for parameter estimation within the geostatistical model. if(require(MASS)) boxcox(wolfcamp)

data(ca20) boxcox(ca20, trend = altitude) boxcoxfitParameter Estimation for the Box-Cox Transformationboxcoxfit .negloglik.boxcoxboxcoxfit.negloglik.boxcox lines.boxcoxfitboxcoxfitlines.boxcoxfit plot.boxcoxfitboxcoxfitplot.boxcoxfit print.boxcoxfitboxcoxfitprint.boxcoxfit regressionboxcoxfit modelsboxcoxfit hplotboxcoxfit Parameter estimation and plotting of the results for the Box-Cox transformed normal distribution.

```
boxcoxfit(object, xmat, lambda, lambda2 = NULL, add.to.data = 0, ...)

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

## S3 method for class 'boxcoxfit'
plot(x, hist = TRUE, data = eval(x$call$object), ...)

## S3 method for class 'boxcoxfit'
lines(x, data = eval(x$call$object), ...)
```

a vector with the data.

a matrix with covariates values. Defaults to rep(1, length(y)).

numerical value(s) for the transformation parameter $\lambda.Usedastheinitialvalueinthefunctionforparameterestimation.Ifnotprovideddefaultvaluesareassu$

a constant value to be added to the data.

a list, typically an output of the function boxcoxfit.

logical indicating whether histograms should to be plotted.

data values.

extra parameters to be passed to the minimization function optimoptim (boxcoxfit), histhist (plot) or curvecurve (lines).
The functions returns the following results:

a list with estimated parameters and results on the numerical minimization.

print estimated parameters. No values returned.

plots histogram of the data (optional) and the model. No values returned. This function is only valid if covariates are not included in boxcoxfit.

adds a line with the fitted model to the current plot. No values returned. This function is only valid if covariates are not included in boxcoxfit. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Box, G.E.P. and Cox, D.R.(1964) An analysis of transformations. JRSS B 26:211–246. rboxcoxrboxcox and dboxcoxdboxcox for the expression and more on the Box-Cox transformation, the minimization function optimoptim, the function boxcoxboxcox in the package MASS and the function box.coxbox.cox in the package car. set.seed(384) Simulating data simul ¡- rboxcox(100, lambda=0.5, mean=10, sd=2) Finding the ML estimates ml ¡- boxcoxfit(simul) ml Ploting histogram and fitted model plot(ml) Comparing models with different lambdas, zero means and unit variances curve(dboxcox(x, lambda=-1), 0, 8) for(lambda in seq(-.5, 1.5, by=0.5)) curve(dboxcox(x, lambda), 0, 8, add = TRUE) Another example, now estimating lambda2 simul ¡- rboxcox(100, lambda=0.5, mean=10, sd=2) ml ¡- boxcoxfit(simul, lambda2 = TRUE) ml plot(ml) An example with a regression model boxcoxfit(object = trees[,3], xmat = trees[,1:2]) ca20Calcium content in soil samplesca20 datasetsca20 This data set contains the calcium content measured in soil samples taken from the 0-20cm layer at 178 locations within a certain study area divided in three sub-areas. The elevation at each location was also recorded.

The first region is typically flooded during the rain season and not used as an experimental area. The calcium levels would represent the natural content in the region. The second region has received fertilisers a while ago and is typically occupied by rice fields. The third region has received fertilisers recently and is frequently used as an experimental area.

```
data(ca20)
```

The object ca20 belongs to the class geodata and is a list with the following elements:

coords a matrix with the coordinates of the soil samples.

data calcium content measured in mmol$_c/dm^3.adata - framewiththecovariates$

covariate **altitude** a vector with the elevation of each sampling location, in meters (m).

area a factor indicating the sub area to which the locations belongs.

a matrix with the coordinates defining the borders of the area.

a matrix with the coordinates of the limits of the sub-area 1.

a matrix with the coordinates of the limits of the sub-area 2.

a matrix with the coordinates of the limits of the sub-area 3.

The data was collected by researchers from PESAGRO and EMBRAPA-Solos, Rio de Janeiro, Brasil and provided by Dra. Maria Cristina Neves de Oliveira.

Capeche, C.L.; Macedo, J.R.; Manzatto, H.R.H.; Silva, E.F. (1997) Caracterização pedológica da fazenda Angra - PESAGRO/RIO - Estação experimental de Campos (RJ). (compact disc). In: Congresso BRASILEIRO de Ciência do Solo. 26., Informação, globalização, uso do solo; Rio de Janeiro, 1997. trabalhos. Rio de Janeiro: Embrapa/SBCS.

Oliveira, M.C.N. (2003) *Métodos de estimação de parâmetros em modelos geoestatísticos com diferentes estruturas de covariâncias: uma aplicação ao teor de cálcio no solo.* Tese de Doutorado, ESALQ/USP/Brasil.

Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. camgCalcium and magnesium content in soil samplescamg datasetscamg This data set contains the calcium content measured in soil samples taken from the 0-20cm layer at 178 locations within a certain study area divided in three sub-areas. The elevation at each location was also recorded.

The first region is tipically flooded during the rain season and not used as an experimental area. The calcium levels would represent the natural content in the region. The second region has received fertilizers a while ago and is tipically occupied by rice fields. The third region has recieved fertilizers recently and is frequently used as an experimental area.

```
data(camg)
```

A data frame with 178 observations on the following 10 variables.

east east-west coordinates, in meters.

north north-south coordinates, in meters.

elevation elevation, in meters

region a factor where numbers indicate different sub-regions within the area

ca020 calcium content in the 0-20cm soil layer, measured in mmol$_c/dm^3.calciumcontentinthe0 - 20cmsoillayer, measuredinmmol_c/dm^3$.

mg020 calcium content in the 0-20cm soil layer.

ca2040 calcium content in the 20-40cm soil layer, measured in mmol$_c/dm^3.calciumcontentinthe20 - 40cmsoillayer, measuredinmmol_c/dm^3$.

mg2040mg2040 calcium content in the 20-40cm soil layer.

More details about this data-set, including coordinates of the region and sub-region borders can be found in the data object ca20ca20. The data was collected by researchers from PESAGRO and EMBRAPA-Solos, Rio de Janeiro, Brasil and provided by Dra. Maria Cristina Neves de Oliveira.

Capeche, C.L.; Macedo, J.R.; Manzatto, H.R.H.; Silva, E.F. (1997) Caracterização pedológica da fazenda Angra - PESAGRO/RIO - Estação experimental de Campos (RJ). (compact disc). In: Congresso BRASILEIRO de Ciência do Solo. 26., Informação, globalização, uso do solo; Rio de Janeiro, 1997. trabalhos. Rio de Janeiro: Embrapa/SBCS. plot(camg[-(1:2),]) mg20 ¡- as.geodata(camg, data.col=6) plot(mg20) points(mg20) coords.anisoGeometric Anisotropy Correctioncoords.aniso spatialcoords.aniso Transforms or back-transforms a set of coordinates according to the geometric anisotropy parameters.

```
coords.aniso(coords, aniso.pars, reverse = FALSE)
```

an n $\times 2 matrix with the coordinates to be transformed. a vector with two elements, \psi_A and \psi_R, the anisotrop$

logical. Defaults to FALSE. If TRUE the reverse transformation is performed. Geometric anisotropy is defined by two parameters:

Anisotropy angle defined here as the azimuth angle of the direction with greater spatial continuity, i.e. the angle between the *y-axis* and the direction with the maximum range.

Anisotropy ratio defined here as the ratio between the ranges of the directions with greater and smaller continuity, i.e. the ratio between maximum and minimum ranges. Therefore, its value is always greater or equal to one.

If reverse = FALSE (the default) the coordinates are transformed from the *anisotropic space* to the *isotropic space.* The transformation consists in multiplying the original coordinates by a rotation matrix R and a shrinking matrix T, as follows: X$_m$ = $XRT, where X_m is a matrix with the modified coordinates (isotropic space), X is a matrix with original coor$

If reverse = TRUE, the back-transformation is performed, i.e. transforming the coordinates from the *isotropic space* to the *anisotropic space* by computing: X = X$_m(RT)^{-1}$

An n $\times 2 matrix with the transformed coordinates.$ Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br

Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:

http://www.leg.ufpr.br/geoR. op ¡- par(no.readonly = TRUE) par(mfrow=c(3,2)) par(mar=c(2.5,0,0,0)) par(mgp=c(2,.5,0)) par(pty="s") Defining a set of coordinates coords ¡- expand.grid(seq(-1, 1, l=3), seq(-1, 1, l=5)) plot(c(-1.5, 1.5), c(-1.5, 1.5), xlab="", ylab="", type="n") text(coords[,1], coords[,2], 1:nrow(coords)) Transforming coordinates according to some anisotropy parameters coordsA ¡- coords.aniso(coords, aniso.pars=c(0, 2)) plot(c(-1.5, 1.5), c(-1.5, 1.5), xlab="", ylab="", type="n") text(coordsA[,1], coordsA[,2], 1:nrow(coords)) coordsB ¡- coords.aniso(coords, aniso.pars=c(pi/2, 2)) plot(c(-1.5, 1.5), c(-1.5, 1.5), xlab="", ylab="", type="n") text(coordsB[,1], coordsB[,2], 1:nrow(coords)) coordsC ¡- coords.aniso(coords, aniso.pars=c(pi/4, 2)) plot(c(-1.5, 1.5), c(-1.5, 1.5), xlab="", ylab="", type="n") text(coordsC[,1], coordsC[,2], 1:nrow(coords)) coordsD ¡- coords.aniso(coords, aniso.pars=c(3*pi/4, 2)) plot(c(-1.5, 1.5), c(-1.5, 1.5), xlab="", ylab="", type="n") text(coordsD[,1], coordsD[,2], 1:nrow(coords)) coordsE ¡- coords.aniso(coords, aniso.pars=c(0, 5)) plot(c(-1.5, 1.5), c(-1.5, 1.5), xlab="", ylab="", type="n") text(coordsE[,1], coordsE[,2], 1:nrow(coords)) par(op) coords2coordsOperations on Coordinatescoords2coords rect.coordscoords2coordsrect.coords zoom.coordscoords2coordszoom.coords zoom.coords.defaultcoords2coordszoom.coords.default zoom.coords.geodatacoords2coordszoom.coords.geodata spatialcoords2coords Functions for shifting, zooming and envolving rectangle of a set of coordinates.

```
coords2coords(coords, xlim, ylim, xlim.ori, ylim.ori)

zoom.coords(x, ...)

## Default S3 method:
zoom.coords(x, xzoom, yzoom, xlim.ori, ylim.ori, xoff=0, yoff=0, ...)

## S3 method for class 'geodata'
zoom.coords(x, ...)

rect.coords(coords, xzoom = 1, yzoom=xzoom, add.to.plot=TRUE,
            quiet = FALSE, ...)
```

two column matrix or data-frame with coordinates.

range of the new x-coordinates.

range of the new y-coordinates.

optional. Range of the original x-coordinates, by default the range of the original x-coordinates.

optional. Range of the original y-coordinates, by default the range of the original y-coordinates.

scalar, expanding factor in the x-direction.

scalar, expanding factor in the y-direction.

scalar, shift in the x-direction.

scalar, shift in the y-direction.

logical, if TRUE the retangle is added to the current plot.

logical, none is returned.

further arguments to be passed to rectrect.

return an object of the same type as given in the argument coords with the transformed coordinates.

returns a matrix with the 4 coordinates of the rectangle defined by the coordinates. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. subareasubarea, rectrect foo ¡- matrix(c(4,6,6,4,2,2,4,4), nc=2) foo1 ¡- zoom.coords(foo, 2) foo1 foo2 ¡- coords2coords(foo, c(6,10), c(6,10)) foo2 plot(1:10, 1:10, type="n") polygon(foo) polygon(foo1, lty=2) polygon(foo2, lwd=2) arrows(foo[,1],

foo[,2],foo1[,1],foo1[,2], lty=2) arrows(foo[,1], foo[,2],foo2[,1],foo2[,2]) legend("topleft", c("foo", "foo1 (zoom.coords)", "foo2 (coords2coords)"), lty=c(1,2,1), lwd=c(1,1,2))

"zooming" part of The Gambia map gb ¡- gambia.borders/1000 gd ¡- gambia[,1:2]/1000 plot(gb, ty="l", asp=1, xlab="W-E (kilometres)", ylab="N-S (kilometres)") points(gd, pch=19, cex=0.5) r1b ¡- gb[gb[,1] ¡ 420,] rc1 ¡- rect.coords(r1b, lty=2)

r1bn ¡- zoom.coords(r1b, 1.8, xoff=90, yoff=-90) rc2 ¡- rect.coords(r1bn, xz=1.05) segments(rc1[c(1,3),1],rc1[c(1,3),2],rc2[c(1,3),1],rc2[c(1,3),2], lty=3)

lines(r1bn) r1d ¡- gd[gd[,1] ¡ 420,] r1dn ¡- zoom.coords(r1d, 1.7, xlim.o=range(r1b[,1],na.rm=TRUE), ylim.o=range(r1b[,2],na.rm=TRUE), xoff=90, yoff=-90) points(r1dn, pch=19, cex=0.5) text(450,1340, "Western Region", cex=1.5) cov.spatialComputes Value of the Covariance Functioncov.spatial .check.cov.modelcov.spatial.check.cov.model .cor.numbercov.spatial.cor.number geoRCovModelscov.spatialgeoRCovModels spatialcov.spatial modelscov.spatial Computes the covariances for pairs variables, given the separation distance of their locations. Options for different correlation functions are available. The results can be seen as a change of metric, from the *Euclidean distances* to *covariances.*

```
cov.spatial(obj, cov.model= "matern",
            cov.pars=stop("no cov.pars argument provided"),
            kappa = 0.5)
```

a numeric object (vector or matrix), typically with values of distances between pairs of spatial locations.

string indicating the type of the correlation function. Available choices are: "matern", "exponential", "gaussian", "spherical", "circular", "cubic", "wave", "power", "powered.exponential", "cauchy", "gencauchy", "gneiting", "gneiting.matern", "pure.nugget". See section DETAILS for available options and expressions of the correlation functions.

a vector with 2 elements or an ns $\times 2 matrix with the covariance parameters. The first element (if a vector)$

numerical value for the additional smoothness parameter of the correlation function. Only required by the following correlation functions: "matern", "powered.exponential", "cauchy", "gencauchy" and "gneiting.matern". Covariance functions return the value of the covariance C(h) between a pair variables located at points separated by the distance h. The covariance function can be written as a product of a variance parameter $\sigma^2 times a positive definite correlation function \rho(h)$: $C(h) = \sigma^2 \rho(h). The expressions of the covariance functions available in geoR are given below. We recomm$

Denote $\phi the basic parameter of the correlation function and name it the range parameter. Some of the co$

cauchy

$\rho(h) = [1 + (\frac{h}{\phi})^2]^{-\kappa}$

gencauchy (generalised Cauchy)

$\rho(h) = [1 + (\frac{h}{\phi})^{\kappa_2}]^{-\kappa_1/\kappa_2}, \kappa_1 > 0, 0 < \kappa_2 \leq 2$

circular

Let $\theta = \min(\frac{h}{\phi}, 1) and g(h) = 2\frac{(\theta\sqrt{1-\theta^2}+\sin^{-1}\sqrt{\theta})}{\pi}. Then, the circular model is given by:$

$$\rho(h) = \begin{cases} 1 - g(h) \text{ , if } h < \phi \\ 0 \text{ , otherwise} \end{cases}$$

cubic
$\rho(h) = \begin{cases} 1 - [7(\frac{h}{\phi})^2 - 8.75(\frac{h}{\phi})^3 + 3.5(\frac{h}{\phi})^5 - 0.75(\frac{h}{\phi})^7] \text{ , if } h < \phi \\ 0 \text{ , otherwise.} \end{cases}$

gaussian
$\rho(h) = \exp[-(\frac{h}{\phi})^2]$

exponential
$\rho(h) = \exp(-\frac{h}{\phi})$

matern
$\rho(h) = \frac{1}{2^{\kappa-1}\Gamma(\kappa)}(\frac{h}{\phi})^\kappa K_\kappa(\frac{h}{\phi})$

spherical
$\rho(h) = \begin{cases} 1 - 1.5\frac{h}{\phi} + 0.5(\frac{h}{\phi})^3 \text{ , if } h \text{ ¡ } \phi \\ 0 \text{ , otherwise} \end{cases}$

power (and linear)
The parameters of the this model $\sigma^2 and \phi cannot be interpreted as partial\ sill and range as for the other models. This mod$
$\gamma(h) = \sigma^2 h^\phi \text{ , } 0 < \phi < 2, \sigma^2 > 0$

Since the corresponding process is not second order stationary the covariance and correlation functions are not defined. For internal calculations the *geoR* functions uses the fact the this model possesses locally stationary representations with covariance functions of the form: $C_(h) = \sigma^2(A - h^\phi), where A is a suitable constant as given in Chilès \& Delfiner (pag. 511, eq. 7.35)$.

The *linear* model corresponds a particular case with $\phi = 1$.

powered.exponential (or stable)
$\rho(h) = \exp[-(\frac{h}{\phi})^\kappa] \text{ , } 0 < \kappa \leq 2$

gneiting
C(h)=$\left(1 + 8sh + 25(sh)^2 + 32(sh)^3\right) (1-sh)^8 1_{[0,1]}(sh) where s = 0.301187465825. For further details see documenta$

It is an alternative to the gaussian model since its graph is visually hardly distinguishable from the graph of the G

gneiting.matern
Let $\alpha = \phi\kappa_2, \rho_m(\cdot) denotes the$ Matérn $model and \rho_g(\cdot) the Gneiting model. Then the$ Gneiting-Matérn $is given by \rho(h) = \rho$

wave
$\rho(h) = \frac{\phi}{h}\sin(\frac{h}{\phi})$

pure.nugget
$\rho(h) = k$
$where k is a constant value. This model corresponds to no spatial correlation.$

Nested models Models with several structures usually called *nested models* in the geostatistical literature are also allowed. In this case the argument cov.pars takes a matrix and cov.model and lambda can either have length equal to the number of rows of this matrix or length 1. For the latter cov.model and/or lambda are recycled, i.e. the same value is used for all structures.
The function returns values of the covariances corresponding to the given distances. The type of output is the same as the type of the object provided in the argument obj, typically a vector, matrix or array. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk.

For a review on correlation functions:
Schlather, M. (1999) *An introduction to positive definite functions and to un-*

conditional simulation of random fields. Technical report ST 99-10, Dept. of Maths and Statistics, Lancaster University.

Chilès, J.P. and Delfiner, P. (1999) Geostatistics: Modelling Spatial Uncertainty, Wiley.

Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. maternmatern for computation of the Matérn model, besselKbesselK for computation of the Bessel function and varcov.spatialvarcov.spatial for computations related to the covariance matrix. Variogram models with the same "practical" range: v.f ¡- function(x, ...)1-cov.spatial(x, ...) curve(v.f(x, cov.pars=c(1, .2)), from = 0, to = 1, xlab = "distance", ylab = expression(gamma(h)), main = "variograms with equivalent p̈ractical range") curve(v.f(x, cov.pars = c(1, .6), cov.model = "sph"), 0, 1, add = TRUE, lty = 2) curve(v.f(x, cov.pars = c(1, .6/sqrt(3)), cov.model = "gau"), 0, 1, add = TRUE, lwd = 2) legend("topleft", c("exponential", "spherical", "gaussian"), lty=c(1,2,1), lwd=c(1,1,2)) Matern models with equivalent "practical range" and varying smoothness parameter curve(v.f(x, cov.pars = c(1, 0.25), kappa = 0.5),from = 0, to = 1, xlab = "distance", ylab = expression(gamma(h)), lty = 2, main = "models with equivalent p̈racticalr̈ange") curve(v.f(x, cov.pars = c(1, 0.188), kappa = 1),from = 0, to = 1, add = TRUE) curve(v.f(x, cov.pars = c(1, 0.14), kappa = 2),from = 0, to = 1, add = TRUE, lwd=2, lty=2) curve(v.f(x, cov.pars = c(1, 0.117), kappa = 2),from = 0, to = 1, add = TRUE, lwd=2) legend("bottomright", expression(list(kappa == 0.5, phi == 0.250), list(kappa == 1, phi == 0.188), list(kappa == 2, phi == 0.140), list(kappa == 3, phi == 0.117)), lty=c(2,1,2,1), lwd=c(1,1,2,2)) plotting a nested variogram model curve(v.f(x, cov.pars = rbind(c(.4, .2), c(.6,.3)), cov.model = c("sph","exp")), 0, 1, ylab='nested model') dup.coordsLocates duplicated coordinatesdup.coords dup.coords.defaultdup.coordsdup.coords.default dup.coords.geodatadup.coordsdup.coords.geodata duplicated.geodatadup.coordsduplicated.geodata spatialdup.coords manipdup.coords This funtions takes an object with 2-D coordinates and returns the positions of the duplicated coordinates. Also sets a method for duplicated

```
dup.coords(x, ...)
## Default S3 method:
dup.coords(x, ...)
## S3 method for class 'geodata'
dup.coords(x, incomparables, ...)
## S3 method for class 'geodata'
duplicated(x, incomparables, ...)
```

a two column numeric matrix or data frame.

unused. Just for compatibility with the generic function duplicatedduplicated.

arguments passed to sapplysapply. If simplify = TRUE (default) results are returned as an array if possible (when the number of replicates are the same at each replicated location) Function and methods returns NULL if there are no duplicates locations.

Otherwise, the default method returns a list where each component is a vector with the positions or the rownames, if available, of the duplicates coordinates.

The method for geodata returns a data-frame with rownames equals to the positions of the duplicated coordinates, the first column is a factor indicating duplicates and the remaning are output of as.data.frame.geodataas.data.frame.geodata. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br
Peter J. Diggle p.diggle@lancaster.ac.uk. as.geodataas.geodata for the definition of geodata class, duplicatedduplicated for the base function to identify duplicated values and jitterDupCoordsjitterDupCoords for a function which jitters duplicated coordinates. simulating data dt ¡- grf(30, cov.p=c(1, .3)) ”forcing” some duplicated locations dt$coords[4,] < -dt$coords[14,] ¡- dt$coords[24,] < -dt$coords[2,] dt$coords[17,] < -dt$coords[23,] ¡- dt$coords[8,] output of the method for geodata dup.coords(dt) which is the same as a method for duplicated() duplicated(dt$ dup.coords(dt$coords) elevationSurface Elevationselevation datasetselevation Surface elevation data taken from Davis (1972). An onject of the class geodata with elevation values at 52 locations.

```
data(elevation)
```

An object of the class geodata which is a list with the following elements:

coords x-y coordinates (multiples of 50 feet).

data elevations (multiples of 10 feet).

Davis, J.C. (1973) *Statistics and Data Analysis in Geology.* Wiley. summary(elevation) plot(elevation) eyefitInteractive Variogram Estimationeyefit lines.eyefiteyefitlines.eyefit plot.eyefiteyefitplot.eyefit print.eyefiteyefitprint.eyefit print.summary.eyefiteyefitprint.summary.eyefit summary.eyefiteyefitsummary.eyefit spatialeyefit modelseyefit dynamiceyefit Function to fit an empirical variogram ”by eye” using an interactive Tcl-Tk interface.

```
eyefit(vario, silent = FALSE)
```

An empirical variogram object as returned by the function variogvariog.

logical indicating wheather or not the fitted variogram must be returned. Returns a list of list with the model parameters for each of the saved fit(s). Andreas Kiefer andreas@inf.ufpr.br
Paulo Justiniano Ribeiro Junior paulojus@leg.ufpr.br. variofitvariofit for least squares variogram fit, likfitlikfit for likelihood based parameter estimation and krige.bayeskrige.bayes to obtain the posterior distribution for the model parameters. gambiaGambia Malaria Datagambia gambia.bordersgambiagambia.borders gambia.mapgambiagambia.map datasetsgambia Malaria prevalence in children recorded at villages in The Gambia, Africa.

```
data(gambia)
```

Two objects are made available:

1. gambia
   A data frame with 2035 observations on the following 8 variables.

x x-coordinate of the village (UTM).

y y-coordinate of the village (UTM).

pos presence (1) or absence (0) of malaria in a blood sample taken from the child.

age age of the child, in days

netuse indicator variable denoting whether (1) or not (0) the child regularly sleeps under a bed-net.

treated indicator variable denoting whether (1) or not (0) the bed-net is treated (coded 0 if netuse=0).

green satellite-derived measure of the green-ness of vegetation in the immediate vicinity of the village (arbitrary units).

phc indicator variable denoting the presence (1) or absence (0) of a health center in the village.

2. gambia.borders
   A data frame with 2 variables:

   x x-coordinate of the country borders.

   y y-coordinate of the country borders.

Thomson, M., Connor, S., D Alessandro, U., Rowlingson, B., Diggle, P., Cresswell, M. & Greenwood, B. (1999). Predicting malaria infection in Gambian children from satellite data and bednet use surveys: the importance of spatial correlation in the interpretation of results. *American Journal of Tropical Medicine and Hygiene* 61: 2–8.

