Package ‘npsp’

July 2, 2014

Type Package

Title Nonparametric spatial (geo)statistics

Version 0.2-4

Date 2013-11-01

Author Ruben Fernandez-Casal

Maintainer Ruben Fernandez-Casal <rubenfcasal@gmail.com>

Depends R (>= 2.14.0)

Imports quadprog

Suggests sp, gstat, geoR, fields, DEoptim

Description Multidimensional nonparametric spatio-temporal geostatistics.
S3 classes and methods for multidimensional: gridded data, linear binning,local polynomial kernel regression, density and variogram estimation.
Nonparametric methods for trend and/or variogram inferences.

LazyData yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-11-04 08:13:41

R topics documented:

`npsp-package` *npsp: Nonparametric spatial (geo)statistics*

Description

This package implements nonparametric methods which may be useful in geostatistical practice.

Main functions

`locpol`, `np.den` and `np.svar` use local polynomial kernel methods to compute nonparametric estimates of a multidimensional regression function, a probability density function or a semivariogram (or their first derivatives), respectively. Estimates of these functions can be constructed for any dimension (the amount of available memory is the only limitation). To speed up computations, linear binning is used to discretize the data. A full bandwidth matrix and a multiplicative triweight kernel is used to compute the weights. Main calculations are performed in FORTRAN using the LAPACK library.

`fitsvar.sb.iso` fits a 'nonparametric' isotropic Shapiro-Botha variogram model by WLS. Currently, only isotropic semivariogram estimation is supported.

Among the other functions intended for direct access by the user, the following could be emphasized: `binning`, `bin.den`, `svar.bin`, `h.cv` and `interp`. There are also some functions which can be used to interact with other packages. For instance, `as.variogram` (**geoR**) or `as.vgm` (**gstat**).

Kriging is not yet implemented in this package. Users are encouraged to use `krige` (or `krige.cv`) utilities in **gstat** package together with `as.vgm`.

Acknowledgments

Important suggestions and contributions to some techniques included here were made by Tomas Cotos-Ya\~nez (Dep. Statistics, University of Vigo, Spain).

Author(s)

Ruben Fernandez-Casal (Dep. Mathematics, University of A Coru\~na, Spain). Please send comments, error reports or suggestions to rubenfcasal@gmail.com.

References

Fernandez Casal R., Gonzalez Manteiga W. and Febrero Bande M. (2003) Flexible Spatio-Temporal Stationary Variogram Models, *Statistics and Computing*, **13**, 127-136.

Rupert D. and Wand M.P. (1994) Multivariate locally weighted least squares regression. *The Annals of Statistics*, **22**, 1346-1370.

Shapiro A. and Botha J.D. (1991) Variogram fitting with a general class of conditionally non-negative definite functions. *Computational Statistics and Data Analysis*, **11**, 87-96.

Wand M.P. (1994) Fast Computation of Multivariate Kernel Estimators. *Journal of Computational and Graphical Statistics*, **3**, 433-445.

Wand M.P. and Jones M.C. (1995) *Kernel Smoothing*. Chapman and Hall, London.

---

aquifer *Wolfcamp aquifer data*

Description

The Deaf Smith County (Texas, bordering New Mexico) was selected as an alternate site for a possible nuclear waste disposal repository in the 1980s. This site was later dropped on grounds of contamination of the aquifer, the source of much of the water supply for west Texas. In a study conducted by the U.S. Department of Energy, piezometric-head data were obtained at 85 locations (irregularly scattered over the Texas panhandle) by drilling a narrow pipe through the aquifer.

This data set has been used in numerous papers. For instance, Cressie (1989) lists the data and uses it to illustrate kriging, and Cressie (1993, section 4.1) gives a detailed description of the data and results of different geostatistical analyses.

Format

A data frame with 85 observations on the following 3 variables:

lon relative longitude position (miles).

lat relative latitude position (miles).

head piezometric-head levels (feet above sea level).

Source

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.