Diggle, P., Moyeed, R., Rowlingson, B. & Thomson, M. (2002). Childhood malaria in The Gambia: a case-study in model-based geostatistics, *Applied Statistics*. plot(gambia.borders, type="l", asp=1) points(gambia[,1:2], pch=19) a built-in function for a zoomed map gambia.map() Building a "village-level" data frame ind ¡- paste("x",gambia[,1], "y", gambia[,2], sep="") village ¡- gambia[!duplicated(ind),c(1:2,7:8)] village$prev < -as.vector(tapply(gambia$pos, ind, mean)) plot(village$green, village$prev) geoR-defunctDefunct Functions in the Package geoRgeoR.Rdash.defunct .proflik.ftaugeoR-defunct.proflik.ftau .proflik.nuggeoR-defunct.proflik.nug .proflik.phigeoR-defunct.proflik.phi distdiaggeoR-defunctdistdiag geoRdefunctgeoR-defunctgeoRdefunct likfit.nospatialgeoR-defunctlikfit.nospatial likfit.oldgeoR-defunctlikfit.old loglik.spatialgeoR-defunctloglik.spatial olsfitgeoR-defunctolsfit wlsfitgeoR-defunctwlsfit spatialgeoR-defunct utilitiesgeoR-defunct The functions listed here are no longer part of the package geoR as they are not needed (any more).

`geoRdefunct()`

The following functions are now defunct:

1. olsfitfunctionality incorporated by variofitvariofit. From geoR_1.0-6.

2. wlsfitfunctionality incorporated by variofitvariofit. From geoR_1.0-6.

3. likfit.oldfunctionality incorporated by likfitlikfit. From geoR_1.0-6. The associated functions were also made defunct:
   likfit.nospatial, loglik.spatial, proflik.nug, proflik.phi, proflik.ftau.

4. distdiagfunctionally is redundant with distdist.

variofitvariofit globalvarComputes global variance globalvar spatialglobalvar Global variance computation for a set of locations using the covarianve model

```
globalvar(geodata, locations, coords = geodata$coords, krige)
```

an object of the class geodata

n by 2 matrix with a set of locations, typically a prediction grid

data coordinates

a list defining the model components and the type of kriging. It can take an output to a call to krige.control or a list with elements as for the arguments in krige.control. An scalar with the value of the global variance
Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Isaaks, E.S and Srivastava, R.M. (1989) An Introduction to Applied Geostatistics, pag. 508, eq. 20.7. Oxford University Press. krige.convkrige.conv for the kriging algorithm. grfSimulation of Gaussian Random Fieldsgrf .grf.aux1grf.grf.aux1 geoR2RFgrfgeoR2RF grfclassgrfgrfclass lines.grfgrflines.grf spatialgrf datagengrf grf() generates (unconditional) simulations of Gaussian random fields for given covariance parameters. geoR2RF converts model specification used by geoR to the correponding one in RandomFields.

```
grf(n, grid = "irreg", nx, ny, xlims = c(0, 1), ylims = c(0, 1),
    borders, nsim = 1, cov.model = "matern",
    cov.pars = stop("missing covariance parameters sigmasq and phi"),
    kappa = 0.5, nugget = 0, lambda = 1, aniso.pars,
    mean = 0, method, RF=TRUE, messages)

geoR2RF(cov.model, cov.pars, nugget = 0, kappa, aniso.pars)
```

number of points (spatial locations) in each simulations.

optional. An n $\times 2 matrixwithcoordinatesofthesimulateddata.optional.NumberofpointsintheXdirection.$

optional. Number of points in the Y direction.

optional. Limits of the area in the X direction. Defaults to [0,1].

optional. Limits of the area in the Y direction. Defaults to [0,1].

optional. Typically a two coluns matrix especifying a polygon. See DETAILS below.

Number of simulations. Defaults to 1.

correlation function. See cov.spatialcov.spatial for further details. Defaults to the *exponential* model.

a vector with 2 elements or an n $\times 2 matrixwithvaluesofthecovarianceparameters\sigma^2 (partialsill) and\phi (rangeparam$
$"matern","powered.exponential","cauchy" and" gneiting.matern".Moredetailsonthedocumentationforthefunct$

the value of the nugget effect parameter $\tau^2.valueoftheBox - Coxtransformationparameter.Thevalue\lambda = 1correspondstonotransformation, thedefault.Foranyot$

geometric anisotropy parameters. By default an isotropic field is assumed and this argument is ignored. If a vector with 2 values is provided, with values for the anisotropy angle $\psi_A(inradians)andanisotropyratio\psi_A, thecoordinatesaretransformed, thesimulationisperformedont$ $transformedsuchthattheresultingfieldisanisotropic.Coordinatestransformationisperformedbythe$

simulation method with options for "cholesky", "svd", "eigen", "RF". Defaults to the *Cholesky* decomposition. See section DETAILS below.

logical, with defaults to TRUE, indicating whether the algorithm should try to use the function GaussRFGaussRF from the package RandomFields in case of method is missing and the number of points is greater than 500.

logical, indicating whether or not status messages are printed on the screen (or output device) while the function is running. Defaults to TRUE. For the methods "cholesky", "svd" and "eigen" the simulation consists of multiplying a vector of standardized normal deviates by a square root of the covariance matrix. The square root of a matrix is not uniquely defined. These three methods differs in the way they compute the square root of the (positive definite) covariance matrix.

The previously available method method = "circular.embedding" is no longer available in geoR. For simulations in a fine grid and/or with a large number of points use the package RandomFields.

The option "RF" calls internally the function GaussRFGaussRF from the package RandomFields.

The argument borders, if provides takes a polygon data set following argument poly for the splancs' function csrcsr, in case of grid="reg" or gridpts-gridpts, in case of grid="irreg". For the latter the simulation will have *approximately* "n" points.

grf returns a list with the components:

an n $\times 2matrixwiththecoordinatesofthesimulateddata.avector(ifnsim = 1)oramatrixwiththesimula$

a string with the name of the correlation function.

the value of the nugget parameter.

a vector with the values of $\sigma^2 and\phi, respectively.valueoftheparameter\kappa.$

value of the Box-Cox transformation parameter $\lambda.avectorwithvaluesoftheanisotropyparameters, ifprovidedinthefunctioncall.$

a string with the name of the simulation method used.

a string "1d" or "2d" indicating the spatial dimension of the simulation.

the random seed by the time the function was called.

messages produced by the function describing the simulation.

the function call.

geoR2grf returns a list with the components:

RandomFields name of the correlation model

RandomFields parameter vector Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Wood, A.T.A. and Chan, G. (1994) Simulation of stationary Gaussian process in [0,1]^d.*Journal of Computatinal and Graphical Statistics*, $3, 409 - -432$.

Schlather, M. (1999) *Introduction to positive definite functions and to unconditional simulation of random fields.* Tech. Report ST–99–10, Dept Maths and Stats, Lancaster University.

Schlather, M. RandomFields: *Simulation and Analysis of Random Fields.* R package. http://www.cu.lu/ schlathe.

Schlather, M. (2001) *Simulation and Analysis of Random Fields.* R-News 1 (2), p. 18-20.

Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. plot.grfplot.grf and image.grfimage.grf for graphical output, coords.anisocoords.aniso for anisotropy coordinates transformation and, cholchol, svdsvd and eigeneigen for methods of matrix decomposition and GaussRFGaussRF function in the package RandomFields. sim1 ¡- grf(100, cov.pars = c(1, .25)) a display of simulated locations and values points(sim1) empirical and theoretical variograms plot(sim1) alternative way plot(variog(sim1, max.dist=1)) lines.variomodel(sim1) a "smallish" simulation sim2 ¡- grf(441, grid = "reg", cov.pars = c(1, .25)) image(sim2) 1-D simulations using the same seed and different noise/signal ratios set.seed(234) sim11 ¡- grf(100, ny=1, cov.pars=c(1, 0.25), nug=0) set.seed(234) sim12 ¡- grf(100, ny=1, cov.pars=c(0.75, 0.25), nug=0.25) set.seed(234) sim13 ¡- grf(100, ny=1, cov.pars=c(0.5, 0.25), nug=0.5) par.ori ¡- par(no.readonly = TRUE) par(mfrow=c(3,1), mar=c(3,3,.5,.5)) yl ¡- range(c(sim11$data, sim$12data, sim13$data))image(sim11, type = "l", ylim = yl)image(sim12, type = "l", ylim = yl)image(sim13, type = "l", ylim = yl)par(par.ori)$

simulating within borders data(parana) pr1 ¡- grf(100, cov.pars=c(200, 40), borders=parana$borders, mean = 500)points(pr1)pr1 < -grf(100, grid = "reg", cov.pars = c(200, 40), borders = parana$borders) points(pr1) pr1 ¡- grf(100, grid="reg", nx=10, ny=5, cov.pars=c(200, 40), borders=parana$borders)points(pr1)$ headHead observations in a regional confined aquiferhead datasetshead Measurements of potentiometric head at 29 locations in a regional confined sandstone aquifer. Extract from Kitanidis' book.

```
data(head)
```

An object of the class geodata which is a list with the elements:

coords coordinates of the data location.

data the data vector with head measurements (feet).

Kitanidis, P.K. Introduction to geostatistics - applications in hidrology (1997). Cambridge University Press. summary(head) plot(head) hist.krige.bayesPlots Sample from Posterior Distributionshist.krige.bayes spatialhist.krige.bayes dplothist.krige.bayes Plots histograms and/or density estimation with samples from the posterior distribution of the model parameters

```
## S3 method for class 'krige.bayes'
hist(x, pars, density.est = TRUE, histogram = TRUE, ...)
```

an object of the class krige.bayes, with an output of the funtions krige.bayeskrige.bayes.

a vector with the names of one or more of the model parameters. Defaults to all model parameters. Setting to -1 excludes the intercept.

logical indication whether a line with the density estimation should be added to the plot.

logical indicating whether the histogram is included in the plot.

further arguments for the plotting functions and or for the density estimation. Produces a plot in the currently graphics device.
Returns a invisibleinvisible list with the components:

with the output of the function histhist for each parameter

with the output of the function densitydensity for each parameter
Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. krige.bayeskrige.bayes, histhist, densitydensity. See documentation for krige.bayes() hoefData for spatial analysis of experimentshoef datasetshoef The hoef data frame has 25 rows and 5 columns.
The data consists of a uniformity trial for which *artificial* treatment effects were assign to the plots.

```
data(hoef)
```

This data frame contains the following columns:

x1 x-coordinate of the plot.

x2 y-coordinate of the plot.

dat the *artificial* data.

trat the treatment number.

ut the data from the uniformity trial, without the treatment effect.

The treatment effects assign to the plots are:

- Treatment 1: $\tau_1 = 0 Treatment 2 : \tau_2 = -3$
- Treatment 3: $\tau_3 = -5 Treatment 4 : \tau_4 = 6$
- Treatment 5: $\tau_5 = 6$

Ver Hoef, J.M. & Cressie, N. (1993) Spatial statistics: analysis of field experiments. In Scheiner S.M. and Gurevitch, J. (Eds) *Design and Analysis of Ecological Experiments.* Chapman and Hall. hoef.geo ¡- as.geodata(hoef, covar.col=4) summary(hoef) summary(hoef.geo) points(hoef.geo, cex.min=2, cex.max=2, pt.div="quintiles") image.grfImage, Contour or Perspective Plot of Simulated Gaussian Random Fieldimage.grf .geoR_fullGridimage.grf.geoR.Rul.fullGrid contour.grfimage.grfcontour.grf persp.grfimage.grfpersp.grf spatialimage.grf dplotimage.grf Methods for image, contour or perspective plot of a realisation of a Gaussian random field, simulated using the function grfgrf.

```
## S3 method for class 'grf'
image(x, sim.number = 1, borders, x.leg, y.leg, ...)
## S3 method for class 'grf'
contour(x, sim.number = 1, borders, filled = FALSE, ...)
## S3 method for class 'grf'
persp(x, sim.number = 1, borders, ...)
```

an object of the class grf, typically an output of the function grfgrf.

simulation number. Indicates the number of the simulation top be plotted. Only valid if the object contains more than one simulation. Defaults to 1.

optional. Typically a two coluns matrix especifying a polygon. Points outside the borders will be set no NA

limits for the legend in the horizontal and vertical directions.

logical. If FALSE the function contourcontour is used otherwise filled.contourfilled.contour. Defaults to FALSE.

further arguments to be passed to the functions imageimage, contourcontour or persppersp. An image or perspective plot is produced on the current graphics device. No values are returned. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information about the package geoR can be found at:
http://www.leg.ufpr.br/geoR. grfgrf for simulation of Gaussian random fields, imageimage and persppersp for the generic plotting functions. generating 4 simulations of a Gaussian random field sim ¡- grf(441, grid="reg", cov.pars=c(1, .25), nsim=4) op ¡- par(no.readonly = TRUE) par(mfrow=c(2,2), mar=c(3,3,1,1), mgp = c(2,1,0)) for (i in 1:4) image(sim, sim.n=i) par(op) image.krige.bayesPlots Results of the Predictive Distributionimage.krige.bayes .prepare.graph.krige.bayesimage.krige.bayes.prepare.graph.krige.bayes contour.krige.bayesimage.krige.bayescontour.krige.bayes persp.krige.bayesimage.krige.bayespersp.krige.bayes spatialimage.krige.bayes This function produces an image or perspective plot of a selected element of the predictive distribution returned by the function krige.bayeskrige.bayes.

```
## S3 method for class 'krige.bayes'
image(x, locations, borders,
                  values.to.plot=c("mean", "variance",
                            "mean.simulations", "variance.simulations",
                            "quantiles", "probabilities", "simulation"),
                  number.col, coords.data, x.leg, y.leg, messages, ...)
## S3 method for class 'krige.bayes'
contour(x, locations, borders,
                  values.to.plot = c("mean", "variance",
                       "mean.simulations", "variance.simulations",
                       "quantiles", "probabilities", "simulation"),
                  filled=FALSE, number.col, coords.data,
                  x.leg, y.leg, messages, ...)
## S3 method for class 'krige.bayes'
persp(x, locations, borders,
                  values.to.plot=c("mean", "variance",
                         "mean.simulations", "variance.simulations",
                         "quantiles", "probabilities", "simulation"),
                    number.col, messages, ...)
```

an object of the class krige.bayes, typically an output of the function krige.bayeskrige.bayes.

an n $\times 2matrixwiththecoordinatesofthepredictionlocations, whichshoulddefinearegulargridinordert$

select the element of the predictive distribution to be plotted. See DETAILS below.

logical. If FALSE the function contourcontour is used otherwise filled.contourfilled.contour. Defaults to FALSE.

Specifies the number of the column to be plotted. Only used if previous argument is set to one of ”quantiles”, ”probabilities” or ”simulation”.

optional. If an n $\times 2matrixwiththedatacoordinatesisprovided, pointsindicatingthedatalocationsareinc$

logical, if TRUE status messages are printed while running the function.

extra arguments to be passed to the plotting function imageimage or persppersp. The function krige.bayeskrige.bayes returns summaries and other results about the predictive distributions. The argument values.to.plot specifies which result will be plotted. It can be passed to the function in two different forms:

- a vector with the object containing the values to be plotted, or
- one of the following options: ”moments.mean”, ”moments.variance”, ”mean.simulations”, ”variance.simulations”, ”quantiles”, ”probability” or ”simulation”.

For the last three options, if the results are stored in matrices, a column number must be provided using the argument number.col.

The documentation for the function krige.bayeskrige.bayes provides further details about these options. An imageimage or persppersp plot is produced on the current graphics device. No values are returned. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. krige.bayeskrige.bayes for Bayesian Kriging computations and, imageimage and persppersp for the generic plotting functions. See examples in the documentation for the function krige.bayes(). image.krigingImage or Perspective Plot with Kriging Resultsimage.kriging .prepare.graph.krigingimage.kriging.prepare.graph.kriging contour.krigingimage.krigingcontour.kriging persp.krigingimage.krigingpersp.kriging plot.1dimage.krigingplot.1d spatialimage.kriging dplotimage.kriging Plots image or perspective plots with results of the kriging calculations.

```
## S3 method for class ’kriging’
image(x, locations, borders, values = x$predict,
               coords.data, x.leg, y.leg, ...)
## S3 method for class ’kriging’
contour(x, locations, borders, values = x$predict,
              coords.data, filled=FALSE, ...)
## S3 method for class 'kriging'
persp(x, locations, borders, values = x$predict, ...)
```

an object of the class kriging, typically with the output of the functions krige.convkrige.conv or kslineksline.

an n $\times 2matrixwiththecoordinatesofthepredictionlocations, whichshoulddefinearegulargridinordertobeplottedb$

a vector with values to be plotted. Defaults to obj$predict.

optional. If an n $\times 2matrixwiththedatacoordinatesisprovided, pointsindicatingthedatalocationsareincludedinthe$

logical. If FALSE the function contourcontour is used otherwise filled.contourfilled.contour. Defaults to FALSE.

further arguments to be passed to the functions imageimage, contourcontour, filled.contourfilled.contour, persppersp or legend.krigelegend.krige. For instance, the argument zlim can be used to set the the minimum and maximum 'z' values for which colors should be plotted. See documentation for those function for possible arguments.
plot1d and prepare.graph.kriging are auxiliary functions called by the others. An image or perspective plot is produced o the current graphics device. No values are returned. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. krige.convkrige.conv and kslineksline for kriging calculations. Documentation for imageimage, contourcontour, filled.contourfilled.contour and persppersp contain basic information on the plotting functions. loci ¡- expand.grid(seq(0,1,l=51), seq(0,1,l=51)) kc ¡- krige.conv(s100, loc=loci, krige=krige.control(cov.pars=c(1, .25))) image(kc) contour(kc) image(kc) contour(kc, add=TRUE, nlev=21) persp(kc, theta=20, phi=20) contour(kc, filled=TRUE) contour(kc, filled=TRUE, color=terrain.colors) contour(kc, filled=TRUE, col=gray(seq(1,0,l=21))) adding data locations image(kc, coords.data=s100$coords)contour(kc, filled = TRUE, coords.data = s$100coords,color=terrain.colors) now dealing with borders bor ¡- matrix(c(.4,.1,.3,.9,.9,.7,.9,.7,.3,.2,.5,.8), ncol=2) plotting just inside borders image(kc, borders=bor) contour(kc, borders=bor) image(kc, borders=bor) contour(kc, borders=bor, add=TRUE) contour(kc, borders=bor, filled=TRUE, color=terrain.colors) kriging just inside borders kc1 ¡- krige.conv(s100, loc=loci, krige=krige.control(cov.pars=c(1, .25)), borders=bor) image(kc1) contour(kc1) avoiding the borders image(kc1, borders=NULL) contour(kc1, borders=NULL)

op ¡- par(no.readonly = TRUE) par(mfrow=c(1,2), mar=c(3,3,0,0), mgp=c(1.5, .8,0)) image(kc) image(kc, val=sqrt(kc$krige.var$))

different ways to add the legends and pass arguments: image(kc, ylim=c(-0.2, 1), x.leg=c(0,1), y.leg=c(-0.2, -0.1)) image(kc, val=kc$krige.var, ylim = c(-0.2,1))legend.krige(y.leg = c(-0.2,-0.1), x.leg = c(0,1), val = sqrt(kc$krige.var))

image(kc, ylim=c(-0.2, 1), x.leg=c(0,1), y.leg=c(-0.2, -0.1), cex=1.5) image(kc, ylim=c(-0.2, 1), x.leg=c(0,1), y.leg=c(-0.2, -0.1), offset.leg=0.5) image(kc, xlim=c(0, 1.2)) legend.krige(x.leg=c(1.05,1.1), y.leg=c(0,1), kc$pred, vert = TRUE)image(kc, xlim = c(0,1.2))legend.krige(x.leg = c(1.05,1.1), y.leg = c(0,1), kc$pred, vert=TRUE, off=1.5, cex=1.5)

par(op) InvChisquareThe (Scaled) Inverse Chi-Squared Distribution-InvChisquare dinvchisqInvChisquaredinvchisq rinvchisqInvChisquarerinvchisq distributionInvChisquare Density and random generation for the scaled inverse chi-squared $(\chi^2_{ScI})distributionwithdfdegreesoffreedomandoptionalnon-centralityparameterscale.$

```
dinvchisq(x, df, scale, log = FALSE)
rinvchisq(n, df, scale = 1/df)
```

vector of quantiles.

number of observations. If length(n) ¿ 1, the length is taken to be the number required.

degrees of freedom.

scale parameter.

logical; if TRUE, densities d are given as log(d). The inverse chi-squared distribution with df= n degrees of freedom has density f(x) = $1\frac{}{2^{n/2}\Gamma(n/2)(1/x)^{n/2+1}e^{-1/(2x)}forx>0.Themeanandvarianceare\frac{1}{(n-2)}and\frac{2}{(n-4)(n-2)^2}}.$

The non-central chi-squared distribution with df= n degrees of freedom and non-centrality parameter scale = S$^2 hasdensityf(x) = \frac{n/2^{n/2}}{\Gamma(n/2)}S^n(1/x)^{n/2+1}e^{-(nS^2)/(2x)}, forx \geq 0.Thefirstisaparticularcaseofthelatter$ dinvchisq gives the density and rinvchisq generates random deviates. rchisqrchisq for the chi-squared distribution which is the basis for this function. set.seed(1234); rinvchisq(5, df=2) set.seed(1234); 1/rchisq(5, df=2)

set.seed(1234); rinvchisq(5, df=2, scale=5) set.seed(1234); 5*2/rchisq(5, df=2)

inverse Chi-squared is a particular case x ¡- 1:10 all.equal(dinvchisq(x, df=2), dinvchisq(x, df=2, scale=1/2)) isaaksData from Isaaks and Srisvastava's bookisaaks datasetsisaaks Toy example used in the book *An Introduction to Geostatistics* to illustrate the effects of different models and parameters in the kriging results when predicting at a given point.

```
data(isaaks)
```

An object of the class geodata which is a list with the elements:

coords coordinates of the data location.

data the data vector.

x0 coordinate of the prediction point.

Isaaks, E.H. & Srisvastava, R.M. (1989) An Introduction to Applied Geostatistics. Oxford University Press. isaaks summary(isaaks) plot(isaaks$coords, asp = 1, type = "n")text(isaaks$coords, as.character(isaaks$data))points(isaaks$x0, pch="?", cex=2, col=2) jitterDupCoordsJitters (duplicated) coordinates.jitterDupCoords jitter2djitterDupCoordsjitter2d jitterDupCoords.defaultjitterDupCoordsjitterDupCoords.default jitterDupCoords.geodatajitterDupCoordsjitterDupCoords.geodata spatialjitterDupCoords manipjitterDupCoords Jitters 2D coordinates uniformily on a region around (duplicated) points.

```
jitter2d(coords, max, min = 0.2 * max, fix.one = TRUE,
         which.fix = c("random", "first", "last"))

jitterDupCoords(x, ...)

## Default S3 method:
jitterDupCoords(x, ...)

## S3 method for class 'geodata'
jitterDupCoords(x, ...)
```

a matrix or data frame with 2D coordinates or geodata object.

numeric scalar defining maximum jittering distance.

numeric scalar defining minimum jittering distance.

logical. Whether or not one of the coordinates should not be jittered.

single element vector of integer or character, defining which coordinate won't be jittered. Only used if fix.one=TRUE.

arguments passed to jitter2d. jitter2d returns an object of the same type fo the input with jittered values

jitterDupCoords returns an object of the same type fo the input with jittered coordinate values only at the duplicated locations
Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. dup.coordsdup.coords, duplicated.geodataduplicated.geodata for functions identifying duplicated locations. simulating data dt ¡- grf(30, cov.p=c(1, .3)) dt$coords <- round(dt$coords, dig=2) "forcing" some duplicated locations dt$coords[4,] <- dt$coords[14,] ¡- dt$coords[24,] <- dt$coords[2,] dt$coords[17,] <- dt$coords[23,] ¡- dt$coords[8,]$

jittering a matrix of duplicated coordinates dt$coords[c(2,4,14,24),] jitter2d(dt$coords[c(2,4,14,24),], max=0.01)

jittering only the duplicated locations and comparing with original cbind(dt$coords, jitterDupCoords(dt$coords, max=0.01))

creating a now geodata object jittering the duplicated locations of the original one: dup.coords(dt) dt1 ¡- jitterDupCoords(dt, max=0.01) dup.coords(dt1) kattegatKattegat basin salinity datakattegat datasetskattegat Salinity measurements at the Kattegat basin, Denmark.

```
data(kattegat)
```

An object of the classclass "geodata", which is list with three components:

coords the coordinates of the data locations. The distance are given in kilometers.

data values of the piezometric head. The unit is heads to meters.

dk a list with cooordinates of lines defining borders and islands across the study area.

National Environmental Research Institute, Arhus University, Denmark and the Swedish Meteorological and Hydrological Institute. Diggle, P. J. and Lophaven, S. (2006). Bayesian geostatistical design. *Scandinavian Journal of Statistics*, 33: 55-64. plot(c(550,770),c(6150,6420),type="n",xlab="X UTM",ylab="Y UTM") points(kattegat, add=TRUE) lapply(kattegat$dk, lines, lwd =$ 2) krige.bayesBayesian Analysis for Gaussian Geostatistical Modelskrige.bayes model.controlkrige.bayesmodel.control post2priorkrige.bayespost2prior print.krige.bayeskrige.bayesprint.krige.bayes print.posterior.krige.bayeskrige.bayesprint.posterior.krige.bayes prior.controlkrige.bayesprior.control spatialkrige.bayes modelskrige.bayes The function krige.bayes performs Bayesian analysis of geostatistical data allowing specifications of different levels of uncertainty in the model parameters. It returns results on the posterior distributions for the model parameters and on the predictive distributions for prediction locations (if provided).

```
krige.bayes(geodata, coords = geodata$coords, data = geodata$data,
            locations = "no", borders, model, prior, output)

model.control(trend.d = "cte", trend.l = "cte", cov.model = "matern",
              kappa = 0.5, aniso.pars = NULL, lambda = 1)

prior.control(beta.prior = c("flat", "normal", "fixed"),
              beta = NULL, beta.var.std = NULL,
              sigmasq.prior = c("reciprocal", "uniform",
                                "sc.inv.chisq", "fixed"),
              sigmasq = NULL, df.sigmasq = NULL,
              phi.prior = c("uniform", "exponential","fixed",
                            "squared.reciprocal", "reciprocal"),
              phi = NULL, phi.discrete = NULL,
              tausq.rel.prior = c("fixed", "uniform", "reciprocal"),
              tausq.rel, tausq.rel.discrete = NULL)

post2prior(obj)
```

a list containing elements coords and data as described next. Typically an object of the class "geodata" - a geoR data-set. If not provided the arguments coords and data must be provided instead.

an n $\times 2 matrixwhereeachrowhasthe 2-Dcoordinatesofthendatalocations.Bydefaultittakesthecompo$

a vector with n data values. By default it takes the component data of the argument geodata, if provided.

an N $\times 2 matrixordata - framewiththe 2 - DcoordinatesoftheNpredictionlocations, oralistforwhichthefirsttwocomponentsareused.Inputisint$

a list defining the fixed components of the model. It can take an output to a call to model.control or a list with elements as for the arguments in model.control. Default values are assumed for arguments not provided. See section DETAILS below.

a list with the specification of priors for the model parameters. It can take an output to a call to prior.control or a list with elements as for the arguments in prior.control. Default values are assumed for arguments not provided. See section DETAILS below.

a list specifying output options. It can take an output to a call to output.control or a list with elements as for the arguments in output.control. Default values are assumed for arguments not provided. See documentation for output.controloutput.control for further details.

specifies the trend (covariates) values at the data locations. See documentation of trend.spatialtrend.spatial for further details. Defaults to "cte".

specifies the trend (covariates) at the prediction locations. Must be of the same type as defined for trend.d. Only used if prediction locations are provided in the argument locations.

string indicating the name of the model for the correlation function. Further details in the documentation for cov.spatialcov.spatial.

additional smoothness parameter. Only used if the correlation function is one of: "matern", "powered.exponential", "cauchy" or "gneiting.matern". In the current implementation this parameter is always regarded as fixed during the Bayesian analysis.

fixed parameters for geometric anisotropy correction. If aniso.pars = FALSE no correction is made, otherwise a two elements vector with values for the anisotropy parameters must be provided. Anisotropy correction consists of a transformation of the data and prediction coordinates performed by the function coords.anisocoords.aniso.

numerical value of the Box-Cox transformation parameter. The value $\lambda = 1 correspondstonotransformation.TheBox - Coxparameter \lambda isalwaysregardedasfixedanddatatransformationisperformedbeforetheanalysis.Predictionres$ $transformedandreturnedisthesamescaleasfortheoriginaldata.For \lambda = 0 thelog-$ $transformationisperformed.If \lambda < 0 themeanpredictordoesn'tmakesense(theresultingdistributionhasnoexpecta$

prior distribution for the mean (vector) parameter $\beta.Theoptionsare"flat"(default),"normal"or"fixed"(knownmean).$

mean hyperparameter for the distribution of the mean (vector) parameter $\beta.Onlyusedifbeta.prior = "normal"orbeta.prior = "fixed".Forthelaterbetadefinesthevalueoftheknownmean.$

standardised (co)variance hyperparameter(s) for the prior for the mean (vector) parameter $\beta.The(co)variancematrixfor \beta isgivenbythemultiplicationofthismatrixby \sigma^2.Onlyusedifbeta.prior = "normal".$

specifies the prior for the parameter $\sigma^2.If"reciprocal"(thedefault),theprior\frac{1}{\sigma^2}isused.Otherwisetheparameterisre$

fixed value of the sill parameter $\sigma^2.Onlyusedifsigmasq.prior = FALSE.numerical.Numberofdegreesoffreedom$

prior distribution for the range parameter $\phi.Optionsare :$ $"uniform","exponential","reciprocal","squared.reciprocal"and"fixed".Alternativelly,auserdefineddiscreted$ $If"fixed"theargument \phi mustbeprovidedandisregardedasfixedwhenperformingpredictions.$ $Fortheexponentialpriortheargumentphimustprovidethevalueforofhyperparameter \nu whichcorrespondstotheexp$

fixed value of the range parameter $\phi.Onlyneededifphi.prior = "fixed"orifphi.prior = "exponential".$

support points of the discrete prior for the range parameter $\phi.Thedefaultisasequenceof 51 valuesbetween 0 and 2 timesthemaximumdistancebetweenthedatalocations.$

specifies a prior distribution for the relative nugget parameter $\tau^2 \frac{}{\sigma^2 . If tausq.rel.prior = "fixed" the relative nugget is considered known (fixed) with value given by the argument tausq.rel. If tau}$
fixed value for the relative nugget parameter. Only used if tausq.rel.prior = "fixed".

support points of the discrete prior for the relative nugget parameter $\tau^2 \frac{}{\sigma^2 .}$

an object of the class krige.bayes or posterior.krige.bayes with the output of a call to krige.bayes. The function post2prior takes the posterior distribution computed by one call to krige.bayes and prepares it to be used a a prior in a subsequent call. Notice that in this case the function post2prior is used instead of prior.control. krige.bayes is a generic function for Bayesian geostatistical analysis of (transformed) Gaussian where predictions take into account the parameter uncertainty.

It can be set to run conventional kriging methods which use known parameters or *plug-in* estimates. However, the functions krige.conv and ksline are preferable for prediction with fixed parameters.

PRIOR SPECIFICATION

The basis of the Bayesian algorithm is the discretisation of the prior distribution for the parameters $\phi and \tau^2_{rel} = \frac{\tau^2}{\sigma^2} . The Tech. Report (see References below) provides details on the results used in the current in$
$The expressions of the implemented priors for the parameter \phi are :$

p(ϕ) $\propto 1 . p(\phi) = \frac{1}{\nu} \exp(-\frac{1}{\nu} * \phi)$.

p(ϕ) $\propto \frac{1}{\phi} . p(\phi) \propto \frac{1}{\phi^2}$.

fixed known or estimated value of ϕ.

The expressions of the implemented priors for the parameter $\tau^2_{rel} are :$

fixed known or estimated value of $\tau^2_{rel} . Defaults to zero . p(\tau^2_{rel}) \propto 1$.

p(τ^2_{rel}) $\propto \frac{1}{\tau^2_{rel}}$.

Apart from those a *user defined* prior can be specifyed by entering a vector of probabilities for a discrete distribution with suport points given by the argument phi.discrete and/or tausq.rel.discrete.

CONTROL FUNCTIONS

The function call includes auxiliary control functions which allows the user to specify and/or change the specification of model components (using model.control), prior distributions (using prior.control) and output options (using output.control). Default options are available in most of the cases.

An object with classclass "krige.bayes" and "kriging". The attribute prediction.locations containing the name of the object with the coordinates of the prediction locations (argument locations) is assigned to the object. Returns a list with the following components:

results on on the posterior distribution of the model parameters. A list with the following possible components:

- beta summary information on the posterior distribution of the mean parameter $\beta . sigmasq summary information on the posterior distribution of the variance parameter \sigma^2 (partial$

- phi summary information on the posterior distribution of the correlation parameter $\phi(rangeparameter).tausq.relsummaryinformationontheposteriordistributionoftherelativenuggetvarianc$
- joint.phi.tausq.relinformation on discrete the joint distribution of these parameters.
- sample a data.frame with a sample from the posterior distribution. Each column corresponds to one of the basic model parameters.

results on the predictive distribution at the prediction locations, if provided. A list with the following possible components:

- mean expected values.
- variance expected variance.
- distribution type of posterior distribution.
- mean.simulations mean of the simulations at each locations.
- variance.simulations variance of the simulations at each locations.
- quantiles.simulations quantiles computed from the the simulations.
- probabilities.simulations probabilities computed from the simulations.
- simulations simulations from the predictive distribution.

a list with information on the prior distribution and hyper-parameters of the model parameters $(\beta, \sigma^2, \phi, \tau^2_{rel})$.

model specification as defined by model.control.

system random seed before running the function. Allows reproduction of results. If the .Random.seed.Random.seed is set to this value and the function is run again, it will produce exactly the same results.

maximum distance found between two data locations.

the function call.

Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Diggle, P.J. & Ribeiro Jr, P.J. (2002) Bayesian inference in Gaussian model-based geostatistics. Geographical and Environmental Modelling, Vol. 6, No. 2, 129-146.

The technical details about the implementation of krige.bayes can be found at:
Ribeiro, P.J. Jr. and Diggle, P.J. (1999) *Bayesian inference in Gaussian model-based geostatistics*. Tech. Report ST-99-08, Dept Maths and Stats, Lancaster University.
Available at: http://www.leg.ufpr.br/geoR/geoRdoc/bayeskrige.pdf

Further information about geoR can be found at:
http://www.leg.ufpr.br/geoR.

For a extended list of examples of the usage see http://www.leg.ufpr.br/geoR/tutorials/examples.krige.bayes.R and/or the geoR tutorials page at http://www.leg.ufpr.br/geoR/tutorials.

lines.variomodel.krige.bayeslines.variomodel.krige.bayes, plot.krige.bayesplot.krige.bayes for outputs related to the parameters in the model, image.krige.bayesimage.krige.bayes and persp.krige.bayespersp.krige.bayes for graphical output of prediction results. krige.convkrige.conv and kslineksline for conventional kriging methods. Not run: generating a simulated data-set ex.data ¡- grf(70, cov.pars=c(10, .15), cov.model=”matern”, kappa=2) defining the grid of prediction locations: ex.grid ¡- as.matrix(expand.grid(seq(0,1,l=21), seq(0,1,l=21))) computing posterior and predictive distributions (warning: the next command can be time demanding) ex.bayes ¡- krige.bayes(ex.data, loc=ex.grid, model = model.control(cov.m=”matern”, kappa=2), prior = prior.control(phi.discrete=seq(0, 0.7, l=51), phi.prior=”reciprocal”)) Prior and posterior for the parameter phi plot(ex.bayes, type=”h”, tausq.rel = FALSE, col=c(”red”, ”blue”)) Plot histograms with samples from the posterior par(mfrow=c(3,1)) hist(ex.bayes) par(mfrow=c(1,1))

Plotting empirical variograms and some Bayesian estimates: Empirical variogram plot(variog(ex.data, max.dist = 1), ylim=c(0, 15)) Since ex.data is a simulated data we can plot the line with the ”true” model lines.variomodel(ex.data, lwd=2) adding lines with summaries of the posterior of the binned variogram lines(ex.bayes, summ = mean, lwd=1, lty=2) lines(ex.bayes, summ = median, lwd=2, lty=2) adding line with summary of the posterior of the parameters lines(ex.bayes, summary = ”mode”, post = ”parameters”)

Plotting again the empirical variogram plot(variog(ex.data, max.dist=1), ylim=c(0, 15)) and adding lines with median and quantiles estimates my.summary ¡- function(x)quantile(x, prob = c(0.05, 0.5, 0.95)) lines(ex.bayes, summ = my.summary, ty=”l”, lty=c(2,1,2), col=1)

Plotting some prediction results op ¡- par(no.readonly = TRUE) par(mfrow=c(2,2), mar=c(4,4,2.5,0.5), mgp = c(2,1,0)) image(ex.bayes, main=”predicted values”) image(ex.bayes, val=”variance”, main=”prediction variance”) image(ex.bayes, val= ”simulation”, number.col=1, main=”a simulation from the distribution”) image(ex.bayes, val= ”simulation”, number.col=2, main=”another simulation from predictive distribution”) par(op)

End(Not run) For a extended list of exemples of the usage of krige.bayes() see http://www.leg.ufpr.br/geoR/tutorials/examples.krige.bayes.R

krige.convSpatial Prediction – Conventional Krigingkrige.conv krige.controlkrige.convkrige.control spatialkrige.conv This function performs spatial prediction for fixed covariance parameters using global neighbourhood.

Options available implement the following types of kriging: *SK* (simple kriging), *OK* (ordinary kriging), *KTE* (external trend kriging) and *UK* (universal kriging).

```
krige.conv(geodata, coords=geodata$coords, data=geodata$data,
           locations, borders, krige, output)

krige.control(type.krige = "ok", trend.d = "cte", trend.l = "cte",
            obj.model = NULL, beta, cov.model, cov.pars, kappa,
            nugget, micro.scale = 0, dist.epsilon = 1e-10,
            aniso.pars, lambda)
```

a list containing elements coords and data as described next. Typically an object of the classclass "geodata" - a geoR data-set. If not provided the arguments coords and data must be provided instead.

an n $\times 2 matrixordata - framewiththe2 - Dcoordinatesofthendatalocations.Bydefaultittakesthecomponentcoordsoftheargumentgeodata, ifprovided.ave$

an N $\times 2 matrixordata - framewiththe2 - DcoordinatesoftheNpredictionlocations, oralistforwhichthefirsttwocomponentsareused.Inputisinternallychec$

a list defining the model components and the type of kriging. It can take an output to a call to krige.control or a list with elements as for the arguments in krige.control. Default values are assumed for arguments or list elements not provided. See the description of arguments in 'krige.control' below.

a list specifying output options. It can take an output to a call to output.control or a list with elements as for the arguments in output.control. Default values are assumed for arguments not provided. See documentation for output.controloutput.control for further details.

type of kriging to be performed. Options are "SK", "OK" corresponding to simple or ordinary kriging. Kriging with external trend and universal kriging can be defined setting type.krige = "OK" and specifying the trend model using the arguments trend.d and trend.l.

specifies the trend (covariate) values at the data locations. See documentation of trend.spatialtrend.spatial for further details. Defaults to "cte".

specifies the trend (covariate) values at prediction locations. It must be of the same type as for trend.d. Only used if prediction locations are provided in the argument locations.

a list with the model parameters. Typically an output of likfitlikfit or variofitvariofit.

numerical value of the mean (vector) parameter. Only used if type.krige="SK".

string indicating the name of the model for the correlation function. Further details can be found in the documentation of the function cov.spatialcov.spatial.

a 2 elements vector with values of the covariance parameters $\sigma^2 (partialsill) and \phi (rangeparameter), respectively.$

additional smoothness parameter required by the following correlation functions: "matern", "powered.exponential", "cauchy" and "gneiting.matern".

the value of the nugget variance parameter $\tau^2 .Defaultstozero.micro - scalevariance.Ifdifferentfromzero, thenuggetvarianceisdividedinto2terms :$ *micro-scale variance*and*measurement error*$.Thisaffecttheprecisionofthepredictions.Ofteninpractice, thesetwov$

a numeric value. Locations which are separated by a distance less than this value are considered co-located.

parameters for geometric anisotropy correction. If aniso.pars = FALSE no correction is made, otherwise a two elements vector with values for the anisotropy parameters must be provided. Anisotropy correction consists of a transformation of the data and prediction coordinates performed by the function coords.anisocoords.aniso.

numeric value of the Box-Cox transformation parameter. The value $\lambda = 1 correspondstonotransformationand \lambda = 0 correspondstothelog -$

$transformation.Predictionresultsareback-transformedandreturnedisthesamescaleasfortheorigin$
According to the arguments provided, one of the following different types of kriging: *SK*, *OK*, *UK* or *KTE* is performed. Defaults correspond to ordinary kriging.
An object of the classclass kriging. The attribute prediction.locations containing the name of the object with the coordinates of the prediction locations (argument locations) is assigned to the object. Returns a list with the following components:

a vector with predicted values.

a vector with predicted variances.

estimates of the $\beta, parameterimplicitinkrigingprocedure.Notincludediftype.krige = "SK".anni \times n.$

messages about the type of prediction performed.

the function call. Other results can be included depending on the options passed to output.controloutput.control. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. output.controloutput.control sets output options, image.krigingimage.kriging and persp.krigingpersp.kriging for graphical output of the results, krige.bayeskrige.bayes for Bayesian prediction and kslineksline for a different implementation of kriging allowing for moving neighborhood. For model fitting see likfitlikfit or variofitvariofit. Not run: Defining a prediction grid loci ¡- expand.grid(seq(0,1,l=21), seq(0,1,l=21)) predicting by ordinary kriging kc ¡- krige.conv(s100, loc=loci, krige=krige.control(cov.pars=c(1, .25))) mapping point estimates and variances par.ori ¡- par(no.readonly = TRUE) par(mfrow=c(1,2), mar=c(3.5,3.5,1,0), mgp=c(1.5,.5,0)) image(kc, main="kriging estimates") image(kc, val=sqrt(kc$krige.var), main = "krigingstd.errors")Nowsettingtheoutputtosimulatefromthepredictive(obtainingconditionalsimula$ $-output.control(n.predictive = 1000, quant = 0.9, thres = 2)set.seed(123)kc < -krige.conv(s100, loc = loci, krige = krige.control(cov.pars = c(1, .25)), output = s.out)par(mfrow = c(2, 2))image(kc, val = kc$simul[,1], main="a cond. simul.") image(kc, val=kc$simul[, 1], main = "anothercond.simul.")image(kc, val = (1 - kc$prob), main="Map of P(Y ¿ 2)") image(kc, val=kc$quant, main = "Mapofys.t.P(Y < y) = 0.9")par(par.ori)$
End(Not run) krweightsComputes kriging weightskrweights spatialkrweights Computes the weights assign for each data point in simple and ordinary krigring

```
krweights(coords, locations, krige)
```

matrix with data coordinates

matrix with coordinates of the prediciton points

kriging parameters. See krige.control in krige.conv
A matrix of weights Figure 8.4 in Webster and Oliver (2001), see help(wo) attach(wo) par(mfrow=c(2,2)) plot(c(-10,130), c(-10,130), ty="n", asp=1) points(rbind(coords, x1)) KC1 ¡- krige.control(cov.pars=c(0.382,90.53)) w1 ¡- krweights(wo$coords, loc = x1, krige = KC1)text(coords[, 1], 5 +$

coords[, 2], *round*(*w*1, *dig* = 3))*plot*(*c*(−10, 130), *c*(−10, 130), *ty* = ”*n*”, *asp* = 1)*points*(*rbind*(*coords*, *x*1))*KC*2 < −*krige.control*(*cov.pars* = *c*(0.282, 90.53), *nug* = 0.1)*w*2 < −*krweights*(*wo*coords, loc=x1, krige=KC2) text(coords[,1], 5+coords[,2], round(w2, dig=3)) plot(c(-10,130), c(-10,130), ty=”n”, asp=1) points(rbind(coords, x1)) KC3 ¡- krige.control(cov.pars=c(0.082,90.53), nug=0.3) w3 ¡- krweights(wo*coords*, *loc* = *x*1, *krige* = *KC*3)*text*(*coords*[, 1], 5 + *coords*[, 2], *round*(*w*3, *dig* = 3))*plot*(*c*(−10, 130), *c*(−10, 130), *ty* = ”*n*”, *asp* = 1)*points*(*rbind*(*coords*, *x*1))*KC*4 < −*krige.control*(*cov.pars* = *c*(0, 90.53), *nug* = 0.382, *micro* = 0.382)*w*4 < −*krweights*(*wo*coords, loc=x1, krige=KC4) text(coords[,1], 5+coords[,2], round(w4, dig=3)) SK vs OK plot(c(-10,130), c(-10,130), ty=”n”, asp=1) points(rbind(coords, x1)) KC5 ¡- krige.control(cov.pars=c(0.382,50)) w5 ¡- krweights(wo*coords*, *loc* = *x*1, *krige* = *KC*5)*KC*6 < −*krige.control*(*type* = ”*sk*”, *beta* = 2, *cov.pars* = *c*(0.382, 50))*w*6 < −*krweights*(*wo*coords, loc=x1, krige=KC6) text(coords[,1], 5+coords[,2], round(w5, dig=3)) text(coords[,1], -5+coords[,2], round(w6, dig=3)) plot(c(-10,130), c(-10,130), ty=”n”, asp=1) points(rbind(coords, x1)) KC7 ¡- krige.control(cov.pars=c(0.382,0)) w7 ¡- krweights(wo*coords*, *loc* = *x*1, *krige* = *KC*7)*KC*8 < −*krige.control*(*type* = ”*sk*”, *beta* = 2, *cov.pars* = *c*(0.382, 0))*w*8 < −*krweights*(*wo*coords, loc=x1, krige=KC8) text(coords[,1], 5+coords[,2], round(w7, dig=3)) text(coords[,1], -5+coords[,2], round(w8, dig=3)) KsatSaturated Hydraulic ConductivityKsat datasetsKsat The data consists of 32 measurements of the saturated hydraulic conductivity of a soil.

```
data(Ksat)
```

The object Ksat is a list of the class geodata with the following elements:

coords a matrix with the coordinates of the soil samples.

data measurements of the saturated hidraulic conductivity.

borders a data-frame with the coordinates of a polygon defining the borders of the area.

Data provided by Dr. Décio Cruciani, ESALQ/USP, Brasil. summary(Ksat) plot(Ksat, border=borders) kslineSpatial Prediction – Conventional Krigingksline .ksline.aux.1ksline.ksline.aux.1 spatialksline This function performs spatial prediction for given covariance parameters. Options implement the following kriging types: *SK* (simple kriging), *OK* (ordinary kriging), *KTE* (external trend kriging) and *UK* (universal kriging).

The function krige.convkrige.conv should be preferred, unless moving neighborhood is to be used.

```
ksline(geodata, coords = geodata$coords, data = geodata$data,
       locations, borders = NULL,
       cov.model = "matern",
       cov.pars=stop("covariance parameters (sigmasq and phi) needed"),
       kappa = 0.5, nugget = 0, micro.scale = 0,
       lambda = 1, m0 = "ok", nwin = "full",
       n.samples.backtransform = 500, trend = 1, d = 2,
       ktedata = NULL, ktelocations = NULL, aniso.pars = NULL,
       signal = FALSE, dist.epsilon = 1e-10, messages)
```

a list containing elements coords and data as described next. Typically an object of the classclass "geodata" - a geoR data-set. If not provided the arguments coords and data must be provided instead.

an n $\times 2 matrixwhereeachrowhasthe 2-D coordinatesofthendatalocations.Bydefaultittakesthecompo$

an N $\times 2 matrixordata - framewiththe 2 - DcoordinatesoftheNpredictionlocations, oralistforwhichthefirsttwocomponentsareused.Inputisint$

a vector with 2 elements or an n $\times 2 matrixwiththecovarianceparameters \sigma^2 (partialsill) and \phi (rangepar$

micro-scale variance. If different from zero, the nugget variance is divided into 2 terms: *micro-scale variance* and *measurement error*. This might affect the precision of the predictions. In practice, these two variance components are usually indistinguishable but the distinction can be made here if justifiable.

string indicating the name of the model for the correlation function. Further details in the documentation for cov.spatialcov.spatial. Defaults are equivalent to the *exponential* model.

additional smoothness parameter required by the following correlation functions: "matern", "powered.exponential", "cauchy" and "gneiting.matern".

numeric value of the Box-Cox transformation parameter. The value $\lambda = 1 correspondstonotransformationand \lambda = 0 correspondstothelog - transformation.Predictionresultsareback-transformedandreturnedisthesamescaleasfortheorigin$

If "full" *global neighborhood* is used i.e., all data values are used in the prediction of every prediction location. An integer number defines the *moving neighborhood* algorithm. The number provided is used as the number closest neighbors to be used for the prediction at each location. Defaults to "full".

number of samples used in the back-transformation. When transformations are used (specified by an argument lambda), back-transformations are usually performed by sampling from the predictive distribution and then back-transforming the sampled values. The exceptions are for $\lambda = 0 (log - transformation) and \lambda = 1 (notransformation).onlyrequiredifm0 = "kt" (universalkriging).Possible$

spatial dimension, 1 defines a prediction on a line, 2 on a plane (the default).

only required if m0 = "kte". A vector or matrix with the values of the external trend (covariates) at the data locations.

only required if m0 = "kte". A vector or matrix with the values of the external trend (covariates) at the prediction locations.

parameters for geometric anisotropy correction. If aniso.pars = FALSE no correction is made, otherwise a two elements vector with values for the anisotropy parameters must be provided. Anisotropy correction consists of a transformation of the data and prediction coordinates performed by the function coords.anisocoords.aniso.

logical. If TRUE the signal is predicted, otherwise the variable is predicted. If no transformation is performed the expectations are the same in both cases and the difference is only for values of the kriging variance, if the value of the nugget is different from zero.

a numeric value. Points which are separated by a distance less than this value are considered co-located.

logical. Indicates whether or not status messages are printed on the screen (or other output device) while the function is running. An object of the classclass kriging which is a list with the following components:

the predicted values.

the kriging variances.

the difference between the predicted value and the global mean. Represents the contribution to the neighboring data to the prediction at each point.

values of the arithmetic and weighted mean of the data and standard deviations. The weighted mean corresponds to the estimated value of the global mean.

the matrix with trend if m0 = ”kt” (universal kriging).

the matrix with trend if m0 = ”kte” (external trend kriging).

the value of the mean which is implicitly estimated for m0 = ”ok”, ”kte” or ”kt”.

weight of mean. The predicted value is an weighted average between the global mean and the values at the neighboring locations. The value returned is the weight of the mean.

the coordinates of the prediction locations.

status messages returned by the algorithm.

the function call. This is a preliminary and inefficient function implementing kriging methods. For predictions using global neighborhood the function krige.convkrige.conv should be used instead. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. krige.convkrige.conv for a more efficient implementation of conventional kriging methods,
krige.bayeskrige.bayes for Bayesian prediction. loci ¡- expand.grid(seq(0,1,l=31), seq(0,1,l=31)) kc ¡- ksline(s100, loc=loci, cov.pars=c(1, .25)) par(mfrow=c(1,2)) image(kc, main=”kriging estimates”) image(kc, val=sqrt(kc$krige.var), main = ”krigingstd.errors$”) landim1Data from Landim’s booklandim1 datasetslandim1 Artificial or non-specified data from Paulo Landim’s book

```
data(landim1)
```

A data frame with 38 observations on the following 4 variables.