References

Cressie, N. (1989) Geostatistics. *The American Statistician*, **43**, 197-202.

Cressie, N. (1993) *Statistics for Spatial Data*. New York. Wiley.

Examples

```
str(aquifer)
summary(aquifer)
with(aquifer, plot(lon, lat))
```

---

bin.den *Linear binning for density estimation*

Description

Creates a `bin.den`-`class` (gridded binned density) object with linear binning counts.

Usage

```
bin.den(x, nbin = NULL)

as.bin.den(object, ...)

## S3 method for class 'bin.data'
as.bin.den(object, ...)
```

Arguments

`x`	vector or matrix of covariates (e.g. spatial coordinates). Columns correspond with dimensions and rows with observations.
`nbin`	vector with the number of bins ($intervals = nbin + 1$) on each dimension.
`object`	(gridded data) used to select a method.
`...`	further arguments passed to or from other methods.

Details

If parameter `nbin` is not specified is set to `rep(25, ncol(x))`.

Value

Returns an S3 object of `class` `bin.den` (extends `data.grid`). A list with the following 3 components:

`binw`	vector or array (dimension `nbin`) with the bin counts (weights).
`grid`	a `grid.par-class` object with the grid parameters.
`data`	a list with a component `$x` with argument `x`.

See Also

data.grid, bin.data, locpol.

`binning` *Linear binning*

Description

Discretizes the data into a regular grid (computes a binned approximation) using the multivariate linear binning technique described in Wand (1994).

Usage

```
binning(x, y = NULL, nbin = NULL, set.NA = FALSE)
```

Arguments

x	vector or matrix of covariates (e.g. spatial coordinates). Columns correspond with covariates (coordinate dimension) and rows with data.
y	vector of data (response variable).
nbin	vector with the number of bins on each dimension.
set.NA	logical. If TRUE, sets binning cells without data to missing.

Details

If parameter nbin is not specified is set to rep(25, ncol(x)).

Setting set.NA = TRUE (equivalent to biny[binw == 0] <- NA) may be useful for plotting the binned averages $biny (the hat matrix should be handle with care when using locpol).

Value

If y != NULL, an S3 object of class bin.data (gridded binned data; extends bin.den) is returned. A data.grid object with the following 4 components:

biny	vector or array (dimension nbin) with the bin averages.
binw	vector or array (dimension nbin) with the bin counts (weights).
grid	a grid.par-class object with the grid parameters.
data	a list with 3 components:

- x argument x.
- y argument y.
- med (weighted) mean of the (binned) data.

If y == NULL, bin.den is called and a bin.den-class object is returned.

References

Wand M.P. (1994) Fast Computation of Multivariate Kernel Estimators. *Journal of Computational and Graphical Statistics*, **3**, 433-445.

See Also

data.grid, locpol, bin.den.

Examples

```
bin <- binning(earthquakes[, c("lon", "lat")], earthquakes$mag, nbin = c(30,30), set.NA = TRUE)

coorvs <- coordvalues(bin)
ns <- names(coorvs)                           # dimnames(bin$grid)
image( coorvs[[1]], coorvs[[2]], bin$biny, main = 'Binning averages', xlab = ns[1], ylab = ns[2])
with(earthquakes, points(lon, lat, pch = 20))
```

---

covar *Covariance values*

Description

Computes covariance values given a variogram model (or pseudo-covariances for unbounded variograms).

Usage

```
covar(x, h, sill = x$sill)
```

Arguments

x	variogram model (svarmod object).
h	vector (isotropic case) or matrix of lags values.
sill	variance $C(0) = \sigma^2$ or pseudo-sill (unbounded variograms).

Value

A vector of (pseudo) covariance values $C(h_i) = \sigma^2 - \gamma(h_i)$.

See Also

sv.

`data.grid` *Gridded data (S3 class "data.grid")*

Description

Defines data on a full regular (spatial) grid. Constructor function of the `data.grid-class`.