EW a numeric vector with the east-west coordinates.

NS a numeric vector with the north-south coordinates.

A a numeric vector with data on a first variable.

B a numeric vector with data on a second variable.

Landim, P. M. B. (2004) *Análise estatística de dados geológicos.* Editora Unesp. Data from Table˜1, pg.12. data(landim) plot(as.geodata(landim1, data.col=3)) plot(as.geodata(landim1, data.col=4)) legend.krigeAdd a legend to a image with kriging resultslegend.krige spatiallegend.krige aplotlegend.krige This function allows adds a legend to an image plot generated by image.krigingimage.kriging or image.krige.bayesimage.krige.bayes. It can be called internally by these functions or directly by the user.

```
legend.krige(x.leg, y.leg, values, scale.vals,
             vertical = FALSE, offset.leg = 1, ...)
```

limits for the legend in the x direction.

limits for the legend in the y direction.

values plotted in the image.

optional. Values to appear in the legend. If not provided the function prettypretty is used to define the values.

If TRUE the legend is drawn in the vertical direction. Defaults to FALSE.

numeric value controlling the distance between the legend text and the legend box.

further arguments to be passed to the function texttext. A legend is added to the current plot. No values are returned. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. image.krigingimage.kriging, image.krige.bayesimage.krige.bayes. See examples in the documentation for image.kriging likfitLikelihood Based Parameter Estimation for Gaussian Random Fieldslikfit .negloglik.GRFlikfit.negloglik.GRF check.parameters.valueslikfitcheck.parameters.values fitted.likGRFlikfitfitted.likGRF likfit.limitslikfitlikfit.limits logLik.likGRFlikfitlogLik.likGRF logLik.likGRFlikfitlogLik.likGRF resid.likGRFlikfitresid.likGRF residuals.likGRFlikfitresiduals.likGRF spatiallikfit modelslikfit *Maximum likelihood* (ML) or *restricted maximum likelihood* (REML) parameter estimation for (transformed) Gaussian random fields.

```
likfit(geodata, coords = geodata$coords, data = geodata$data,
       trend = "cte", ini.cov.pars, fix.nugget = FALSE, nugget = 0,
       fix.kappa = TRUE, kappa = 0.5, fix.lambda = TRUE, lambda = 1,
       fix.psiA = TRUE, psiA = 0, fix.psiR = TRUE, psiR = 1,
       cov.model, realisations, lik.method = "ML", components = TRUE,
       nospatial = TRUE, limits = pars.limits(),
       print.pars = FALSE, messages, ...)

## S3 method for class 'likGRF'
fitted(object, spatial = TRUE, ...)

## S3 method for class 'likGRF'
resid(object, spatial = FALSE, ...)
```

a list containing elements coords and data as described next. Typically an object of the class "geodata". If not provided the arguments coords and data must be provided instead.

an n $\times 2 matrixwhereeachrowhasthe 2-Dcoordinatesofthendatalocations.Bydefaultittakesthecomponentcoordsc$

a vector with n data values. By default it takes the component data of the argument geodata, if provided.

specifies the mean part of the model. See documentation of trend.spatialtrend.spatial for further details. Defaults to "cte".

initial values for the covariance parameters: $\sigma^2 (partialsill) and \phi (rangeparameter).Typicallyavectorwithtwocomponents.Howeveramatrixcanbeusedtoprovide$

value of the nugget parameter. Regarded as a fixed value if fix.nugget = TRUE otherwise as the initial value for the minimisation algorithm. Defaults to zero.

logical, indicating whether the extra parameter $\kappa shouldberegardedasfixed(fix.kappa = TRUE)orshouldbeestimated(fix.kappa = FALSE).DefaultstoTRUE.v$
$"matern","powered.exponential","cauchy"or"gneiting.matern".Formoredetailsoncovariancefunctionsseedocu$

logical, indicating whether the Box-Cox transformation parameter $\lambda shouldberegardedasfixed(fix.lambda = TRUE)orshouldbebeestimated(fix.lambda = FALSE).DefaultstoTR$
$Coxtransformationparameter\lambda.Regardedasafixedvalueiffix.lambda = TRUEotherwiseastheinitialvaluefort$
$transformation.$

logical, indicating whether the anisotropy angle parameter $\psi_R shouldberegardedasfixed(fix.psiA = TRUE)orshouldbeestimated(fix.psiA = FALSE).DefaultstoTRUE.v$

logical, indicating whether the anisotropy ratio parameter $\psi_R shouldberegardedasfixed(fix.psiR = TRUE)orshouldbeestimated(fix.psiR = FALSE).DefaultstoTRUE.v$

a string specifying the model for the correlation function. For further details see documentation for cov.spatialcov.spatial. Reads values from an variomodel object passed to ini.cov.pars if any, otherwise defaults to the *exponential* model.

optional. Logical or a vector indicating the number of replication for each datum. For further information see DETAILS below and documentation for as.geodataas.geodata.

(formely method.lik) options are "ML" for maximum likelihood and "REML" for restricted maximum likelihood. Defaults to "ML".

an n $\times 3 data - framewithfittedvaluesforthethreemodelcomponents :$
$trend, spatialandresiduals.SeethesectionDETAILSbelowforthemodelspecification.logical.IfTRUEparameter$

values defining lower and upper limits for the model parameters used in the numerical minimisation. The auxiliary function pars.limitspars.limits is called to set the limits. See also Limits in DETAILS below.

logical. If TRUE the parameters and the value of the negative log-likelihood (up to a constant) are printed each time the function to be minimised is called.

logical. Indicates whether status messages should be printed on the screen (or output device) while the function is running.

additional parameters to be passed to the minimisation function. Typically arguments of the type control() which controls the behavior of the minimisation algorithm. For further details see documentation for the minimisation function optimoptim.

an object with output of the function likfit.

logical, determines whether the spatial component of the model in included in the output. The geostatistical model components are: *trend*, *spatial* and *residulas*. See DETAILS. This function estimate the parameters of the Gaussian random field model, specified as: Y(x) $= \mu(x) + S(x) + ewhere$

x defines a spatial location. Typically Euclidean coordinates on a plane.

Y is the variable been observed.

$\mu(x) = X\beta isthemeancomponentofthemodel(trend).S(x)isastationaryGaussianprocesswithvar$

e is the error term with variance parameter $\tau^2(nuggetvariance)$.

The additional parameter $\lambda allowsfortheBox - Coxtransformationoftheresponsevariable.Ifused(i.e.if\lambda \neq 1)Y(x)aboveisreplacedbyg(Y(x))suchth$

Two particular cases are $\lambda = 1 whichindicatesnotransformationand\lambda = 0 indicatingthelog-transformation.$

Numerical minimization

In general parameter estimation is performed numerically using the function optimoptim to minimise the negative log-likelihood computed by the function negloglik.GRF. If the nugget, anisotropy $(\psi_A, \psi_R), smoothness(\kappa)andtransformation(\lambda)parametersareheldfixedthenthenumericalminimisat$ $dimensionandthefunctionoptimizeoptimizeisusedinsteadofoptim.Inthiscaseinitialvaluesareirrelev$

Limits

Lower and upper limits for parameter values can be individually specified using the function link{pars.limits}. For example, including the following in the function call:

limits = pars.limits(phi=c(0, 10), lambda=c(-2.5, 2.5)),

will change the limits for the parameters $\phi and\lambda.Defaultvaluesareusediftheargumentlimitsisnotprovided.$

There are internal reparametrisation depending on the options for parameters to be estimated. For instance for the common situation when fix.nugget=FALSE the minimisation is performed in a reduced parameter space using $\tau^2_{rel} = \frac{\tau^2}{\sigma^2}.Inthiscasevaluesof\sigma^2and\beta arethengivenbyanalyticalexpressionswhicharefunctionofthetwopara$

Since parameter values are found by numerical optimization using the function optimoptim, in given circunstances the algorithm may not converge to correct parameter values when called with default options and the user may need to pass extra options for the optimizer. For instance the function optim takes a control argument. The user should try different initial values and if the parameters have different orders of magnitude may need to use options to scale the parameters. Some possible workarounds in case of problems include:

- rescale you data values (dividing by a constant, say)
- rescale your coordinates (subtracting values and/or dividing by constants)
- Use the mechanism to pass control() options for the optimiser internally

Transformation If the fix.lambda = FALSE and nospatial = FALSE the Box-Cox parameter for the model without the spatial component is obtained numerically, with log-likelihood computed by the function boxcox.ns.

Multiple initial values can be specified providing a n matrix for the argument ini.cov.pars and/or providing a vector for the values of the remaining model parameters. In this case the log-likelihood is computed for all combinations of the model parameters. The parameter set which maximises the value of the log-likelihood is then used to start the minimisation algorithm.

Alternatively the argument ini.cov.pars can take an object of the class eyefit or variomodel. This allows the usage of an output of the functions eyefiteyefit, variofitvariofit or likfitlikfit be used as initial value.

The argument realisations allows sets of data *assumed to be independent* replications of the same process. Data on different realisations may or may not be co-located. For instance, data collected at different times can be pooled together in the parameter estimation assuming time independence. The argument realisations takes a vector indicating the replication number (e.g. the times). If realisations = TRUE the code looks for an element named realisations in the geodata object. The log-likelihoods are computed for each replication and added together.   An object of the classes "likGRF" and "variomodel".
The function summary.likGRFsummary.likGRF is used to print a summary of the fitted model.
The object is a list with the following components:

a string with the name of the correlation function.

value of the nugget parameter $\tau^2. Thisisanestimateiffix.nugget = FALSEotherwise, afixedvalue.avectorwithth$

value of the smoothness parameter. Valid only if the correlation function is one of: "matern", "powered.exponential", "cauchy" or "gneiting.matern".

estimate of mean parameter $\beta. Thiscanbeascalarorvectordependingonthetrend(covariates)specifiedinthemodel.es$

values of the Box-Cox transformation parameter. A fixed value if fix.lambda = TRUE otherwise the estimate value.

fixed values or estimates of the anisotropy parameters, according to the function call.

estimation method used, "ML" (maximum likelihood) or "REML" (restricted maximum likelihood).

the value of the maximized likelihood.

number of estimated parameters.

value of the Akaike Information Criteria, AIC=-2 ln(L) + 2 p where L is the maximised likelihood and p is the number of parameters in the model.

value of the Bayesian Information Criteria, BIC=-2ln(L) + p log(n), where n is the number of data, L,p as for AIC above.

a data-frame with all model parameters, their status (estimated or fixed) and values.

results returned by the minimisation function.

maximum distance between 2 data points. This information relevant for other functions which use outputs from likfit.

the trend (covariates) matrix X.

numerical value of the logarithm of the Jacobian of the transformation.

estimates for the model without the spatial component.

the function call. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. summary.likGRFsummary.likGRF for summary of the results, plot.variogramplot.variogram, lines.variogramlines.variogram and lines.variomodellines.variomodel for graphical output, proflikproflik for computing profile likelihoods, variofitvariofit and for other estimation methods, and optimoptim for the numerical minimisation function. Not run: ml ¡- likfit(s100, ini=c(0.5, 0.5), fix.nug = TRUE) ml summary(ml) reml ¡- likfit(s100, ini=c(0.5, 0.5), fix.nug = TRUE, lik.met = "REML") summary(reml) plot(variog(s100)) lines(ml) lines(reml, lty = 2)

End(Not run) likfitBGCCMFits the bivariate Gaussian common component geostatistical modellikfitBGCCM .naiveLL.BGCCMlikfitBGCCM.naiveLL.BGCCM .negloglikBGCCMlikfitBGCCM.negloglikBGCCM loglikBGCCMlikfitBGCCMloglikBGCCM spatiallikfitBGCCM modelslikfitBGCCM Computes maximum likelihood estimates of the bivariate Gaussian common component geostatistical model.

```
likfitBGCCM(geodata1, geodata2, ini.sigmasq, ini.phi,
            cov0.model="matern", cov1.model="matern", cov2.model="matern",
            kappa0=0.5, kappa1=0.5, kappa2=0.5,
            fc.min = c("optim", "nlminb"), ...)
```

an object of the class geodata with the first variable.

an object of the class geodata with the second variable.

optional, a vector with initial values for the variance parameters. If not provided default values are used.

optional, a vector with initial values for the correlation range parameters. If not provided default values are used.

covariance model for each of the processes. See cov.spatialcov.spatial for details.

extra parameter for some covariance models.

a string indication which function should be used to minimise the negative of the log-likelihood.

further arguments to be passed to optimoptim or nlminbnlminb. A list with model fitting information to which the class BGCCM is assigned.

a 2 elements vector with mean estimates.

a 4 elements vector with variance estimates.

a 3 elements vector with estimated correlation parameters values.

a scalar. Maximised value of the log-likelihood.

results returned by optimoptim or nlminbnlminb.

and other information related to the model fitting. Warning This is a new function and still in draft format and pretty much untested. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. optimoptim, nlminbnlminb, varcovBGCCMvarcovBGCCM, as.geodataas.geodata, likfitlikfit. see http://www.leg.ufpr.br/geoR/tutorials/CCM.R lines.variogramLine with a

Empirical Variogramlines.variogram spatiallines.variogram aplotlines.variogram A sample (empirical) variogram computed using the function variogvariog is added to the current plot.

```
## S3 method for class 'variogram'
lines(x, max.dist, type = "o",  scaled = FALSE,
         pts.range.cex, ...)
```

an object of the class "variogram", typically an output from the function variogvariog.

maximum distance for the x-axis. By default takes the maximum distance for which the sample variogram was computed.

type of line for the empirical variogram. The default is "o" (dots and lines). See documentation for lineslines for further details.

logical. If TRUE the variogram values are divided by the sample variance. This allows comparison between variograms of different variables.

optional. A two elements vector with maximum and minimum values for the caracter expansion factor cex. If provided the point sizes in binned variogram are proportional to the number of pairs of points used to compute each bin.

other arguments to be passed to the function lineslines. A line with the empirical variogram is added to the plot in the current graphics device. No values are returned. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. variogvariog, lines.variogramlines.variogram, lines.variomodellines.variomodel, variog.model.envvariog.model.env, variog.mc.envvariog.mc.env, plot.grfplot.grf, lineslines. lines.variogram.envelopeAdds Envelopes Lines to a Variogram Plotlines.variogram.envelope spatiallines.variogram.envelope aplotlines.variogram.envelope Variogram envelopes computed by variog.model.envvariog.model.env or variog.mc.envvariog.mc.env are added to the current variogram plot.

```
## S3 method for class 'variogram.envelope'
lines(x, lty = 3, ...)
```

an object of the class "variogram.envelope", typically an output of the functions variog.model.envvariog.model.env or variog.mc.envvariog.mc.env.

line type. Defaults to 3.

arguments to be passed to the function lineslines. Lines defining the variogram envelope are added to the plotin the current graphics device. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. variogvariog for variogram computation, variog.model.envvariog.model.env and variog.mc.envvariog.mc.env for computation of variogram envelopes, and lineslines for the generic function. s100.vario ¡- variog(s100, max.dist = 1) s100.ml ¡- likfit(s100, ini=c(.5, .5)) s100.mod.env ¡- variog.model.env(s100, obj.variog = s100.vario, model

= s100.ml) s100.mc.env ¡- variog.mc.env(s100, obj.variog = s100.vario) plot(s100.vario) lines(s100.mod.env) lines(s100.mc.env, lwd=2)
lines.variomodelAdds a Line with a Variogram Model to a Variogram Plotlines.variomodel lines.variomodel.defaultlines.variomodellines.variomodel.default spatiallines.variomodel aplotlines.variomodel This function adds a line with a variogram model specifyed by the user to a current variogram plot. The variogram is specifyed either by passing a list with values for the variogram elements or using each argument in the function.

```
## S3 method for class 'variomodel'
lines(x, ...)
## Default S3 method:
lines.variomodel(x, cov.model, cov.pars, nugget, kappa,
                              max.dist, scaled = FALSE, ...)
```

a list with the values for the following components: cov.model, cov.pars, nugget, kappa , max.dist; or a numeric vector with values for x-axis values for the variogram (distances). This argument is not required if the other arguments in the function are provided, otherwise is compulsory. If a list is provided the arguments which match the list elements are ignored.

a string with the type of the variogram function. See documentation of cov.spatialcov.spatial for further details.

a vector or matrix with the values for the partial sill $(\sigma^2) and range (\phi) parameters. a scalar with the value of the nugget (\tau^2) parameter.$

a scalar with the value of the smoothness $(\kappa) parameters. Only required if cov.model is one of the following: "matern", "powered.exponential", "cauchy" and "gneiting.matern"$

maximum distance (x-axis) to compute and draw the line representing the variogram model. If a list is provided in x the default is the distance given by x$max.dist. If a vector is provided in x it takes max(x).

logical. If TRUE the total sill in the plot is equals to 1.

arguments to be passed to the function curvecurve. Adds a line with a variogram model to a plot. In conjuction with plot.variogramplot.variogram can be used for instance to compare sample variograms against fitted models returned by variofitvariofit and/or likfitlikfit. A line with a variogram model is added to a plot on the current graphics device. No values are returned. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. lines.variomodel.krige.bayeslines.variomodel.krige.bayes, lines.variomodel.grflines.variomodel.grf, lines.variomodel.variofitlines.variomodel.variofit, lines.variomodel.likGRFlines.variomodel.likGRF, plot.variogramplot.variogram, lines.variogramlines.variogram, variofitvariofit, likfitlikfit, curvecurve. computing and ploting empirical variogram vario ¡- variog(s100, max.dist = 1) plot(vario) estimating parameters by weighted least squares vario.wls ¡- variofit(vario, ini = c(1, .3), fix.nugget = TRUE) adding fitted model to the plot lines(vario.wls) Ploting different variogram models plot(0:1, 0:1, type="n")

lines.variomodel(cov.model = "exp", cov.pars = c(.7, .25), nug = 0.3, max.dist = 1) an alternative way to do this is: my.model ¡- list(cov.model = "exp", cov.pars = c(.7, .25), nugget = 0.3, max.dist = 1) lines.variomodel(my.model, lwd = 2) now adding another model lines.variomodel(cov.m = "mat", cov.p = c(.7, .25), nug = 0.3, max.dist = 1, kappa = 1, lty = 2) adding the so-called "nested" models two exponential structures lines.variomodel(seq(0,1,l=101), cov.model="exp", cov.pars=rbind(c(0.6,0.15),c(0.4,0.25)), nug=0, col=2) exponential and spherical structures lines.variomodel(seq(0,1,l=101), cov.model=c("exp", "sph"), cov.pars=rbind(c(0.6,0.15), c(0.4,0.75)), nug=0, col=3) lines.variomodel.grfLines with True Variogram for Simulated Datalines.variomodel.grf spatiallines.variomodel.grf aplotlines.variomodel.grf This functions adds to the graphics device a line with the theoretical (true) variogram used when generating simulations with the function grfgrf.

```
## S3 method for class 'grf'
lines.variomodel(x, max.dist, n = 100, ...)
```

an object from the class grf typically an output of the function grfgrf.

maximum distance to compute and plot the true variogram. Defaults to the maximum distance between two data locations.

number of points used to compute and draw the variogram line.

further arguments to be passed to the function curvecurve. A line with the true variogram model is added to the current plot on the graphics device. No values are returned. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. lines.variomodellines.variomodel, grfgrf, plot.grfplot.grf, curvecurve. sim ¡- grf(100, cov.pars=c(1, .25)) simulates data plot(variog(sim, max.dist=1)) plot empirical variogram lines.variomodel.krige.bayesAdds a Bayesian Estimate of the Variogram to a Plotlines.variomodel.krige.bayes spatiallines.variomodel.krige.bayes aplot-lines.variomodel.krige.bayes Adds a Bayesian estimate of the variogram model to a plot typically with an empirical variogram. The estimate is a chosen summary (mean, mode or mean) of the posterior distribution returned by the function krige.bayeskrige.bayes.

```
## S3 method for class 'krige.bayes'
lines.variomodel(x, summary.posterior, max.dist, uvec,
                 posterior = c("variogram", "parameters"),  ...)
```

an object of the class krige.bayes, typically an output of the function krige.bayeskrige.bayes.

specify which summary of the posterior distribution should be used as the parameter estimate. Options are "mean", "median" or "mode". See DETAILS below.

numerical, the maximum distance for the x-axis.

a numerical vector with support points to compute the variogram values. Only used if posterior = "variogram". Defaults to seq(0, max.dist, length = 51).

indicates whether the the variogram line is based on the posterior of the variogram function (default) or the posterior of the model parameters.

arguments passed to the functions lineslines or curvecurve. The function krige.bayeskrige.bayes returns samples from the posterior distribution of the parameters $(\sigma^2, \phi, \tau_{rel}^2)$.

This function allows for two basic options to draw a line with a summary of the variogram function.

1. "variogram": for each sample of the parameters the variogram function is computed at the support points defined in the argument uvec. Then a function provided by the user in the argument summary.posterior is used to compute a summary of the values obtained at each support point.

2. "parameters": in this case summaries of the posterior distribution of the model parameters as "plugged-in" in the variogram function. One of the options "mode" (default) ,"median" or "mean" can be provided in the argument summary.posterior. The option mode, uses the mode of $(\phi, \tau_{rel}^2) andthemodeofof\sigma^2 conditionalonthemodesoftheformerparameters.Fortheoptionsmea$

A line with the estimated variogram plot is added to the plot in the current graphics device. No values are returned. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. lines.variomodellines.variomodel, krige.bayeskrige.bayes and lineslines. See examples in the documentation of the function krige.bayes(). lines.variomodel.likGRFAdds a Variogram Line to a Variogram Plotlines.variomodel.likGRF spatiallines.variomodel.likGRF aplotlines.variomodel.likGRF This function adds a fitted variogram based on the estimates of the model parameters returned by the function likfitlikfit to a current variogram plot.

```
## S3 method for class 'likGRF'
lines.variomodel(x, max.dist, scaled = FALSE, ...)
```

an object of the class likGRF which is a list containing information about the fitted model parameters, typically an output of the function likfitlikfit.

maximum distance (x-axis) to compute and draw the line representing the variogram model. The default is the distance given by obj$max.dist.

logical. If TRUE the total sill in the plot is equals to 1.

arguments to be passed to the function curvecurve. Adds variogram model(s) to a plot. In conjuction with plot.variogramplot.variogram can be used to compare sample variograms against fitted models returned by likfitlikfit. A line with a variogram model is added to a plot on the current graphics device. No values are returned. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. lines.variomodellines.variomodel, lines.variomodel.variofitlines.variomodel.variofit, plot.variogramplot.variogram, lines.variogramlines.variogram, variofitvariofit, likfitlikfit, curvecurve. compute and plot empirical variogram vario ¡- variog(s100, max.dist = 1)

plot(vario) estimate parameters vario.ml ¡- likfit(s100, ini = c(1, .3), fix.nugget = TRUE) adds fitted model to the plot lines(vario.ml) lines.variomodel.variofitAdds a Line with a Fitted Variogram Model to a Variogram Plotlines.variomodel.variofit spatiallines.variomodel.variofit aplot-lines.variomodel.variofit This function adds a line with the variogram model fitted by the function variofitvariofit to a current variogram plot.

```
## S3 method for class 'variofit'
lines.variomodel(x, max.dist, scaled = FALSE, ...)
```

an object of the class variofit which is a list containing information about the fitted model parameters, typically an output of the function variofitvariofit.

maximum distance (x-axis) to compute and draw the line representing the variogram model. The default is the distance given by x$max.dist.

logical. If TRUE the total sill in the plot is set to 1.

arguments to be passed to the function curvecurve. Adds fitted variogram model to a plot. In conjuction with plot.variogramplot.variogram can be used to compare empirical variograms against fitted models returned by variofitvariofit. A line with a variogram model is added to a plot on the current graphics device. No values are returned. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. lines.variomodellines.variomodel, lines.variomodel.likGRFlines.variomodel.likGRF, plot.variogramplot.variogram, lines.variogramlines.variogram, variofitvariofit, likfitlikfit, curvecurve. compute and plot empirical variogram vario ¡- variog(s100, max.dist = 1) plot(vario) estimate parameters vario.wls ¡- variofit(vario, ini = c(1, .3), fix.nugget = TRUE) adds fitted model to the plot lines(vario.wls) locations.insideSelect prediction locations inside borderslocations.inside spatiallocations.inside Selects the prediction locations located inside a polygon defining borders of a region where prediction is aimed. Typically internally called by geoR functions krige.bayeskrige.bayes, krige.convkrige.conv, kslineksline.

```
locations.inside(locations, borders, as.is = TRUE, ...)
```

a two columns matrix or dqata frame with coordinates of the prediction locations.

a two column matrix or data-frame with coordinates of a polygon defining the borders of the region.

logical defining if the returned object of of the same type (list, data-frame or matrix) as the provided in locations. If FALSE the function returns a matrix.

arguments to be passed to the internal function .geoR_pip and currently not used. A two columns matrix, data-frame or a list with 2 elements with coordinates of points inside the borders. overlayoverlay,coordinatescoordinates, SpatialPointsSpatialPoints. gr ¡- pred$_grid(parana$borders, by=20) plot(gr, asp=1, pch="+") polygon(parana$borders)gr.in < -locations.inside(gr, parana$borders) points(gr.in, col=2, pch="+") loglik.GRFLog-Likelihood for a Gaussian Random Fieldloglik.GRF spatialloglik.GRF This function computes the value of the log-likelihood for a Gaussian random field.

```
loglik.GRF(geodata, coords = geodata$coords, data = geodata$data,
           obj.model = NULL, cov.model = "exp", cov.pars, nugget = 0,
           kappa = 0.5, lambda = 1, psiR = 1, psiA = 0,
           trend = "cte", method.lik = "ML", compute.dists = TRUE,
           realisations = NULL)
```

a list containing elements coords and data as described next. Typically an object of the class "geodata" - a geoR data-set. If not provided the arguments coords and data must be provided instead.

an n $\times 2 matrix, each row containing Euclidean coordinates of the n data locations. By default it takes the el$

a object of the class variomodel with a fitted model. Tipically an output of likfitlikfit or variofitvariofit.

a string specifying the model for the correlation function. For further details see documentation for cov.spatialcov.spatial.

a vector with 2 elements with values of the covariance parameters $\sigma^2 (partial sill) and \phi (range parameter). value of the nugget parameter. Defaults to 0.$

value of the smoothness parameter. Defaults to 0.5.

value of the Box-Cox tranformation parameter. Defaults to 1.

value of the anisotropy ratio parameter. Defaults to 1, corresponding to isotropy.

value (in radians) of the anisotropy rotation parameter. Defaults to zero.

specifies the mean part of the model. The options are: "cte" (constant mean), "1st" (a first order polynomial on the coordinates), "2nd" (a second order polynomial on the coordinates), or a formula of the type ˜X where X is a matrix with the covariates (external trend). Defaults to "cte".

options are "ML" for likelihood and "REML" for restricted likelihood. Defaults to "ML".

for internal use with other function. Don't change the default unless you know what you are doing.

optional. A vector indicating replication number for each data. For more details see as.geodataas.geodata. The expression log-likelihood is: l$(\theta) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(y - F\beta)'\Sigma^{-1}(y - F\beta), where n is the size of the data vector y, \beta is the mean (vector) parameter with dimention p, \Sigma is the covari$

The expression restricted log-likelihood is: rl$(\theta) = -\frac{n-p}{2}\log(2\pi) + \frac{1}{2}\log|F'F| - \frac{1}{2}\log|\Sigma| - \frac{1}{2}\log|F'\Sigma F| - \frac{1}{2}(y - F\beta)'\Sigma^{-1}(y - F\beta)$.
The numerical value of the log-likelihood. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. likfitlikfit for likelihood-based parameter estimation. loglik.GRF(s100, cov.pars=c(0.8, .25), nugget=0.2) loglik.GRF(s100, cov.pars=c(0.8, .25), nugget=0.2, met="RML")

Computing the likelihood of a model fitted by ML s100.ml ¡- likfit(s100, ini=c(1, .5)) s100.ml s100.ml$loglik loglik.GRF(s100, obj = s100.ml)$

Computing the likelihood of a variogram fitted model s100.v ¡- variog(s100, max.dist=1) s100.vf ¡- variofit(s100.v, ini=c(1, .5)) s100.vf loglik.GRF(s100, obj=s100.vf) maternComputer Values of the Matern Correlation Functionmatern spatialmatern This function computes values of the Matérn correlation function for given distances and parameters.

```
matern(u, phi, kappa)
```

a vector, matrix or array with values of the distances between pairs of data locations.

value of the range parameter $\phi.valueofthesmoothnessparameter\kappa.$ The Matérn model is defined as:

$\rho(u;\phi,\kappa) = \{2^{\kappa-1}\Gamma(\kappa)\}^{-1}(u/\phi)^{\kappa}K_{\kappa}(u/\phi)$

where $\phi and \kappa are parameters and K_{\kappa}(\cdot) denotes the modified Bessel function of the third kind of order \kappa. The family$
A vector matrix or array, according to the argument u, with the values of the Matérn correlation function for the given distances. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. cov.spatialcov.spatial for the correlation functions implemented in geoR, and besselKbesselK for computation of the Bessel functions. Models with fixed range and varying smoothness parameter curve(matern(x, phi= 0.25, kappa = 0.5),from = 0, to = 1.5, xlab = "distance", ylab = expression(rho(h)), lty = 2, main=expression(paste("varying ", kappa, " and fixed ", phi))) curve(matern(x, phi= 0.25, kappa = 1),from = 0, to = 1.5, add = TRUE) curve(matern(x, phi= 0.25, kappa = 2),from = 0, to = 1.5, add = TRUE, lwd = 2, lty=2) curve(matern(x, phi= 0.25, kappa = 3),from = 0, to = 1.5, add = TRUE, lwd = 2) legend("topright", expression(kappa==0.5, kappa==1.5, kappa==2, kappa==3), lty=c(2,1,2,1), lwd=c(1,1,2,2))

Correlations with equivalent "practical range" and varying smoothness parameter curve(matern(x, phi = 0.25, kappa = 0.5),from = 0, to = 1, xlab = "distance", ylab = expression(gamma(h)), lty = 2, main = "models with equivalent p̈racticalr̈ange") curve(matern(x, phi = 0.188, kappa = 1),from = 0, to = 1, add = TRUE) curve(matern(x, phi = 0.14, kappa = 2),from = 0, to = 1, add = TRUE, lwd=2, lty=2) curve(matern(x, phi = 0.117, kappa = 2), from = 0, to = 1, add = TRUE, lwd=2) legend("topright", expression(list(kappa == 0.5, phi == 0.250), list(kappa == 1, phi == 0.188), list(kappa == 2, phi == 0.140), list(kappa == 3, phi == 0.117)), lty=c(2,1,2,1), lwd=c(1,1,2,2)) names.geodataLists names of the key elements of a geodata objectnames.geodata spatialnames.geodata manipnames.geodata Produces a list with the names of the main elements of geodata object: coords, data, units.m, covariate and realisation. Can be useful to list names before using {subset.geodata}.

```
## S3 method for class 'geodata'
names(x)
```

an object of the class geodata. A list with

names of the coordinates in the geodata object.

name(s) of the data elements in the geodata object.

returns the string units.m.

return the covariate(s) name(s) if present in the geodata object returns the string units.m if present in the geodata object.

other elements in the geodata object. namesnames, subset.geodatasubset.geodata, as.geodataas.geodata. names(ca20) nearlocNear location to a pointnearloc spatialnearloc For a given set of points and locations identified by 2D coordinates this function finds the nearest location of each point

```
nearloc(points, locations, positions = FALSE)
```

a matrix, data-frame or list with the 2D coordinates of a set of points for which you want to find the nearest location.

a matrix, data-frame or list with the 2D coordinates of a set of locations.

logical defining what to be returned. If TRUE the function returns the positions of the locations, otherwise the coordinates of the locations. Defaults to FALSE. If positions = FALSE the function returns a matrix, data-frame or list of the same type and size as the object provided in the argument points with the coordinates of the nearest locations.

If positions = FALSE the function returns a vector with the position of the nearest points in the locations object. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. loccoordsloccoords set.seed(276) gr ¡- expand.grid(seq(0,1, l=11), seq(0,1, l=11)) plot(gr, asp=1) pts ¡- matrix(runif(10), nc=2) points(pts, pch=19) near ¡- nearloc(points=pts, locations=gr) points(near, pch=19, col=2) rownames(near) nearloc(points=pts, locations=gr, pos=TRUE) output.control Defines output options for prediction functions output.control spatialoutput.control Auxiliary function defining output options for krige.bayeskrige.bayes and krige.convkrige.conv.

```
output.control(n.posterior, n.predictive, moments, n.back.moments,
               simulations.predictive, mean.var, quantile,
               threshold, sim.means, sim.vars, signal, messages)
```

number of samples to be taken from the posterior distribution. Defaults to 1000.

number of samples to be taken from the predictive distribution. Default equals to n.posterior.

logical. Indicates whether the moments of the predictive distribution are returned. If lambda = 1 there is no transformation/back-transformation. If lambda = 0 or lambda = 0.5 the moments are back-transformed by analytical expressions. For other cases the back-transformation is done by simulation. Defaults to TRUE.

number of sample to back-transform moments by simulation. Defaults to 1000.

logical. Defines whether to draw simulations from the predictive distribution. Only considered if prediction locations are provided in the argument locations of the main functions. Defaults to FALSE but changed to TRUE if an integer greater then zero is provided in the argument n.predictive and/or simulations are required in order to compute quantities required by other arguments such as threshold, quantiles and some values of the transformation parameter.

logical (optional). Indicates whether mean and variances of the simulations of the predictive distributions are computed and returned.

a (optional) numeric vector. If provided indicates whether quantiles of the simulations from the predictive distribution are computed and returned. If a vector with numbers in the interval [0,1] is provided, the output includes the object quantiles, which contains values of corresponding estimated quantiles. For example, if quantile = c(0.25, 0.50, 0.75) the function returns the quartiles of the predictive distributions at each of the prediction locations. If quantile = TRUE default values c(0.025, 0.5, 0.975) are assumed. A measure of uncertainty of the predictions, an alternative to the kriging standard error, computed by (quantile$_0.975 - quantile_0.025)/4. Onlyusedifpredictionlocationsareprovidedintheargumentlocations. Optional. Anumericalvector$

logical (optional). Indicates whether mean of each of the conditional simulations of the predictive distribution should be computed and returned. Defaults to TRUE, if simulations from the predictive are required.

logical (optional). Indicates whether variance of each of the conditional simulations of the predictive distribution should be computed and returned. Defaults to FALSE.

logical indicating whether the signal or the variable is to be predicted. Different defaults are set internally by functions calling output.control. See DETAILS below.

logical. Indicates whether or not status messages are printed on the output device while the function is running. Defaults to TRUE.

SIGNAL

This function is typically called by the geoR's prediction functions krige.bayeskrige.bayes and krige.convkrige.conv defining the output.

By default, krige.bayeskrige.bayes sets signal = TRUE and krige.convkrige.conv sets signal = FALSE.

The underlying model Y(x) = $\mu + S(x) + \epsilon assumesthatobservationsY(x)arenoisyversionsofasignalS(x)andVar(\epsilon) = \tau^2 isthenuggetvariance.$

If $\tau^2 = 0 theYandSareindistiguishable.$

If $\tau^2 > 0 andregardedasmeasurementerror, theoptionsignaldefineswhethertheS(signal = TRUE) orthevaria$
$Forthelatterthepredictionswill" honor" thedata, i.e. predictedvalueswillcoincidewiththedata, atdatalocations.$
$Forunsampledlocationsanduntransformeddata, thepredictedvaluesequalsdataregardlesssignal = TRUEorFAL$

The function krige.convkrige.conv has an argument micro.scale. If micro.scale ¿ 0 the error term is divided as $\epsilon = \epsilon_{ms} + \epsilon_{me} andthenuggetvarianceisdividedintotwoterms :$ *micro-scale variance*$andmeasurement$ *error*.
$Ifsignal = TRUEtheterm\epsilon_{ms} isregardedaspartofthesignalandconsequentlythe$*micro-scale variance*$isaddedtothe$
$Ifsignal = FALSEthetotalerrorvariance\tau^2 isaddedtothepredictionvariance.$

A list with processed arguments to be passed to the main function. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. The prediction functions krige.bayeskrige.bayes and krige.convkrige.conv. paranaRainfall Data from Parana State, Brasilparana maijunparanamaijun datasetsparana This data-set was used by Diggle and Ribeiro (2001) to illustrate the methods discussed in the paper. The data reported analysis was carried out using the package geoR.

The data refers to average rainfall over different years for the period May-June (dry-season). It was collected at 143 recording stations throughout Paraná State, Brasil.

```
data(parana)
```

The object parana of the class geodata, which is a list containing the following components:

coords a matrix with the coordinates of the recording stations.

data a vector with the average recorded rainfall for the May-June period.

borders a matrix with the coordinates defining the borders of Paraná state.

loci.paper a matrix with the coordinates of the four prediction locations discussed in the paper.

The data were collected at several recording stations at Paraná State, Brasil, belonging to the following companies: COPEL, IAPAR, DNAEE, SUREHMA and INEMET.

The data base was organized by Laura Regina Bernardes Kiihl (IAPAR, Instituto Agronômico do Paraná, Londrina, Brasil) and the fraction of the data included in this data-set was provided by Jacinta Loudovico Zamboti (Universidade Estadual de Londrina, Brasil). The coordinates of the borders of Paraná State were provided by João Henrique Caviglione (IAPAR). Diggle, P.J. & Ribeiro Jr, P.J. (2002) Bayesian inference in Gaussian model-based geostatistics. Geographical and Environmental Modelling, Vol. 6, No. 2, 129-146. summary(parana) plot(parana, bor=borders) pars.limitsSet limits for the parameter valuespars.limits spatialpars.limits modelspars.limits The functions likfitlikfit and variofitvariofit in the package geoR

```
pars.limits(phi = c(lower = 0, upper = +Inf),
            sigmasq = c(lower = 0, upper = +Inf),
            nugget.rel = c(lower = 0, upper = +Inf),
            kappa = c(lower = 0, upper = +Inf),
            kappa2 = c(lower = 0, upper = +Inf),
            lambda = c(lower = -3, upper = 3),
            psiR = c(lower = 1, upper = +Inf),
            psiA = c(lower = 0, upper = 2 * pi),
            tausq.rel = nugget.rel)
```

a two elements vector with limits for the parameter phi. Defaults to [0, +Inf]

idem for sigmasq. Defaults to [0, +Inf]

idem for nugget.rel. Defaults to [0, +Inf]

idem. Defaults to [0, +Inf]

idem for lambda. Defaults to [-3, +3]. Only used in likfitlikfit.

idem for psiR. Defaults to [1, +Inf]. Only used in likfitlikfit.

idem for psiA. Defaults to [0, 2 pi]. Only used in likfitlikfit.

idem for tausq.rel. Defaults to [0, +Inf] Lower and upper limits for parameter values can be individually specified. For example, including the following in the function call in likfit or variofit:
limits = pars.limits(phi=c(0, 10), lambda=c(-2.5, 2.5)),
will change the limits for the parameters $\phi and\lambda.Defaultvaluesareusediftheargumentlimitsisnotprovided.$
A list of a 2 elements vector with limits for each parameters likfitlikfit, variofit-variofit pars.limits(phi=c(0,10)) pars.limits(phi=c(0,10), sigmasq=c(0, 100)) plot.geodataExploratory Geostatistical Plotsplot.geodata spatialplot.geodata dplotplot.geodata This function produces a 2 $\times 2 displaywiththefollowingplots:$
$thefirstindicatesthespatiallocationsassigndifferentcolorstodataindifferentquartiles, thenexttwoshowsdataag$
$Dplotwithspatiallocationsandassociateddatavalues.$

```
## S3 method for class 'geodata'
plot(x, coords=x$coords, data = x$data,
             borders, trend="cte", lambda = 1, col.data = 1,
             weights.divide = "units.m", lowess = FALSE, scatter3d = FALSE,
             density = TRUE, rug = TRUE, qt.col, ...)
```

a list containing elements coords and data described next. Typically an object of the class "geodata" - a geoR data-set. If not provided the arguments coords and data must be provided instead.

an n $\times 2 matrixcontainingineachrowEuclideancoordinatesofthendatalocations.Bydefaultittakestheelementcoor$

If an n $\times 2 matrixordata-framewiththebordersoftheareaisprovided, thebordersareincludedinthefirstplot.Bydef$
$"cte"(constantmean-defaultoption), "1st"(afirstorderpolynomialonthecoordinates), "2nd"(asecondorderpolyn$

value of the Box-Cox transformation parameter. Two particular cases are $\lambda = 1 whichcorrespondstonotransformationand\lambda = 0 correspondingtothelog-transformation.indicatesthecolumnnumberforthedatatobeplotted.Onlyvalidifmorethanonedata-setisavailablei.e., iftheargumentdataisamatrix.$

if a vector of weights with the same length as the data is provided each data is divided by the corresponding element in this vector. Defaults divides the data by the element units.m in the data object, if present, otherwise no action is taken and original data is used. The usage of units.m is common for data objects to be analysed using the package geoRglm.

logical. Indicates whether the function lowesslowess should be used in the plots of the data against the coordinates.

logical. If TRUE the last plot is produced by scatterplot3dscatterplot3d showing a 3d plot with data locations and corresponding values.

logical. If TRUE (default) a line with density estimation is added to the histogram.

logical. If TRUE a rug plot is added to the histogram.

colors for the quartiles in the first plot. If missing defaults to blue, green, yellow and red.

further arguments to be passed to the function histhist or scatterplot3dscatterplot3d. A plot is produced on the graphics device. No values are returned. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package

geoR can be found at:
http://www.leg.ufpr.br/geoR. points.geodatapoints.geodata, scatterplot3dscatterplot3d, lowesslowess, densitydensity, rugrug. require(geoR) plot(s100) plot(s100, scatter3d=TRUE) plot(s100, qt.col=1)

plot(ca20, bor=borders) original data plot(ca20, trend= altitude+area) residuals from an external trend plot(ca20, trend='1st') residuals from a polynomial trend

plot(sic.100, bor=sic.borders) original data plot(sic.100, bor=sic.borders, lambda=0) logarithm of the data plot.grfPlots Variograms for Simulated Dataplot.grf spatialplot.grf dplotplot.grf This function plots variograms for simulated geostatistical data generated by the function grfgrf.

```
## S3 method for class 'grf'
plot(x, model.line = TRUE, plot.locations = FALSE, ...)
```

an object of the class grf, typically an output of the function grfgrf.

logical. If TRUE the true variogram model is added to the plot with the sample variogram(s).

logical. If TRUE a plot with data locations is also shown.

further arguments to be passed to the functions variogvariog and plotplot. A plot with the empirical variogram(s) is produced on the output device. No values are returned. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. grfgrf for simulation of Gaussian random fields, plot.variogramplot.variogram for plotting empirical variogram, variogvariog for computation of empirical variograms and plotplot for the generic plotting function. op ¡- par(no.readonly = TRUE) par(mfrow=c(2,1)) sim1 ¡- grf(100, cov.pars=c(10, .25)) generates simulated data plot(sim1, plot.locations = TRUE) plots the locations and the sample true variogram model par(mfrow=c(1,1)) sim2 ¡- grf(100, cov.pars=c(10, .25), nsim=10) generates 10 simulated data plot(sim1) plots sample variograms for all simulations with the true model par(op) plot.krige.bayesPlots Prior and/or Posterior Distributionsplot.krige.bayes spatialplot.krige.bayes dplotplot.krige.bayes aplotplot.krige.bayes Produces plots the priors and posteriors distribuitions for the paramters phi and tausq.rel based on results returned by krige.bayeskrige.bayes.

```
## S3 method for class 'krige.bayes'
plot(x, phi.dist = TRUE, tausq.rel.dist = TRUE, add = FALSE,
                 type=c("bars", "h", "l", "b", "o", "p"), thin, ...)
```

an object of the class krige.bayes, with an output of the funtions krige.bayeskrige.bayes.

logical indicating whether or not plot the distributions for this parameter.

logical indicating whether or not plot the distributions for this parameter.

logical. If TRUE plots is added to current one.

indicates the type of plot. Option "bars" uses the function barplotbarplot and the others uses matplotmatplot.

a numerical vector defining the thining for values of the parameters phi and tausq.rel respectively. This improves visualisation when there are many values in the discrete distribution of the parameters.

further arguments for the plotting function. For plot.krige.bayes a plot is produced or added to the current graphics device. No values are returned.
Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. krige.bayeskrige.bayes, barplotbarplot, matplotmatplot. See documentation for krige.bayes plot.proflikPlots Profile Likelihoodsplot.proflik .proflik.plot.aux1plot.proflik.proflik.plot.aux1 spatialplot.proflik dplot-plot.proflik This function produces plots of the profile likelihoods computed by the function proflikproflik.

```
## S3 method for class 'proflik'
plot(x, pages = c("user", "one", "two"), uni.only, bi.only,
             type.bi = c("contour", "persp"), conf.int = c(0.90, 0.95),
             yaxis.lims = c("conf.int", "as.computed"),
             by.col = TRUE, log.scale = FALSE, use.splines = TRUE,
             par.mar.persp = c(0, 0, 0, 0), ask = FALSE, ...)
```

an object of the class proflik, typically an output of the function proflikproflik.

specify how the plots will be arranged in the graphics device. The default option, "user", uses the current graphical parameters. The option "one" places all the profiles in the same page and the option "two" places the univariate profiles in one page and the bivariate profiles in a second page.

only 1-D profiles are plotted even if the object contains results about the 2-D profiles.

only 2-D profile are plotted even if the object contains results about the 1-D profiles.

Type of plot for the 2-D profiles. Options are "contour" for contour plot (the default) and "persp" for perspective plot.

a vector with numbers in the interval [0,1] specifying levels of the (approximated) confidence intervals. Defaults corresponds to the levels 90% and 95%.

defines the lower limits for the y-axis in the 1-D plots. If "conf.int" the limit is determined by the level of the confidence interval (the default) otherwise will be determined by the smallest computed value.

logical, If TRUE the plots are arranged by columns in a multiple graphics device.

plots the x-axis in the logarithmic scale. Defaults to FALSE.

logical. If TRUE (the default) the function splinespline is used to interpolate between the points computed by proflik.

graphical parameters to be used with persppersp plots. For more details see parpar.

logical. Defines whether or not the user is prompted before each plot is produced.

additional arguments to be passed to the functions plotplot, contourcontour and/or persppersp. Produces plots with the profile likelihoods on the current graphics device. No values are returned. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. proflikproflik for computation of the profile likelihoods. For the generic plotting functions see plotplot, contourcontour, persppersp. See splinespline for interpolation. see examples in the documentation for the function proflik() plot.variog4Plot Directional Variogramsplot.variog4 spatialplot.variog4 dplotplot.variog4 This function plot directional variograms computed by the function variog4variog4. The omnidirectional variogram can be also included in the plot.

```
## S3 method for class 'variog4'
plot(x, omnidirectional=FALSE, same.plot=TRUE, legend = TRUE, ...)
```

an object of the class variog4, typically an output of the function variog4variog4.

logical. Indicates whether the omnidirectional variogram is included in the plot.

logical. Indicates whether the directional variograms are plotted in the same or separated plots.

logical indicating whether legends are automatically included in the plots.

further arguments to be passed to the function plotplot. Typical arguments are col, lty, lwd. For same.plot = TRUE the arguments are passed to the function matplotmatplot which is used to produce the plot. A plot is produced on the output device. No values returned. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br, Peter J. Diggle p.diggle@lancaster.ac.uk. Further information about the geoR package can be found at:
http://www.leg.ufpr.br/geoR. variog4variog4 for variogram calculations and matplotmatplot for multiple lines plotting. s100.v4 ¡- variog4(s100, max.dist=1) Plotting variograms for the four directions plot(s100.v4) changing plot options plot(s100.v4, lwd=2) plot(s100.v4, lty=1, col=c("darkorange", "darkblue", "darkgreen","darkviolet")) plot(s100.v4, lty=1, lwd=2) including the omnidirectional variogram plot(s100.v4, omni=TRUE) variograms on different plots plot(s100.v4, omni=TRUE, same=FALSE) plot.variogramPlot Empirical Variogramplot.variogram spatialplot.variogram dplotplot.variogram Plots sample (empirical) variogram computed using the function variogvariog.

```
## S3 method for class 'variogram'
plot(x, max.dist, vario.col = "all", scaled = FALSE,
               var.lines = FALSE, envelope.obj = NULL,
               pts.range.cex, bin.cloud = FALSE,  ...)
```

an object of the class "variogram", typically an output of the function variogvariog.

maximum distance for the x-axis. The default is the maximum distance for which the sample variogram was computed.

only used if obj has information on more than one empirical variogram. The default "all" indicates that variograms of all variables should be plotted. Alternativelly a numerical vector can be used to select variables.

If TRUE the variogram values are divided by the sample variance. This allows comparison of variograms of variables measured in different scales.

If TRUE a horizontal line is drawn at the value of the variance of the data (if scaled = F) or at 1 (if scaled = T).

adds a variogram envelope computed by the function variog.model.envvariog.model.env or variog.mc.envvariog.mc.env.

optional. A two elements vector with maximum and minimum values for the caracter expansion factor cex. If provided the point sizes in binned variogram are proportional to the number of pairs of points used to compute each bin.

logical. If TRUE and the sample variogram was computed with the option bin.cloud = TRUE, box-plots of values at each bin are plotted instead of the empirical variograms.

other arguments to be passed to the function plotplot or matplotmatplot
This function plots empirical variograms. Toghether with lines.variogramlines.variogram can be used to compare sample variograms of different variables and to compare variogram models against the empirical variogram.

It uses the function matplotmatplot when plotting variograms for more them one variable. Produces a plot with the sample variogram on the current graphics device. No values are returned. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. variogvariog for variogram calculations, lines.variogramlines.variogram and lines.variomodellines.variomodel for adding lines to the current plot, variog.model.envvariog.model.env and variog.mc.envvariog.mc.env for variogram envelops computation, matplotmatplot for multiple lines plot and plotplot for generic plot function. op ¡- par(no.readonly = TRUE) sim ¡- grf(100, cov.pars=c(1, .2)) simulates data vario ¡- variog(sim, max.dist=1) computes sample variogram par(mfrow=c(2,2)) plot(vario) the sample variogram plot(vario, scaled = TRUE) the scaled sample variogram plot(vario, max.dist = 1) limiting the maximum distance plot(vario, pts.range = c(1,3)) points sizes proportional to number of pairs par(op) plot.xvalidPlot Cross-Validation Resultsplot.xvalid spatialplot.xvalid dplotplot.xvalid This function produces ten plots with the results produced by the cross-validation function xvalidxvalid.

```
## S3 method for class 'xvalid'
plot(x, coords, borders = NULL, ask = TRUE,
            error = TRUE, std.error = TRUE, data.predicted = TRUE,
            pp = TRUE, map = TRUE, histogram = TRUE,
            error.predicted = TRUE, error.data = TRUE, ...)
```

an object of the class "xvalid", typically an output from the function xvalidxvalid.

an n $\times 2 object containing coordinates of the (cross-) validation locations. optional. Takes a two column matrix or data-frame with coordinates of the borders. If provided the borders are included in the errors maps.$

logical. Defines whether or not the user is prompted before each plot is produced.

logical. Defines whether the plots for the errors (error = data - predicted) will be produced.

logical. Defines whether the plots for the standardised errors will be produced.

logical defining whether a plot of data versus predicted should be displayed. Defaults to TRUE.

logical defining whether a *pp* plot should be displayed. Defaults to TRUE.

logical defining whether a map of the errors should be displayed. Defaults to TRUE.

logical defining whether a histogram of the errors should be displayed. Defaults to TRUE.

logical defining whether a plot of errors versus predicted should be displayed. Defaults to TRUE.

logical defining whether a plot of errors versus data should be displayed. Defaults to TRUE.

other arguments to be passed to the function plotplot.
The number of plots to be produced will depend on the input options. If the graphics device is set to just one plot (something equivalent to par(mfcol=c(1,1))) after each graphic being displayed the user will be prompt to press ¡return¿ to see the next graphic.

Alternativaly the user can set the graphical parameter to have several plots in one page. With default options for the arguments the maximum number of plots (10) is produced and setting par(mfcol=c(5,2))) will display them in the same page.

The “errors” for the plots are defined as error = data - predicted and the plots uses the color blue to indicate positive errors and red to indicate negative erros. No value returned. Plots are produced on the current graphics device. xvalidxvalid for the cross-validation computations. wls ¡- variofit(variog(s100, max.dist = 1), ini = c(.5, .5), fix.n = TRUE) xvl ¡- xvalid(s100, model = wls) op ¡- par(no.readonly = TRUE) par(mfcol = c(3,2)) par(mar = c(3,3,0,1)) par(mgp = c(2,1,0)) plot(xvl, error = FALSE, ask = FALSE) plot(xvl, std.err = FALSE, ask = FALSE) par(op) points.geodataPlots Spatial Locations and Data Valuespoints.geodata spatialpoints.geodata dplotpoints.geodata aplotpoints.geodata This function produces a plot with points indicating the data locations. Arguments can control the points sizes, patterns and colors. These can be set to be proportional to data values, ranks or quantiles. Alternatively, points can be added to the current plot.

```
## S3 method for class 'geodata'
points(x, coords=x$coords, data=x$data, data.col = 1, borders,
               pt.divide=c("data.proportional","rank.proportional",
                           "quintiles", "quartiles", "deciles", "equal"),
               lambda = 1, trend = "cte", abs.residuals = FALSE,
               weights.divide = "units.m", cex.min, cex.max, cex.var,
               pch.seq, col.seq, add.to.plot = FALSE,
               x.leg, y.leg = NULL, dig.leg = 2,
               round.quantiles = FALSE, permute = FALSE, ...)
```

a list containing elements coords and data described next. Typically an object of the class "geodata" - a geoR data-set. If not provided the arguments coords and data must be provided instead.

an n $\times 2 matrixcontainingcoordinatesofthendatalocationsineachrow.Defaultstogeodata\$coords.avectorormatrix$

the number of the data column. Only used if data is a matrix with columns corresponding to different variables or simulations.

If an n $\times 2 matrixordata-framewiththecoordinatesofthebordersoftheregionsisprovided, thebordersareaddedtoth$

specifies the mean part of the model. The options are: "cte" (constant mean - default option), "1st" (a first order polynomial on the coordinates), "2nd" (a second order polynomial on the coordinates), or a formula of the type ˜X where X is a matrix with the covariates (external trend). If provided the trend is "removed" using the function lmlm and the residuals are plotted.

logical. If TRUE and the value passed to the argument trend is different from "cte" the point sizes are proportional to absolute values of the residuals.

value of the Box-Cox transformation parameter. Two particular cases are $\lambda = 1 whichcorrespondstonotransformationand \lambda = 0 correspondingtothelog - transformation.ifavectorofweightswiththesamelengthasthedataisprovidedeachdataisdividedbythecorrespondin$

minimum value for the graphical parameter cex. This value defines the size of the point corresponding the minimum of the data. Defaults to 0.5.

maximum value for the graphical parameter cex. This value defines the size of the point corresponding the maximum of the data. If pt.divide = "equal" it is used to set the value for the graphical parameter cex. Defaults to 1.5.

a numeric vector with the values of a variable defining the size of the points. Particularly useful for displaying 2 variables at once.

number(s) defining the graphical parameter pch.

number(s) defining the colors in the graphical parameter col.

logical. If TRUE the points are added to the current plot or image otherwise a display is open. Defaults to FALSE.

x and y location of the legend as documented in legendlegend.

the desired number of digits after the decimal point. Printing values in the legend uses formatCformatC with argument format = "f".

logical. Defines whether or not the values of the quantiles should be rounded. Defaults to FALSE.

logical indication whether the data values should be randomly re-alocatted to the coordinates. See DETAILS below.

further arguments to be passed to the function plotplot, if add.to.plot = FALSE; or to the function pointspoints, if add.to.plot = TRUE. The points can be devided in categories and have different sizes and/or colours according to the argument pt.divide. The options are:

"data.proportional" sizes proportional to the data values.

"rank.proportional" sizes proportional to the rank of the data.

"quintiles" five different sizes according to the quintiles of the data.

"quartiles" four different sizes according to the quartiles of the data.

"deciles" ten different sizes according to the deciles of the data.

"equal" all points with the same size.

a scalar defines a number of quantiles, the number provided defines the number of different points sizes and colors.

a numerical vector with quantiles and length ¿ 1 the values in the vector will be used by the function cutcut as break points to divide the data in classes.

For cases where points have different sizes the arguments cex.min and cex.max set the minimum and the maximum point sizes. Additionally, pch.seq can set different patterns for the points and col.seq can be used to define colors. For example, different colors can be used for quartiles, quintiles and deciles while a sequence of gray tones (or a color sequence) can be used for point sizes proportional to the data or their ranks. For more details see the section EXAMPLES.

The argument cex.var allows for displaying 2 variables at once. In this case one variable defines the backgroung colour of the points and the other defines the points size.

The argument permute if set to TRUE randomly realocates the data in the coordinates. This may be used to contrast the spatial pattern of original data against another situation where there is no spatial dependence (when setting permute = TRUE). If a trend is provided the residuals (and not the original data) are permuted. A plot is created or points are added to the current graphics device.
A list with graphical parameters used to produce the plot is returned invisibily. According to the input options, the list has some or all of the following components:

the values of the quantiles used to divide the data.

the values of the graphics expansion parameter cex.

the values of the graphics color parameter col.

the values of the graphics pattern parameter pch.
Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. plot.geodataplot.geodata for another display of the data and pointspoints and plotplot for information on the generic functions. The documentation of parpar provides details on graphical parameters. For color schemes in see graygray and rainbowrainbow. op ¡- par(no.readonly = TRUE) par(mfrow=c(2,2), mar=c(3,3,1,1), mgp = c(2,1,0)) points(s100, xlab="Coord X", ylab="Coord Y") points(s100, xlab="Coord X", ylab="Coord Y", pt.divide="rank.prop") points(s100, xlab="Coord X", ylab="Coord Y", cex.max=1.7, col=gray(seq(1, 0.1, l=100)), pt.divide="equal") points(s100, pt.divide="quintile", xlab="Coord X", ylab="Coord Y") par(op)

points(ca20, pt.div='quartile', x.leg=4900, y.leg=5850, bor=borders) par(mfrow=c(1,2), mar=c(3,3,1,1), mgp = c(2,1,0)) points(s100, main="Original data") points(s100, permute=TRUE, main="Permuting locations")

Now an example using 2 variable, 1 defining the gray scale and the other the points size points.geodata(coords=camg[,1:2], data=camg[,3], col="gray", cex.var=camg[,5]) points.geodata(coords=camg[,1:2], data=camg[,3], col="gray", cex.var=camg[,5], pt.div="quint") polygridCoordinates of Points Inside a Polygon polygrid spatialpolygrid This function builds a rectangular grid and extracts points which are inside of an internal polygonal region.

```
polygrid(xgrid, ygrid, borders, vec.inout = FALSE, ...)
```

grid values in the x-direction.

grid values in the y-direction.

a matrix with polygon coordinates defining the borders of the region.

logical. If TRUE a logical vector is included in the output indicating whether each point of the grid is inside the polygon. Defaults to FALSE.

currently not used (kept for back compatibility). The function works as follows: First it creates a grid using the function expand.gridexpand.grid and then it uses the geoR' internal function .geoR_inout() which wraps usage of SpatialPointsSpatialPoints and overlayoverlay from the package sp to extract the points of the grid which are inside the polygon.

Within the package geoR this function is typically used to select points in a non-rectangular region to perform spatial prediction using krige.bayeskrige.bayes, krige.convkrige.conv or kslineksline. It is also useful to produce image or perspective plots of the prediction results. A list with components:

an n $\times 2 matrixwiththecoordinatesofthepointsinsidethepolygon.logical, avectorindicatingwhethereachpointofthe$
Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. pred_gridpred.Rul.grid, expand.gridexpand.grid, overlayoverlay, SpatialPointsSpatialPoints. poly ¡- matrix(c(.2, .8, .7, .1, .2, .1, .2, .7, .7, .1), ncol=2) plot(0:1, 0:1, type="n") lines(poly) poly.in ¡- polygrid(seq(0,1,l=11), seq(0,1,l=11), poly, vec=TRUE) points(poly.inxy) practicalRangePratical range for correlation functionspracticalRange spatialpracticalRange Computes practical ranges for the correlation functions implemented in the geoR package

```
practicalRange(cov.model, phi, kappa = 0.5, correlation = 0.05, ...)
```

correlation model as documented in cov.spatialcov.spatial.

correlation parameter as documented in cov.spatialcov.spatial.

additional correlation parameter as documented in cov.spatialcov.spatial.

correlation threshold for asymptotic models. Defaults to 0.05.

arguments to be passed to optimiseoptimise. A scalar with the value of the practical range. cov.spatialcov.spatial practicalRange(”exp”, phi=10) practicalRange(”sph”, phi=10) practical-Range(”gaus”, phi=10) practicalRange(”matern”, phi=10, kappa=0.5) practicalRange(”matern”, phi=10, kappa=1.5) practicalRange(”matern”, phi=10, kappa=2.5) predict.BGCCMPrediction for the bivariate Gaussian common component geostatistical modelpredict.BGCCM spatialpredict.BGCCM Performs prediction for the bivariate Gaussian common component geostatistical model

```
## S3 method for class 'BGCCM'
predict(object, locations, borders,
              variable.to.predict = 1, ...)
```

on object of the class BGCCMfit, which is an output of likfitBGCCMlikfit-BGCCM.

an N $\times 2 matrixordata - framewiththe2 - Dcoordinatesof theNpredictionlocations, oralistforwhichthefirsttwocomponentsareused.Inputisint$

scalar with options for values or 2 indicating which variable is to be predicted.

not yet used. A list of the class BGCCMpred with components:

predicted values.

prediction variances. Warning This is a new function and still in draft format and pretty much untested. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br, Peter J. Diggle p.diggle@lancaster.ac.uk. likfitBGCCMlikfitBGCCM see http://www.leg.ufpr.br/geoR/tutorials/CCM.R pred_gridGenerates a 2D Prediction Grid pred.Rul.grid spatialpred_grid This function facilitates the generation of a 2D prediction grid for geostatistical kriging.

```
pred_grid(coords, y.coords = NULL, ..., y.by = NULL,
          y.length.out = NULL, y.along.with = NULL)
```

a list, matrix or data-frame with xy-coordinates of prediction points or a vector with x-coordinates.

a vector with y-coordinates. Needed if argument coords provides only x-coordinates.

arguments by or length.out to be passed to the function reprep. These arguments are used for the x-coordinates and are default optioons for y-coordinates.

Optional. by argument for reprep to be used with the y-coordinates.

Optional. length.out argument for reprep to be used with the y-coordinates.

Optional. along.with argument for reprep to be used with the y-coordinates.
An two column data-frame which is on output of expand.gridexpand.grid. See seqseq and expand.gridexpand.grid which are used internally and locations.insidelocations.inside and polygridpolygrid to select points inside a border. pred$_grid(c(0,1), c(0,1), by = 0.25) createagridinaunitsquareloc0 < -pred_grid(ca20$borders, by=20) points(ca20, borders=borders) points(loc0, pch=”+”) points(locations.inside(loc0, ca20$border), pch = " + ", col =$ 2) print.BGCCMPrints an summary of of the output from likfitBGCCM. print.BGCCM spatialprint.BGCCM Prints a short version of an object of the class BGCCM.

```
## S3 method for class 'BGCCM'
print(x, ...)
```

an object of the class BGCCM.

arguments to be passed to formatformat. formatformat for options to format the output. proflikComputes Profile Likelihoodsproflik .proflik.aux0proflik.proflik.aux0 .proflik.aux1proflik.proflik.aux1 .proflik.aux1.1proflik.proflik.aux1.1 .proflik.aux10proflik.proflik.aux10 .proflik.aux11proflik.proflik.aux11 .proflik.aux12proflik.proflik.aux12 .proflik.aux13proflik.proflik.aux13 .proflik.aux14proflik.proflik.aux14 .proflik.aux15proflik.proflik.aux15 .proflik.aux16proflik.proflik.aux16 .proflik.aux17proflik.proflik.aux17 .proflik.aux18proflik.proflik.aux18 .proflik.aux19proflik.proflik.aux19 .proflik.aux2proflik.proflik.aux2 .proflik.aux20proflik.proflik.aux20 .proflik.aux21proflik.proflik.aux21 .proflik.aux21.1proflik.proflik.aux21.1 .proflik.aux22proflik.proflik.aux22 .proflik.aux23proflik.proflik.aux23 .proflik.aux24proflik.proflik.aux24 .proflik.aux27proflik.proflik.aux27 .proflik.aux28proflik.proflik.aux28 .proflik.aux3proflik.proflik.aux3 .proflik.aux30proflik.proflik.aux30 .proflik.aux31proflik.proflik.aux31 .proflik.aux32proflik.proflik.aux32 .proflik.aux33proflik.proflik.aux33 .proflik.aux4proflik.proflik.aux4 .proflik.aux5proflik.proflik.aux5 .proflik.aux6proflik.proflik.aux6 .proflik.aux7proflik.proflik.aux7 .proflik.aux8proflik.proflik.aux8 .proflik.aux9proflik.proflik.aux9 .proflik.covproflik.proflik.cov .proflik.lambdaproflik.proflik.lambda .proflik.mainproflik.proflik.main spatialproflik Computes profile likelihoods for model parameters previously estimated using the function likfitlikfit.

```
proflik(obj.likfit, geodata, coords = geodata$coords,
        data = geodata$data, sill.values, range.values,
        nugget.values, nugget.rel.values, lambda.values,
        sillrange.values = TRUE, sillnugget.values = TRUE,
        rangenugget.values = TRUE, sillnugget.rel.values = FALSE,
        rangenugget.rel.values = FALSE, silllambda.values = FALSE,
        rangelambda.values = TRUE,  nuggetlambda.values = FALSE,
        nugget.rellambda.values = FALSE,
        uni.only = TRUE, bi.only = FALSE, messages, ...)
```

an object of the class likfit, typically an output of the function likfitlikfit.

a list containing elements coords and data described next. Typically an object of the class "geodata" - a geoR data-set. If not provided the arguments coords and data must be provided instead.

an n $\times 2 matrix containing in each row Euclidean coordinates of the n data locations. By default it takes the element coor$

set of values of the partial sill parameter $\sigma^2 for which the profile likelihood will be computed. set of values of the range parameter \phi for which the profile likelihoo$

set of values of the nugget parameter $\tau^2 for which the profile likelihood will be computed. Only used if the model was fitt$

set of values of the Box-Cox transformation parameter $\lambda for which the profile likelihood will be computed. Only to be used if the model was fitted using the function likfitlikfitu$
$D profile likelihood should be computed. Only valid if uni.only = FALSE.$

as above.

as above.

as above.

as above.

as above.

as above.

as above.

as above.

as above.

as above.

logical. Indicates whether status messages should be printed on the screen (i.e. current output device) while the function is running.

additional parameters to be passed to the minimization function. The functions .proflik.* are auxiliary functions used to compute the profile likelihoods. These functions are internally called by the minimization functions when estimating the model parameters. An object of the class "proflik" which is a list. Each element contains values of a parameter (or a pair of parameters for 2-D profiles) and the corresponding value of the profile likelihood. The components of the output will vary according to the input options.

1. Profile likelihoods for Gaussian Random Fields are usually uni-modal. Unusual or jagged shapes can be due to the lack of the convergence in the numerical minimization for particular values of the parameter(s). If this is the case it might be necessary to pass control arguments to the minimization functions using the argument .... It's also advisable to try the different options for the minimisation.function argument. See documentation of the functions optimoptim and/or nlmnlm for further details.

2. 2-D profiles can be computed by setting the argument uni.only = FALSE. However, before computing 2-D profiles be sure they are really necessary. Their computation can be time demanding since it is performed on a grid determined by the cross-product of the values defining the 1-D profiles.

3. There is no "default strategy" to find reasonable values for the x-axis. They must be found in a "try-and-error" exercise. It's recommended to use short sequences in the initial attempts. The EXAMPLE section below illustrates this.

Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. plot.proflikplot.proflik for graphical output, likfitlikfit for the parameter estimation, optimoptim and nlmnlm for further details about the minimization functions. op ¡- par(no.readonly=TRUE) ml ¡- likfit(s100, ini=c(.5, .5), fix.nug=TRUE) a first atempt to find reasonable values for the x-axis: prof ¡- proflik(ml, s100, sill.values=seq(0.5, 1.5, l=4), range.val=seq(0.1, .5, l=4)) par(mfrow=c(1,2)) plot(prof) a nicer setting Not run: prof ¡- proflik(ml, s100, sill.values=seq(0.45, 2, l=11), range.val=seq(0.1, .55, l=11)) plot(prof) to include 2-D profiles use: (commented because

this is time demanding) prof ¡- proflik(ml, s100, sill.values=seq(0.45, 2, l=11), range.val=seq(0.1, .55, l=11), uni.only=FALSE) par(mfrow=c(2,2)) plot(prof, nlevels=16)

End(Not run) par(op) read.geodataReads and Converts Data to geoR Formatread.geodata spatialread.geodata manipread.geodata Reads data from a *ASCII* file and converts it to an object of the classclass geodata, the standard data format for the geoR package.

```
read.geodata(file, header = FALSE, coords.col = 1:2, data.col = 3,
             data.names = NULL, covar.col = NULL, covar.names = "header",
             units.m.col = NULL, realisations = NULL,
             na.action = c("ifany", "ifdata", "ifcovar", "none"),
             rep.data.action, rep.covar.action, rep.units.action, ...)
```

a string with the name of the *ASCII* file.

logical. Indicates whether the variables names should be read from the first line of the input file.

a vector with the numbers of the columns containing the coordinates.

a scalar or vector with the number of the column(s) containing the data.

a string or vector of strings with names for the data columns. Only valid if there is more than one column of data. By default the names in the original object are used.

optional. A scalar or vector with the number of the column(s) with the values of the covariate(s).

optional. A vector with the names of the the covariates. By default the names in the original object are used.

optional. A scalar with the column number corresponding to the offset variable. Alternativelly can be a character vector with the name of the offset. This option is particularly relevant when using the package geoRglm.

optional. A vector indicating the replication number. For more details see documentation for as.geodataas.geodata.

a string. Defines action to be taken in the presence of NA's. For more details see documentation for as.geodataas.geodata.

a string or a function. Defines action to be taken when there is more than one data at the same location. For more details see documentation for as.geodataas.geodata.

a string or a function. Defines action to be taken when there is more than one covariate at the same location. For more details see documentation for as.geodataas.geodata.

a string or a function. Defines action to be taken on the element units.m, if present when there is more than one data at the same location. The default option is the same value set for rep.data.action.

further arguments to be passed to the function read.tableread.table. The function read.tableread.table is used to read the data from the *ASCII* file and then as.geodataas.geodata is used to convert to an object of the classclass geodata. An object of the classclass geodata. See documentation for the function

as.geodataas.geodata for further details. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. as.geodataas.geodata to convert existing objects, read.tableread.table, the basic function used to read *ASCII* files, and listlist for detailed information about lists. s100 and s121Simulated Data-Sets which Illustrate the Usage of the Package geoR s100 and s121 s100s100 and s121s100 s121s100 and s121s121 datasetss100 and s121 These two simulated data sets are the ones used in the Technical Report which describes the package geoR (see reference below). These data-sets are used in several examples throughout the package documentation.

```
data(s100)

data(s121)
```

Two objects of the classclass geodata. Both are lists with the following components:

coords the coordinates of data locations.

data the simulated data. Notice that for s121 this a 121 $\times 10 matrix with 10 simulations. the correlation model.$

cov.modelnugget the values of the nugget parameter.

cov.pars the covariance parameters.

kappa the value of the parameter *kappa*.

lambda the value of the parameter *lambda*.

Ribeiro Jr, P.J. and Diggle, P.J. (1999) geoS: A geostatistical library for S-PLUS. *Technical report ST-99-09, Dept of Maths and Stats, Lancaster University.*

Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. plot(s100) plot(s121, type="l") s256iSimulated Data-Set which Illustrate the Usage of krige.bayes s256i datasetss256i This is the simulated data-set used in the Technical Report which describes the implementation of the function krige.bayes (see reference below).

```
data(s256i)
```

Two objects of the classclass geodata. Both are lists with the following components:

coords the coordinates of data locations.

data the simulated data.

Ribeiro Jr, P.J. and Diggle, P.J. (1999) Bayesian inference in Gaussian model-based geostatistics. *Technical report ST-99-08, Dept of Maths and Stats, Lancaster University.*

Further information about the geoR package can be found at:
http://www.leg.ufpr.br/geoR. points(s256i, pt.div="quintiles", cex.min=1,

cex.max=1) sample.geodataSampling from geodata objectssample.geodata spatialsample.geodata manipsample.geodata This functions facilitates extracting samples from geodata objects.

```
sample.geodata(x, size, replace = FALSE, prob = NULL, coef.logCox,
               external)
```

an object of the class geodata.

non-negative integer giving the number of items to choose.

Should sampling be with replacement?

A vector of probability weights for obtaining the elements of the data points being sampled.

optional. A scalar with the coeficient for the log-Cox process. See DETAILS below.

numeric values of a random field to be used in the log-Cox inhomogeneous poisson process. If prob=NULL and the argument coef.logCox, is provided, sampling follows a log-Cox proccess, i.e. the probability of each point being sampled is proportional to: exp(b Y(x)) with b given by the value passed to the argument coef.logCox and Y(x) taking values passed to the argument external or, if this is missing, the element data of the geodata object. Therefore, the latter generates a preferential sampling. a list which is an object of the class geodata. as.geodataas.geodata, sample-sample. par(mfrow=c(1,2)) S1 ¡- grf(2500, grid="reg", cov.pars=c(1, .23)) image(S1, col=gray(seq(0.9,0.1,l=100))) y1 ¡- sample.geodata(S1, 80) points(y1$coords, pch = 19)Nowapreferentialsamplingy2 < -sample.geodata(S1, 80, coef = 1.3)whichisequivalenttoppsy2 < -sample.geodata(S1, 80, prob = exp(1.3 * S1$data)) points(y2$coords, pch = 19, col = 2)andnowaclustered(butnotpreferential)S2 < -grf(2500, grid = "reg", cov.pars = c(1, .23))y3 < -sample.geodata(S1, 80, prob = exp(1.3 * S2$data)) which is equivalent to points(y3$coords, pch = 19, col = 4)image(S2, col = gray(seq(0.9, 0.1, l = 100)))points(y3$coords, pch=19, col=4) sample.posteriorSamples from the posterior distributionsample.posterior spatialsample.posterior distributionsample.posterior Sample quadruples $(\beta, \sigma^2, \phi, \tau^2_{rel})fromtheposteriordistributionreturnedbykrige.bayeskrige.bayes.$

```
sample.posterior(n, kb.obj)
```

number of samples

on object with an output of krige.bayeskrige.bayes. A n $\times 4 data - framewithsamplesfromtheposteriordistributionofthemodelparameters.$
Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. krige.bayeskrige.bayes and sample.posteriorsample.posterior. sample.priorSample the prior distributionsample.prior spatialsample.prior distributionsample.prior Sample quadruples $(\beta, \sigma^2, \phi, \tau^2_{rel})fromthepriordistributionofparametersspecifyingaGaussianrandomfield.Typicallythepriorisspec$

```
sample.prior(n, kb.obj=NULL, prior=prior.control())
```

number of samples

on object with an output of krige.bayeskrige.bayes.

an call to prior.controlprior.control. Unnecessary if kb.obj is provided. A p+3 $\times 4 data - frame with a sample of the prior distribution of model parameters, where p is the length of the mean parame$ Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. krige.bayeskrige.bayes and sample.posteriorsample.posterior. sample.prior(50, prior=prior.control(beta.prior = ”normal”, beta = .5, beta.var.std=0.1, sigmasq.prior=”sc”, sigmasq=1.2, df.sigmasq= 2, phi.prior=”rec”, phi.discrete = seq(0,2, l=21))) set.coords.limsSets Limits to Scale Plotsset.coords.lims spatialset.coords.lims This is an function typically called by functions in the package geoR to set limits for the axis when plotting spatial data.

```
set.coords.lims(coords, borders = coords, xlim, ylim, ...)
```

an n $\times 2 matrix with coordinates. an n \times 2 matrix with coordinates.$

the ranges to be encompassed by the x and y axes.

not used, included just for internal handling. A 2 $\times 2 matrix with limits for the axis.$ Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. SICSpatial Interpolation Comparison dataSIC sicSICsic sic.100SICsic.100 sic.367SICsic.367 sic.allSICsic.all sic.bordersSICsic.borders sic.someSICsic.some datasetsSIC spatialSIC Data from the SIC-97 project: Spatial Interpolation Comparison.

```
data(SIC)
```

Four objects of the classclass ”geodata”: sic.all, sic.100, sic.367, sic.some. Each is a list with two components:

coords the coordinates of the data locations. The distance are given in kilometers.

data rainfall values. The unit is milimeters.

altitude elevation values. The unit is milimeters.

Additionally an matrix sic.borders with the borders of the country. Data from the project *Spatial Interpolation Comparison 97*; see ftp://ftp.geog.uwo.ca/SIC97/

Christensen, O.F., Diggle, P.J. and Ribeiro Jr, P.J. (2001) Analysing positive-valued spatial data: the transformed Gaussian model. In Monestiez, P., Allard, D. and Froidevaux (eds), GeoENV III - Geostatistics for environmental applications. Quantitative Geology and Geostatistics, Kluwer Series, 11, 287–298.
points(sic.100, borders=sic.borders) points(sic.all, borders=sic.borders) soil250Soil chemistry properties data setsoil250 datasetssoil250 spatialsoil250 Several soil chemistry properties measured on a regular grid with 10x25 points spaced by 5 meters.

`data(soil250)`

A data frame with 250 observations on the following 22 variables.

Linha x-coordinate

Coluna y-coordinate

Cota elevation

AGrossa a numeric vecto, sand portion of the sample.

Silte a numeric vector, silt portion of the sample.

Argila a numeric vector, sand portion of the sample.

pHAgua a numeric vector, soil pH at water

pHKCl a numeric vector, soil pH by KCl

Ca a numeric vector, calcium content

Mg a numeric vector, magnesium content

K a numeric vector, potassio content

Al a numeric vector, aluminium content

H a numeric vector, hidrogen content

C a numeric vector, carbon content

N a numeric vector, nitrogen content

CTC a numeric vector, catium exchange capability

S a numeric vector, enxofrar content

V a numeric vector

M a numeric vector

NC a numeric vector

CEC a numeric vector

CN a numeric vector, carbon/nitrogen relation

Uniformity trial with 250 undisturbed soil samples collected at 25cm soil depth of spacing of 5 meters, resulting on a regular grid of 25 $\times 10 points$.

See also the data-set *wrc* with other variables colected at the same points. Bassoi thesis Bassoi papers data(soil250) ctc ¡- as.geodata(soil250, data.col=16) plot(ctc) soja98Soya bean production and other variables in a uniformity trialsoja98 datasetssoja98 Data on soya bean production in a uniformity trial measured at plots of size 5x5 meters and other soil properties measured in points given by the data coordinates.

`data(soja98)`

A data frame with 256 observations on the following 10 variables.

X a numeric vector with X-coordinates of the plot centres.

Y a numeric vector with X-coordinates of the plot centres.

P a numeric vector, phosphorous content.

PH a numeric vector, soil pH.

K a numeric vector, potassium content. (Cmol/dmˆ3)

MO a numeric vector, organic matter. (percentage)

SB a numeric vector, basis saturation.

iCone a numeric vector, cone index, measuring mechanic resistence of the soil. (kg/cmˆ2)

Rend a numeric vector, total soya production within the plot (kg).

PROD a numeric vector, production converted to productivity by a unit of area - hectare (ton/ha).

Souza, E.G.; Jojann, J. A.; Rocha, J. V.; Ribeiro, S. R. A.; Silva, M. S., Upazo, M. A. U.; Molin, J. P.; Oliveira, E. F.; Nóbrega, L. H. P. (1999) Variabilidade espacial dos atributos químicos do solo em um latossolo roxo distrófico da região de Cascavel-PR. *Engenharia Agrícola*. Jaboticabal, volume 18, nr. 3, p.80-92. data(soja98) plot(soja98) require(geoR) prod98 ¡- as.geodata(soja98, coords.col=1:2, data.col=) plot(prod98, low=TRUE) statistics.predictiveSummary statistics from predictive distributionsstatistics.predictive spatialstatistics.predictive Computes summaries based on simulations of predictive distribution returned by krige.bayeskrige.bayes and krige.convkrige.conv.

```
statistics.predictive(simuls, mean.var = TRUE, quantile, threshold,
                      sim.means, sim.vars)
```

object with simulations from the predictive distribution

Logical. Indicates whether or not to compute mean and variances of the simulations at each location.

defines quantile estimator. See documentation for output.controloutput.control .

defines probability estimator. See documentation for output.controloutput.control.

Logical. Indicates whether or not to compute the mean of of the conditional simulations.

Logical. Indicates whether or not to compute the variances of the conditional simulations. A list with one ore more of the following components.

mean at each prediction location.

variance at each prediction location.

quantiles, at each prediction location.

probabilities, at each prediction location, of been below the provided threshold.

vector with means of each conditional simulation.

vector with variances of each conditional simulation.
Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.legufpr.br/geoR. subareaSelects a Subarea from a Geodata ObjectsubArea spatialsubarea Selects elements of a geodata object wich are within a rectangular (sub)area

```
subarea(geodata, xlim, ylim, ...)
```

an object of the class geodata as defined in as.geodataas.geodata.

optional, a vector with selected range of the x-coordinates.

optional, a vector with selected range of the y-coordinates.

further arguments to be passed to zoom.coordszoom.coords. The function copies the original geodata object and selects values of $coords, $data, $borders, $covariate and $units.m which lies within the selected sub-area. Remaining components of the geodata objects are untouched.

If xlim and/or ylim are not provided the function prompts the user to click 2 points defining an retangle defining the subarea on a existing plot.
Returns an geodata object, subsetting of the original one provided. Paulo Justiniano Ribeiro Jr. paulojus@legufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. zoom.coordszoom.coords, locatorlocator foo ¡- matrix(c(4,6,6,4,2,2,4,4), nc=2) foo1 ¡- zoom.coords(foo, 2) foo1 foo2 ¡- coords2coords(foo, c(6,10), c(6,10)) foo2 plot(1:10, 1:10, type="n") polygon(foo) polygon(foo1, lty=2) polygon(foo2, lwd=2) arrows(foo[,1], foo[,2],foo1[,1],foo1[,2], lty=2) arrows(foo[,1], foo[,2],foo2[,1],foo2[,2]) legend("topleft", c("foo", "foo1 (zoom.coords)", "foo2 (coords2coords)"), lty=c(1,2,1), lwd=c(1,1,2))

"zooming" part of The Gambia map gb ¡- gambia.borders/1000 gd ¡- gambia[,1:2]/1000 plot(gb, ty="l", asp=1, xlab="W-E (kilometres)", ylab="N-S (kilometres)") points(gd, pch=19, cex=0.5) r1b ¡- gb[gb[,1] ¡ 420,] rc1 ¡- rect.coords(r1b, lty=2)

r1bn ¡- zoom.coords(r1b, 1.8, xoff=90, yoff=-90) rc2 ¡- rect.coords(r1bn, xz=1.05) segments(rc1[c(1,3),1],rc1[c(1,3),2],rc2[c(1,3),1],rc2[c(1,3),2], lty=3)

lines(r1bn) r1d ¡- gd[gd[,1] ¡ 420,] r1dn ¡- zoom.coords(r1d, 1.7, xlim.o=range(r1b[,1],na.rm=TRUE), ylim.o=range(r1b[,2],na.rm=TRUE), xoff=90, yoff=-90) points(r1dn, pch=19, cex=0.5) text(450,1340, "Western Region", cex=1.5)

if(require(geoRglm)) data(rongelap) points(rongelap, bor=borders) zooming the western area rongwest ¡- subarea(rongelap, xlim=c(-6300, -4800)) points(rongwest, bor=borders) now zooming in the same plot points(rongelap, bor=borders) rongwest.z ¡- zoom.coords(rongwest, xzoom=3, xoff=2000, yoff=3000) points(rongwest.z, bor=borders, add=TRUE) rect.coords(rongwest$sub, quiet = TRUE)rect.coords(rongwest.z$sub, quiet=TRUE) subset.geodataMethod for subsetting geodata objectssubset.geodata spatialsubset.geodata manipsubset.geodata Subsets a object of the class geodata by transforming it to a data-frame, using subset and back transforming to a geodata object.

```
## S3 method for class 'geodata'
subset(x, ..., other = TRUE)
```

an object of the class geodata.

arguments to be passed to subset.data.framesubset.data.frame.

logical. If TRUE non-standard geodata elements of the original geodata object are copied to the resulting object. A list which is an object of the class geodata. subsetsubset for the generic function and methods and as.geodataas.geodata for more information on geodata objects. subset(ca20, data ¿ 70) subset(ca20, area == 1) summary.geodataSummaries for geodata objectsummary.geodata print.summary.geodatasummary.geodataprint.summary.geodata univarsummary.geodata spatialsummary.geodata Sumarises each of the main elements of an object of the class geodata.

```
## S3 method for class 'geodata'
summary(object, lambda =1, add.to.data = 0,
                by.realisations=TRUE, ...)
```

an object of the class geodata.

value of the Box-Cox transformation parameter. Two particular cases are $\lambda = 1 whichcorrespondstonotransformationand \lambda = 0 correspondingtothelog - transformation.scalar, Constantvaluetobeaddedtothedatavalues.Onlyusedifavaluedifferentfrom1i$

logical. Indicates whether the summary must be performed separatly for each realisation, if the geodata object contains the element realisations. Defaults to TRUE.

further arguments to be passed to the function summary.defaultsummary.default. A list with components

a matrix with minimum and maximum values for the coordinates.

minimum and maximum distances between pairs of points.

a matrix with minimum and maximum values for the coordinates. Only returned if there is an element borders in the geodata object.

summary statistics (min, max, quartiles and mean) for the data.

summary statistics (min, max, quartiles and mean) for the offset variable. Only returned if there is an element units.m in the geodata object.

summary statistics (min, max, quartiles and mean) for the covariate(s). Only returned if there is an element covariate in the geodata object.

names of other elements if present in the geodata object. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. summarysummary, as.geodataas.geodata. summary(s100)

summary(ca20) summary.likGRFSummarizes Parameter Estimation Results for Gaussian Random Fieldssummary.likGRF print.likGRFsummary.likGRFprint.likGRF print.summary.likGRFsummary.likGRFprint.summary.likGRF spatialsummary.likGRF printsummary.likGRF Summarizes results returned by the function likfitlikfit.
Functions are *methods* for summarysummary and printprint for the classes likGRF and summary.likGRF.

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

an object of classclass *likGRF*, typically a result of a call to likfitlikfit.

an object of classclass *likGRF* or classclass *summary.likGRF*, typically resulting from a call to likfitlikfit.

the number of significant digits to use when printing.

extra arguments for printprint. A detailed summary of a object of the class likGRF is produced by by summary.likGRF and printed by print.summary.likGRF. This includes model specification with values of fixed and estimated parameters. A simplified summary of the parameter estimation is printed by print.likGRF. print.likGRF prints the parameter estimates and the value of the maximized likelihood.
summary.likGRF returns a list with main results of a call to likfitlikfit.
print.summary.likGRF prints these results on the screen (or other output device) in a "nice" format. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br, Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. likfitlikfit, printprint, summarysummary. See examples for the function likfit() summary.variofitSummarize Results of Variogram Estimationsummary.variofit print.summary.variofitsummary.variofitprint.summary.variofit print.variofitsummary.variofitprint.variofit spatialsummary.variofit This function prints a summary of the parameter estimation results given by variofitvariofit.

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

an object of the class "variomodel" typically an output of variofitvariofit.

other arguments to be passed to the function printprint or summarysummary. Prints a summary of the estimation results on the screen or other output device. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. The functions variofitvariofit for variogram based estimation. For likelihood based parameter estimation see likfitlikfit.

s100.vario ¡- variog(s100, max.dist=1) wls ¡- variofit(s100.vario, ini=c(.5, .5), fix.nugget = TRUE) wls summary(wls) tceTCE concentrations in groundwater in a vertical cross sectiontce datasetstce Measurements at 56 locations of concentration of trichloroethylene (TCE) in groundwater on a transect in a fine-sand superficial aquifer. Extract from Kitanidis' book.

```
data(tce)
```

An object of the class geodata which is a list with the elements:

coords coordinates of the data location (feet).

data the data vector with measurements of the TCE concentration (ppb).

Kitanidis, P.K. Introduction to geostatistics - applications in hidrology (1997). Cambridge University Press. summary(tce) summary(tce, lambda=0) plot(tce) points(tce) points(tce, lambda=0) trend.spatialBuilds the Trend Matrixtrend.spatial spatialtrend.spatial Builds the *trend* matrix in accordance to a specification of the mean provided by the user.

```
trend.spatial(trend, geodata, add.to.trend)
```

specifies the mean part of the model. See DETAILS below.

optional. An object of the class geodata as described in as.geodataas.geodata.

optional. Specifies aditional terms to the mean part of the model. See details below. The implicity model assumes that there is an underlying process with mean $\mu(x), where x = (x_1, x_2) denotes the coordinates of a spatial location. The argument trend defines the form$

- "cte"the mean is assumed to be constant over the region, in which case $\mu(x) = \mu. This is the default option."1st" the mean is assumed to be a first order polynomial on the coordinates$ $\mu(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
- "2nd"the mean is assumed to be a second order polynomial on the coordinates: $\mu(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 (x_1)^2 + \beta_4 (x_2)^2 + \beta_5 x_1 * x_2 \tilde{} model a model specification. See formula formula for further details on how to specify a model inu$
-

Denote by x$_1 and x_2 the spatial coordinates. The following specifications are equivalent :$
trend = "1st" and trend = ˜ x1 + x2

trend = "2nd" and trend = ˜ x1 + x2 + I(x1ˆ2) + I(x2ˆ2) + I(x1*x2)
Search path for covariates
Typically, functions in the package geoR which calls trend.spatial will have the arguments geodata, coords and data.

When the trend is specifed as trend = ˜ model the terms included in the model will be searched for in the following path sequence (in this order):

1. in the users/session Global environment
2. in the session search path
3. as elements of the list geodata

4. as columns in a data-frame geodata$covariates
5. as columns in a data-frame geodata$data
6. in remainder of the session search path

The argument add.to.trend adds terms to what is specified in the argument trend. This seems redundant but allow specifications of the type: trend="2nd", add.trend=~other.covariates.

An object of the class trend.spatial which is an n $\times p trend matrix, where n is the number of spatial locations and p is the number of mean parameters in the model.$ This is an auxiliary function typically called by other geoR functions. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. The section DETAILS in the documentation for likfitlikfit for more about the underlying model. a first order polynomial trend trend.spatial("1st", sic.100)[1:5,] a second order polynomial trend trend.spatial("2nd", sic.100)[1:5,] a trend with a covariate trend.spatial(altitude, sic.100)[1:5,] a first degree trend plus a covariate trend.spatial(coords+altitude, sic.100)[1:5,] with produces the same as trend.spatial("1st", sic.100, add= altitude)[1:5,] and yet another exemple trend.spatial("2nd", sic.100, add= altitude)[1:5,] varcov.spatialComputes Covariance Matrix and Related Resultsvarcov.spatial spatialvarcov.spatial This function builds the covariance matrix for a set of spatial locations, given the covariance parameters. According to the input options other results related to the covariance matrix (such as decompositions, determinants, inverse. etc) can also be returned.

```
varcov.spatial(coords = NULL, dists.lowertri = NULL,
               cov.model = "matern", kappa = 0.5, nugget = 0,
               cov.pars = stop("no cov.pars argument"),
               inv = FALSE, det = FALSE,
               func.inv = c("cholesky", "eigen", "svd", "solve"),
               scaled = FALSE,  only.decomposition = FALSE,
               sqrt.inv = FALSE, try.another.decomposition = TRUE,
               only.inv.lower.diag = FALSE, ...)
```

an n $\times 2 matrix with the coordinates of the data locations. If not provided the argument dists.lowertri should be provided$

a string indicating the type of the correlation function. More details in the documentation for cov.spatialcov.spatial. Defaults are equivalent to the *exponential* model.

values of the additional smoothness parameter, only required by the following correlation functions: "matern", "powered.exponential", "cauchy" and "gneiting.matern".

the value of the nugget parameter $\tau^2. a vector with 2 elements or an n s \times 2 matrix with the covariance parameters. The f$

if TRUE the inverse of covariance matrix is returned. Defaults to FALSE.

if TRUE the logarithmic of the square root of the determinant of the covariance matrix is returned. Defaults to FALSE.

algorithm used for the decomposition and inversion of the covariance matrix. Options are "chol" for Cholesky decomposition, "svd" for singular value decomposition and "eigen" for eigenvalues/eigenvectors decomposition. Defaults to "chol".

logical indicating whether the covariance matrix should be scaled. If TRUE the partial sill parameter $\sigma^2 issetto1.DefaultstoFALSE.logical.IfTRUEonlythesquarerootofthecovariancematrixisreturned$

if TRUE the square root of the inverse of covariance matrix is returned. Defaults to FALSE.

logical. If TRUE and the argument func.inv is one of "cholesky", "svd" or "solve", the matrix decomposition or inversion is tested and, if it fails, the argument func.inv is re-set to "eigen".

logical. If TRUE only the lower triangle and the diagonal of the inverse of the covariance matrix are returned. Defaults to FALSE.

for naw, only for internal usage. The elements of the covariance matrix are computed by the function cov.spatialcov.spatial. Typically this is an auxiliary function called by other functions in the geoR package. The result is always list. The components will vary according to the input options. The possible components are:

the covariance matrix.

a square root of the covariance matrix.

the lower triangle of the inverse of covariance matrix.

the diagonal of the inverse of covariance matrix.

the inverse of covariance matrix.

a square root of the inverse of covariance matrix.

the logarithmic of the square root of the determinant of the covariance matrix. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. cov.spatialcov.spatial for more information on the correlation functions; cholchol, solvesolve, svdsvd and eigeneigen for matrix inversion and/or decomposition. varcovBGCCMCovariance matrix for the bivariate Gaussian common component geostatistical modelvarcovBGCCM .cov012.modelvarcovBGCCM.cov012.model .dist12varcovBGCCM.dist12 spatialvarcovBGCCM Covariance matrix for the bivariate Gaussian common component geostatistical model or its inverse, and optionally the determinant of the matrix.

```
varcovBGCCM(dists.obj, cov0.pars, cov1.pars, cov2.pars,
            cov0.model = "matern", cov1.model = "matern",
            cov2.model = "matern", kappa0 = 0.5, kappa1 = 0.5,
            kappa2 = 0.5, scaled = TRUE, inv = FALSE, det = FALSE)
```

a vector with distance values

covarianve paremeter values for the common component

covariance parameter for the individual structure of the first variable covariance parameter for the individual structure of the second variable

character indicating a valid correlation model

character indicating a valid correlation model

character indicating a valid correlation model

scalar

scalar

scalar

logical

logical. If TRUE the inverse of the covariance matrix is returned instead.

logical. Optional, if TRUE the logarithm of the detarminant of the covariance matrix is returned as an attribute. A matrix which is the covariance matrix for the bivariate Gaussian common component geostatistical model or its inverse if inv=TRUE. If det=T the logarithm of the determinant of the matrix is also returned as an attribute named logdetS. Warning This is a new function and still in draft format and pretty much untested. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. cov.spatialcov.spatial, varcov.spatialvarcov.spatial see http://www.leg.ufpr.br/geoR/tutorials/CCM.R variofit Variogram Based Parameter Estimation variofit .loss.variovariofit.loss.vario spatialvariofit Estimate covariance parameters by fitting a parametric model to a empirical variogram. Variograms models can be fitted by using weighted or ordinary least squares.

```
variofit(vario, ini.cov.pars, cov.model,
         fix.nugget = FALSE, nugget = 0,
         fix.kappa = TRUE, kappa = 0.5,
         simul.number = NULL, max.dist = vario$max.dist,
         weights, minimisation.function,
         limits = pars.limits(), messages, ...)
```

an object of the class "variogram", typically an output of the function variogvariog. The object is a list with information about the empirical variogram.

initial values for the covariance parameters: $\sigma^2 (partialsill) and \phi (rangeparameter). SeeDETAILSbelow.astringwiththenameofthecorrelationfunction.Forf$

logical, indicating whether the parameter $\tau^2 (nuggetvariance) shouldberegardedasfixed (fix.nugget = TRUE) orsh$

logical, indicating whether the parameter $\kappa shouldberegardedasfixedorbeestimated.DefaultstoTRUE.valueofthe$
$"matern", "powered.exponential", "cauchy" and "gneiting.matern". Defaultsto 0.5.$

number of simulation. To be used when the object passed to the argument vario has empirical variograms for more than one data-set (or simulation). Indicates to which one the model will be fitted.

maximum distance considered when fitting the variogram. Defaults to vario$max.dist.

type weights used in the loss function. See DETAILS below.

values defining lower and upper limits for the model parameters used in the numerical minimisation. Only valid if minimisation.function = "optim". The auxiliary function pars.limitspars.limits is called to set the limits.

minimization function used to estimate the parameters. Options are "optim", "nlm". If weights = "equal" the option "nls" is also valid and det as default. Otherwise defaults to "optim".

logical. Indicates whether or not status messages are printed on the screen (or other output device) while the function is running.

further parameters to be passed to the minimization function. Typically arguments of the type control() which controls the behavior of the minimization algorithm. See documentation for the selected minimization function for further details.

Numerical minimization

The parameter values are found by numerical optimization using one of the functions: optimoptim, nlmnlm and nlsnls. In given circunstances the algorithm may not converge to correct parameter values when called with default options and the user may need to pass extra options for the optimizers. For instance the function optim takes a control argument. The user should try different initial values and if the parameters have different orders of magnitude may need to use options to scale the parameters. Some possible workarounds in case of problems include:

- rescale you data values (dividing by a constant, say)
- rescale your coordinates (subtracting values and/or dividing by constants)
- Use the mechanism to pass control() options for the optimiser internally

Initial values

The algorithms for minimization functions require initial values of the parameters.

A unique initial value is used if a vector is provided in the argument ini.cov.pars. The elements are initial values for $\sigma^2 and \phi, respectively. This vector is concatenated with the value of the argument nugget if fix.nugget = F$

Specification of multiple initial values is also possible. If this is the case, the function searches for the one which minimizes the loss function and uses this as the initial value for the minimization algorithm. Multiple initial values are specified by providing a matrix in the argument ini.cov.pars and/or, vectors in the arguments nugget and kappa (if included in the estimation). If ini.cov.pars is a matrix, the first column has values of $\sigma^2 and the second has values of \phi.$

Alternatively the argument ini.cov.pars can take an object of the class eyefit or variomodel. This allows the usage of an output of the functions eyefiteyefit, variofitvariofit or likfitlikfit be used as initial value.

If minimisation.function = "nls" only the values of $\phi and \kappa (if this is included in the estimation) are used. Values for the remaning are not need by the algorithm.$

If cov.model = "linear" only the value of $\sigma^2 is used. Values for the remaning are not need by this algorithm.$

If cov.model = "pure.nugget" no initial values are needed since no minimisation function is used.

Weights

The different options for the argument weights are used to define the loss function to be minimised. The available options are as follows.

"npairs" indicates that the weights are given by the number of pairs in each bin. This is the default option unless variog$output.type == "cloud". The loss function is: LOSS(θ) = $\sum_k n_k[(\hat{\gamma}_k) - \gamma_k(\theta)]^2$

"cressie" weights as suggested by Cressie (1985). LOSS(θ) = $\sum_k n_k[\frac{\hat{\gamma}_k - \gamma_k(\theta)}{\gamma_k(\theta)}]^2$

"equal" equal values for the weights. For this case the estimation corresponds to the ordinary least squares variogram fitting. This is the default option if variog$output.type == "cloud". LOSS(θ) = $\sum_k[(\hat{\gamma}_k) - \gamma_k(\theta)]^2$

Where $\theta isthevectorwiththevariogramparametersandforeachk^{th} - binn_k isthenumberofpairs, (\hat{\gamma}_k)isthevalueoftheempiricalvariogramand\gamma_k(\theta)isthevalueofthetheoreticalvar$

See also Cressie (1993) and Barry, Crowder and Diggle (1997) for further discussions on methods to estimate the variogram parameters.

An object of the classclass "variomodel" and "variofit" which is list with the following components:

value of the nugget parameter. An estimated value if fix.nugget = FALSE or a fixed value if fix.nugget = TRUE.

a two elements vector with estimated values of the covariance parameters $\sigma^2 and \phi, respectively.astringwiththenameofthecorrelationfunction.$

fixed value of the smoothness parameter.

minimized value of the loss function.

maximum distance considered in the variogram fitting.

minimization function used.

a string indicating the type of weights used for the variogram fitting.

a string indicating the type of variogram fitting method (OLS or WLS).

logical indicating whether the parameter $\kappa wasfixed.logicalindicatingwhetherthenuggetparameterwasfixed.$

transformation parameters inherith from the object provided in the argument vario.

status messages returned by the function.

the function call. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br, Peter J. Diggle p.diggle@lancaster.ac.uk. Barry, J.T., Crowder, M.J. and Diggle, P.J. (1997) Parametric estimation of the variogram. *Tech. Report, Dept Maths & Stats, Lancaster University.*

Cressie, N.A.C (1985) *Mathematical Geology.* 17, 563-586.

Cressie, N.A.C (1993) *Statistics for Spatial Data.* New York: Wiley.

Further information on the package geoR can be found at: http://www.leg.ufpr.br/geoR. cov.spatialcov.spatial for a detailed description of the available correlation (variogram) functions, likfitlikfit for maximum and restricted maximum likelihood estimation, lines.variomodellines.variomodel for graphical output of the fitted model. For details on the minimization functions see optimoptim, nlmnlm and nlsnls. vario100 ¡- variog(s100, max.dist=1) ini.vals ¡- expand.grid(seq(0,1,l=5), seq(0,1,l=5)) ols ¡- variofit(vario100, ini=ini.vals, fix.nug=TRUE, wei="equal") summary(ols) wls

¡- variofit(vario100, ini=ini.vals, fix.nug=TRUE) summary(wls) plot(vario100) lines(wls) lines(ols, lty=2)
variogCompute Empirical Variogramsvariog .define.binsvariog.define.bins .rfm.binvariog.rfm.bin spatialvariog smoothvariog robustvariog Computes sample (empirical) variograms with options for the *classical* or *robust* estimators. Output can be returned as a binned variogram, a variogram cloud or a smoothed variogram. Data transformation (Box-Cox) is allowed. “Trends” can be specified and are fitted by ordinary least squares in which case the variograms are computed using the residuals.

```
variog(geodata, coords = geodata$coords, data = geodata$data,
       uvec = "default", breaks = "default",
       trend = "cte", lambda = 1,
       option = c("bin", "cloud", "smooth"),
       estimator.type = c("classical", "modulus"),
       nugget.tolerance, max.dist, pairs.min = 2,
       bin.cloud = FALSE, direction = "omnidirectional", tolerance = pi/8,
       unit.angle = c("radians","degrees"), angles = FALSE, messages, ...)
```

a list containing element coords as described next. Typically an object of the class ”geodata” - a geoR data-set. If not provided the arguments coords must be provided instead.

an n $\times 2 matrixcontainingcoordinatesofthendatalocationsineachrow.Defaultstogeodata\$coords, ifpr$

a vector with values used to define the variogram binning. Only used when option = ”bin”. See DETAILS below for more details on how to specify the bins.

a vector with values to define the variogram binning. Only used when option = ”bin”. See DETAILS below for more details on how to specify the bins.

specifies the mean part of the model. See documentation of trend.spatialtrend.spatial for further details. Defaults to ”cte”.

values of the Box-Cox transformation parameter. Defaults to 1 (no transformation). If another value is provided the variogram is computed after transforming the data. A case of particular interest is $\lambda = 0 whichcorrespondstolog - transformation.definestheoutputtype :$
$theoptions"bin"returnsvaluesofbinnedvariogram,"cloud"returnsthevariogramcloudand"smooth"re$

”classical” computes the classical method of moments estimator. ”modulus” returns the variogram estimator suggested by Hawkins and Cressie (see Cressie, 1993, pg 75). Defaults to ”classical”.

a numeric value. Points which are separated by a distance less than this value are considered co-located. Defaults to zero.

a numerical value defining the maximum distance for the variogram. Pairs of locations separated for distance larger than this value are ignored for the variogram calculation. If not provided defaults takes the maximum distance among all pairs of data locations.

a integer number defining the minimum numbers of pairs for the bins. For option = ”bin”, bins with number of pairs smaller than this value are ignored. Defaults to NULL.

logical. If TRUE and option = "bin" the cloud values for each class are included in the output. Defaults to FALSE.

a numerical value for the directional (azimuth) angle. This used to specify directional variograms. Default defines the omnidirectional variogram. The value must be in the interval [0, $\pi] radians ([0, 180] degrees). numerical value for the tolerance angle, when computing directional variograms. The valu$

defines the unit for the specification of angles in the two previous arguments. Options are "radians" and "degrees", with default to "radians".

Logical with default to FALSE. If TRUE the function also returns the angles between the pairs of points (unimplemented).

logical. Indicates whether status messages should be printed on the screen (or output device) while the function is running.

arguments to be passed to the function ksmoothksmooth, if option = "smooth". Variograms are widely used in geostatistical analysis for exploratory purposes, to estimate covariance parameters and/or to compare theoretical and fitted models against sample variograms.

Estimators

The two estimators currently implemented are:

- *classical* (method of moments) estimator: $\gamma(h) = \frac{1}{2N_h}\sum_{i=1}^{N_h}[Y(x_{i+h}) - Y(x_i)]^2$
- Hawkins and Cressie's *modulus* estimator $\gamma(h) = \frac{[\frac{1}{N_h}\sum_{i=1}^{N_h}|Y(x_{i+h})-Y(x_i)|^{\frac{1}{2}}]^4}{0.914+\frac{0.988}{N_h}}$

Defining the bins

The defaults

If arguments breaks and uvec are not provided, the binning is defined as follows:

1. read the argument max.dist. If not provided it is set to the maximum distance between the pairs of points.
2. the center of the bins are initially defined by the sequence u = seq(0, max.dist, l = 13).
3. the interval spanned by each bin is given by the mid-points between the centers of the bins.

If an vector is passed to the argument breaks its elements are taken as the limits of the bins (classes of distance) and the argument uvec is ignored.

Variations on the default

The default definition of the bins is different for some particular cases.

1. if there are coincident data locations the bins follows the default above but one more bin is added at the origin (distance zero) for the collocated points.
2. if the argument nugget.tolerance is provided the separation distance between all pairs in the interval [0, nugget.tolerance] are set to zero. The first bin distance is set to zero (u[1] = 0). The remaining bins follows the default.

3. if a scalar is provided to the argument uvec the default number of bins is defined by this number.

4. if a vector is provided to the argument uvec, its elements are taken as central points of the bins.

An object of the classclass variogram which is a list with the following components:

a vector with distances.

a vector with estimated variogram values at distances given in u.

number of pairs in each bin, if option = "bin".

standard deviation of the values in each bin.

limits defining the interval spanned by each bin.

a logical vector indicating whether the number of pairs in each bin is greater or equal to the value in the argument pairs.min.

variance of the data.

parameters of the mean part of the model fitted by ordinary least squares.

echoes the option argument.

maximum distance between pairs allowed in the variogram calculations.

echoes the type of estimator used.

number of data.

value of the transformation parameter.

trend specification.

value of the nugget tolerance argument.

direction for which the variogram was computed.

tolerance angle for directional variogram.

lags provided in the function call.

the function call. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Cressie, N.A.C (1993) *Statistics for Spatial Data.* New York: Wiley.

Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. variog4variog4 for more on computation of directional variograms, variog.model.envvariog.model.env and variog.mc.envvariog.mc.env for variogram envelopes, variofitvariofit for variogram based parameter estimation and plot.variogramplot.variogram for graphical output. computing variograms: binned variogram vario.b ¡- variog(s100, max.dist=1) variogram cloud vario.c ¡- variog(s100, max.dist=1, op="cloud") binned variogram and stores the cloud vario.bc ¡- variog(s100, max.dist=1, bin.cloud=TRUE) smoothed variogram vario.s ¡- variog(s100, max.dist=1, op="sm", band=0.2) plotting the variograms: par(mfrow=c(2,2)) plot(vario.b, main="binned variogram") plot(vario.c, main="variogram cloud") plot(vario.bc, bin.cloud=TRUE, main="clouds for binned variogram") plot(vario.s, main="smoothed variogram")

computing a directional variogram vario.0 ¡- variog(s100, max.dist=1, dir=0, tol=pi/8) plot(vario.b, type="l", lty=2) lines(vario.0) legend("topleft", legend=c("omnidirectional", expression(0 * degree)), lty=c(2,1)) variog.mc.envEnvelops for Empirical Variograms Based on Permutationvariog.mc.env spatialvariog.mc.env nonparametricvariog.mc.env Computes envelops for empirical variograms by permutation of the data values on the spatial locations.

```
variog.mc.env(geodata, coords = geodata$coords, data = geodata$data,
              obj.variog, nsim = 99, save.sim = FALSE, messages)
```

a list containing elements coords and data as described next. Typically an object of the class "geodata" - a geoR data-set. If not provided the arguments coords and data must be provided instead.

an n $\times 2matrix, eachrowcontainingEuclideancoordinatesofthendatalocations.Bydefaultittakestheelementcoord$

an object of the class "variogram", typically an output of the function variogvariog.

number of simulations used to compute the envelope. Defaults to 99.

logical. Indicates whether or not the simulated data are included in the output. Defaults to FALSE.

logical. If TRUE, the default, status messages are printed while the function is running. The envelops are obtained by permutation. For each simulations data values are randomly allocated to the spatial locations. The empirical variogram is computed for each simulation using the same lags as for the variogram originally computed for the data. The envelops are computed by taking, at each lag, the maximum and minimum values of the variograms for the simulated data. An object of the classclass "variogram.envelope" which is a list with the following components:

a vector with distances.

a vector with the minimum variogram values at each distance in u.

a vector with the maximum variogram values at each distance in u.

a matrix with simulated data. Only returned if save.sim = TRUE. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. variog.model.envvariog.model.env for envelops computed by from a model specification, variogvariog for variogram calculations, plot.variogramplot.variogram and variog.mc.envvariog.mc.env for graphical output. s100.vario ¡- variog(s100, max.dist=1) s100.env ¡- variog.mc.env(s100, obj.var = s100.vario) plot(s100.vario, envelope = s100.env) variog.model.envEnvelops for Empirical Variograms Based on Model Parametersvariog.model.env boot.variofitvariog.model.envboot.variofit spatialvariog.model.env Computes envelopes for a empirical variogram by simulating data for given model parameters.

Computes bootstrap paremeter estimates

```
variog.model.env(geodata, coords = geodata$coords, obj.variog,
                 model.pars, nsim = 99, save.sim = FALSE, messages)
```

```
boot.variofit(geodata, coords = geodata$coords, obj.variog,
              model.pars, nsim = 99, trace = FALSE, messages)
```

a list containing element coords as described next. Typically an object of the class "geodata" - a geoR data-set. If not provided the argument coords must be provided instead.

an n $\times 2 matrix, eachrowcontainingEuclideancoordinatesofthendatalocations.Bydefaultittakestheel$

a list with model specification and parameter values. The input is typically an object of the class variomodel which is an output of likfitlikfit, variofitvariofit. The required components of the list are:

- beta, the mean parameter. Defaults to zero.
- cov.model, the covariance model. Defaults to "exponential".
- cov.pars, the covariance parameters $\sigma^2 and \phi.kappa, theextracovarianceparametersforsomeofthe$
- nugget, the error component variance. Defaults to zero.
- estimator.type, the type of variogram estimator. Options for "classical" and "robust". Defaults to obj.variog$estimator.

number of simulations used to compute the envelopes. Defaults to 99.

logical. Indicates whether or not the simulated data are included in the output. Defaults to FALSE.

logical. If TRUE the fitted values for the bootstrap parameter estimation are printend while the function is running.

logical. If TRUE, the default, status messages are printed while the function is running. The envelopes are computed assuming a (transformed) Gaussian random field model. Simulated values are generated at the data locations, given the model parameters. The empirical variogram is computed for each simulation using the same lags as for the original variogram of the data. The envelopes are computed by taking, at each lag, the maximum and minimum values of the variograms for the simulated data. An object of the classclass "variogram.envelope" which is a list with the components:

a vector with distances.

a vector with the minimum variogram values at each distance in u.

a vector with the maximum variogram values at each distance in u.

a matrix with the simulated data. Only returned if save.sim = TRUE. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. variog.mc.envvariog.mc.env for envelops computed by using data permutation, variogvariog for variogram calculations, plot.variogramplot.variogram and variog.mc.envvariog.mc.env for graphical output. The functions likfitlikfit, variofitvariofit are used to estimate the model parameters. s100.ml ¡- likfit(s100, ini = c(0.5, 0.5), fix.nugget = TRUE) s100.vario ¡- variog(s100, max.dist = 1) s100.env ¡- variog.model.env(s100, obj.v

= s100.vario, model.pars = s100.ml) plot(s100.vario, env = s100.env) variog4Computes Directional Variogramsvariog4 spatialvariog4 Computes directional variograms for 4 directions provided by the user.

```
variog4(geodata, coords = geodata$coords, data = geodata$data,
        uvec = "default", breaks = "default", trend = "cte", lambda = 1,
        option = c("bin", "cloud", "smooth"),
        estimator.type = c("classical", "modulus"),
        nugget.tolerance, max.dist, pairs.min = 2,
        bin.cloud = FALSE, direction = c(0, pi/4, pi/2, 3*pi/4), tolerance = pi/8,
        unit.angle = c("radians", "degrees"), messages, ...)
```

a list containing element coords as described next. Typically an object of the class "geodata" - a geoR data-set. If not provided the arguments coords must be provided instead.

an n $\times 2 matrixcontainingcoordinatesofthendatalocationsineachrow.Defaultstogeodata\$coords, ifprovided.avec$

a vector with values to define the variogram binning. For further details see documentation for variogvariog.

a vector with values to define the variogram binning. For further details see documentation for variogvariog.

specifies the mean part of the model. The options are: "cte" (constant mean), "1st" (a first order polynomial on the coordinates), "2nd" (a second order polynomial on the coordinates), or a formula of the type ˜X where X is a matrix with the covariates (external trend). Defaults to "cte".

values of the Box-Cox transformation parameter. Defaults to 1 (no transformation). If another value is provided the variogram is computed after transforming the data. A case of particular interest is $\lambda = 0 whichcorrespondstolog - transformation.definestheoutputtype :$ $theoptions"bin"returnsvaluesofbinnedvariogram,"cloud"returnsthevariogramcloudand"smooth"returnsthekei$

"classical" computes the classical method of moments estimator. "modulus" returns the variogram estimator suggested by Hawkins and Cressie (see Cressie, 1993, pg 75). Defaults to "classical".

a numeric value. Points which are separated by a distance less than this value are considered co-located. Defaults to zero.

a numerical value defining the maximum distance for the variogram. Pairs of locations separated for distance larger than this value are ignored for the variogram calculation. Defaults to the maximum distance among the pairs of data locations.

a integer number defining the minimum numbers of pairs for the bins. For option = "bin", bins with number of pairs smaller than this value are ignored. Defaults to NULL.

logical. If TRUE and option = "bin" the cloud values for each class are included in the output. Defaults to FALSE.

a vector with values of 4 angles, indicating the directions for which the variograms will be computed. Default corresponds to c(0, 45, 90, 135) (degrees).

numerical value for the tolerance angle, when computing directional variograms. The value must be in the interval [0, 90] degrees. Defaults to $\pi/8. definestheunitforthespecificationofanglesinthetwopreviousarguments.Optionsare"degrees"ar$

logical. Indicates whether status messages should be printed on the screen (or output device) while the function is running.

arguments to be passed to the function ksmoothksmooth, if option = "smooth". The output is an object of the class variog4, a list with five components. The first four elements are estimated variograms for the directions provided and the last is the omnidirectional variogram. Each individual component is an object of the class variogram, an output of the function variogvariog. Paulo J. Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. variogvariog for variogram calculations and plot.variog4plot.variog4 for plotting results var4 ¡- variog4(s100, max.dist=1) plot(var4) woKriging example data from Webster and Oliverwo datasetswo Data used in Chapter 8, page 156 of Webster and Oliver (2001) to illustrate properties of the kriging predictor.

```
data(wo)
```

An object of the class geodata which is a list with the elements:

coords coordinates of the data location.

data the data vector.

x1 coordinate of the centrally located prediction point.

x2 coordinate of the off-centre prediction point.

Webster, R. and Oliver, M.A. (2001). Geostatistics for Environmental Scientists. Wiley. attach(wo) par(mfrow=c(1,2)) plot(c(-10,130), c(-10,130), ty="n", asp=1) points(rbind(coords, x1)) text(coords[,1], 5+coords[,2], format(data)) text(x1[1]+5, x1[2]+5, "?", col=2) plot(c(-10,130), c(-10,130), ty="n", asp=1) points(rbind(coords, x2)) text(coords[,1], 5+coords[,2], format(data)) text(x2[1]+5, x2[2]+5, "?", col=2) wolfcampWolfcamp Aquifer Datawolfcamp wolfwolfcampwolf datasetswolfcamp Piezometric head measurements taken at the Wolfcamp Aquifer, Texas, USA. See Cressie (1993, p.212–214) for description of the scientific problem and the data. Original data were converted to SI units: coordinates are given in kilometers and pressure heads to meters.

```
data(wolfcamp)
```

An object of the classclass "geodata", which is list with two components:

coords the coordinates of the data locations. The distance are given in kilometers.

data values of the piezometric head. The unit is heads to meters.

Harper, W.V and Furr, J.M. (1986) Geostatistical analysis of potentiometric data in the Wolfcamp Aquifer of the Palo Duro Basin, Texas. *Technical Report BMI/ONWI-587, Bettelle Memorial Institute, Columbus, OH.* Cressie, N.A.C (1993) *Statistics for Spatial Data.* New York: Wiley.

Papritz, A. and Moyeed, R. (2001) Parameter uncertainty in spatial prediction: checking its importance by cross-validating the Wolfcamp and Rongelap data sets. *GeoENV 2000: Geostatistical for Environmental Applications. Ed. P. Monestiez, D. Allard, R. Froidevaux.* Kluwer. summary(wolfcamp) plot(wolfcamp) wrappersWrappers for the C functions used in geoRwrappers .bilinearformXAYwrappers.bilinearformXAY .Ccor.spatialwrappers.Ccor.spatial .corr.diaglowertriwrappers.corr.diaglowertri .diagquadraticformXAXwrappers.diagquadraticformXAX diffpairswrappersdiffpairs loccoordswrappersloccoords spatialwrappers programmingwrappers interfacewrappers These functions are *wrappers* for some (but not all) the C functions included in the geoR package.
Typically the C code is directly called from the geoR functions but these functions allows independent calls.

```
diffpairs(coords, data)
loccoords(coords, locations)
.diagquadraticformXAX(X, lowerA, diagA)
.bilinearformXAY(X, lowerA, diagA, Y)
.corr.diaglowertri(coords, cov.model, phi, kappa)
.Ccor.spatial(x, phi, kappa, cov.model)
```

an n $\times 2 matrixwiththedatacoordinates.anvectorwiththedatavalues.$

an N $\times 2 matrixwiththecoordinatesofthepredictionlocations.avectorwiththediagonaltermsofthesymmetricmatri$

a vector with the diagonal terms of the symmetric matrix A.

a matrix with conforming dimensions.

a matrix with conforming dimensions.

covariance model, see cov.spatialcov.spatial for options and more details.

numerical value of the correlation function parameter phi.

numerical value of the correlation function parameter kappa.

a vector of distances. The outputs for the different functions are:

returns a list with elements dist - the distance between pairs of points, and diff - the difference between the values of the attributes.

returns a n $\times N matrixwithdistancesbetweendatapointsandpredictionlocations.$

returns a vector with the diagonal term of the quadratic form X' A X.

returns a vector or a matrix with the terms of the quadratic form X' A Y.

returns the lower triangle of the correlation matrix, including the diagonal.

returns a vector of values of spatial correlations. Paulo Justiniano Ribeiro Jr. paulojus@leg.ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. wrcPoints of a water retention curve data setwrc

datasetswrc spatialwrc Soil density and measures of the water retention curve obtained at different pressures on a regular grid with 10x25 points spaced by 5 meters.

```
data(wrc)
```

A data frame with 250 observations on the following 11 variables.

CoordX a numeric vector with the X coordinates of the samples.

CoordY a numeric vector with the Y coordinate of the samples.

Densidade a numeric vector, soil density (g/cm^3)$anumericvector, watercontentatapressureof 5mca - -5 \times 10^2 Pa(atm)$

Pr10 a numeric vector, water content at a pressure of 10 mca – 1 $\times 10^3 Pa(atm) anumericvector, watercontentatapressureof 60mca - -6 \times 10^3 Pa(atm)$

Pr100 a numeric vector, water content at a pressure of 100 mca – 1 $\times 10^4 Pa(atm) anumericvector, watercontentatapressureof 306mca - -3 \times 10^4 Pa(atm)$

Pr816 a numeric vector, water content at a pressure of 816 mca – 8 $\times 10^4 Pa(atm) anumericvector, watercontentatapressureof 3060mca - -3 \times 10^5 Pa(atm)$

Pr15300 a numeric vector, water content at a pressure of 15300 mca – 1.5 $\times 10^6 Pa(atm)$

Uniformity trial with 250 undisturbed soil samples collected at 25cm soil depth of spacing of 5 meters, resulting on a regular grid of 25 $\times 10 samplingpoints.$

For each sampling point there are measurents of the soil density and water content obtained at eight pressures: 5, 10, 60, 100, 306, 816, 3060 and 15300 meters of column of water (mca), corresponding to 5 $\times 10^2, 1 \times 10^3, 6 \times 10^3, 1 \times 10^4, 3 \times 10^4, 8 \times 10^4, 3 \times 10^5, 1.5 \times 10^6 Pa.$

The experiment aimed to use the water contents of the samples to estimate the water retention curve at the 250 data points.

See also the data-set soil250soil250 with soil chemistry properties measured at the same points. MORAES, S.O. (1991) Heterogeneidade hidráulica de uma terra roxa estruturada. PhD Thesis. ESALQ/USP. MORAES, S. O. ; LIBARDI, P. L. ; REICHARDT, K. (1993) Problemas metodológicos na obtenção da curva de retenção de água pelo solo. Scientia Agricola, Piracicaba, v. 50, n. 3, p. 383-392.

MORAES, S. O. ; LIBARDI, P. L. ; REICHARDT, K. ; BACCHI, O. O. S. (1993) Heterogeneidade dos pontos experimentais de curvas de retenção da água do solo.. Scientia Agricola, Piracicaba, v. 50, n. 3, p. 393-402.

MORAES, S. O. ; LIBARDI, P. L. (1993) Variabilidade da água disponível em uma terra roxa estruturada latossólica. Scientia Agricola, Piracicaba, v. 50, n. 3, p. 393-402, 1993. pr100 ¡- as.geodata(wrc, data.col=7) summary(pr100) plot(pr100) xvalidCross-validation by krigingxvalid print.summary.xvalidxvalidprint.summary.xvalid summary.xvalidxvalidsummary.xvalid spatialxvalid A function to perform model validation by comparing observed and values predicted by kriging. Options include: (i) *leaving-one-out* cross-validation where each data location is removed from the data set and the variable at this location is predicted using the remaining locations, for a given model. This can be computed for all or a subset of the data locations; (ii) *external validation* can be performed by using validation locations other than data locations.

```
xvalid(geodata, coords = geodata$coords, data = geodata$data,
       model, reestimate = FALSE, variog.obj = NULL,
       output.reestimate = FALSE, locations.xvalid = "all",
       data.xvalid = NULL, messages, ...)
```

a list containing element coords as described next. Typically an object of the class "geodata" - a geoR data-set. If not provided the arguments coords must be provided instead.

an n $\times 2 matrixcontainingcoordinatesofthendatalocationsineachrow.Defaultstogeodata\$coords, ifprovided.avec$

an object containing information on a fitted model. Typically an output of likfitlikfit, variofitvariofit. If an object of the class eyefit is passed it takes the first model specified in the object.

logical. Indicates whether or not the model parameters should be re-estimated for each point removed from the data-set.

on object with the empirical variogram, typically an output of the function variogvariog. Only used if reestimate = TRUE and the object passed to the argument model is the result of a variogram based estimation, i.e. if the model was fitted by variofitvariofit.

logical. Only valid if reestimate = TRUE. Specifies whether the re-estimated parameters are returned.

there are three possible specifications for this argument: "all" indicates the *leaving-on-out* method is used at all data locations. The second possibility is to use only a sub-set of the data for cross-validation in which case the argument takes a vector with numbers (indexes) indicating at which of the data locations the cross-validation should be performed. The third option is to perform external validation, on locations other than data locations used for the model. For the latter a matrix with the coordinates of the validation points should be provided and the argument data.xvalid mandatory.

data values at the validation locations. Only used if the validation locations are other than the data locations.

logical. Indicates whether status messages should be printed on the screen (or output device) while the function is running.

further arguments to the minimization functions used by likfitlikfit, variofitvariofit. The cross-validation uses internally the function krige.conv to predict at each location.

For models fitted by variofitvariofit the parameters $\kappa, \psi_A, \psi_R and \lambda arealwaysregardedasfixedwhenreestimatingthemodel.$

See documentation of the function likfitlikfit for further details on the model specification and parameters. An object of the classclass "xvalid" which is a list with the following components:

the original data.

the values predicted by cross-validation.

the cross-validation prediction variance.

the differences data - predicted value.

the errors divided by the square root of the prediction variances.

the cumulative probability at original value under a normal distribution with parameters given by the cross-validation results.
A method for summary returns summary statistics for the errors and standard errors.

If reestimate = TRUE and output = TRUE additional columns are added to the resulting data-frame with the values of the re-estimated parameters. Paulo J. Ribeiro Jr. paulojus@ufpr.br,
Peter J. Diggle p.diggle@lancaster.ac.uk. Further information on the package geoR can be found at:
http://www.leg.ufpr.br/geoR. plot.xvalidplot.xvalid for plotting of the results, likfitlikfit, variofitvariofit for parameter estimation and krige.convkrige.conv for the kriging method used for predictions. Maximum likelihood estimation s100.ml ¡- likfit(s100, ini = c(.5, .5), fix.nug = TRUE) Weighted least squares estimation s100.var ¡- variog(s100, max.dist = 1) s100.wls ¡- variofit(s100.var, ini = c(.5, .5), fix.nug = TRUE) Now, performing cross-validation without reestimating the model s100.xv.ml ¡- xvalid(s100, model = s100.ml) s100.xv.wls ¡- xvalid(s100, model = s100.wls) Plotting results par.ori ¡- par(no.readonly = TRUE) par(mfcol=c(5,2), mar=c(2.3,2.3,.5,.5), mgp=c(1.3, .6, 0)) plot(s100.xv.ml) par(mfcol=c(5,2)) plot(s100.xv.wls) par(par.ori)