Usage

```
data.grid(..., grid = NULL)
```

Arguments

`...`	vectors or arrays of data with length equal to `prod(grid$n)`.
`grid`	a `grid.par-class` object (optional).

Details

If parameter `grid.par` is not specified it is set from first argument.

S3 "version" of the `SpatialGridDataFrame-class` of the **sp** package.

Value

Returns an object of `class` `data.grid`, a list with the arguments as components.

See Also

`grid.par`, `binning`, `locpol`.

`disc.sb` *Discretization nodes of a Shapiro-Botha variogram model*

Description

Computes the discretization nodes of a 'nonparametric' extended Shapiro-Botha variogram model, following Gorsich and Genton (2004), as the scaled roots of Bessel functions.

Usage

```
disc.sb(nx, dk = 0, rmax = 1)
```

Arguments

`nx`	number of discretization nodes.
`dk`	dimension of the kappa function.
`rmax`	maximum lag considered.

Details

If dk >= 1, the nodes are computed as:

$$x_i = q_i/rmax; i = 1, \ldots, n,$$

where q_i are the first n roots of $J_{(d-2)/2}$, J_p is the Bessel function of order p and $rmax$ is the maximum lag considered. The computation of the zeros of the Bessel function is done using the efficient algorithm developed by Ball (2000).

If dk == 0 (corresponding to a model valid in any spatial dimension), the nodes are computed so the gaussian variogram models involved have practical ranges:

$$r_i = (1 + 1.2(i-1))rmax/n; i = 1, \ldots, n.$$

References

Ball, J.S. (2000) Automatic computation of zeros of Bessel functions and other special functions. *SIAM Journal on Scientific Computing*, **21**, 1458-1464.

Gorsich, D.J. and Genton, M.G. (2004) On the discretization of nonparametric covariogram estimators. *Statistics and Computing*, **14**, 99-108.

See Also

kappasb, fitsvar.sb.iso.

Examples

```
disc.sb( 12, 1, 1.0)

nx <- 1
dk <- 0
x <- disc.sb(nx, dk, 1.0)
h <- seq(0, 1, length = 100)
plot(h, kappasb(x * h, 0), type="l", ylim = c(0, 1))
abline(h = 0.05, lty = 2)
```

---

earthquakes *Earthquake data*

Description

The data set consists of 1859 earthquakes (with magnitude above or equal to 2.0 in Richter's scale), which occurred from 25 November 1944 to 16 October 2013 in the northwest (NW) part of the Iberian Peninsula. The area considered is limited by the coordinates 41N-44N and 6W-10W, which contains the autonomic region of Galicia (Spain) and northern Portugal.

Format

A data frame with 1859 observations on the following 6 variables:

date Date and time (POSIXct format).

time Time (years since first event).

lon Longitude.

lat Latitude.

depth Depth (km).

mag Magnitude (Richter's scale).

Source

National Geographic Institute (IGN) of Spain:
http://www.ign.es/ign/layout/sismologiaObtencionDatosSismiscos.do.

References

Francisco-Fernandez M., Quintela-del-Rio A. and Fernandez-Casal R. (2012) Nonparametric methods for spatial regression. An application to seismic events, *Environmetrics*, **23**, 85-93.

Examples

```
str(earthquakes)
summary(earthquakes)
with(earthquakes, plot(lon, lat))
```

---

fitsvar.sb.iso *Fit an isotropic Shapiro-Botha variogram model*

Description

Fits a 'nonparametric' isotropic Shapiro-Botha variogram model by WLS through quadratic programming. Following Gorsich and Genton (2004), the nodes are selected as the scaled roots of Bessel functions (see disc.sb).

Usage

```
fitsvar.sb.iso(esv, dk = ncol(esv$data$x), nx = NULL,
  rmax = esv$grid$max, min.contrib = 15,
  method = c("cressie", "equal", "npairs", "gstat"),
  iter = 10, tol = sqrt(.Machine$double.eps))
```

Arguments

esv	pilot semivariogram estimate, a np.svar-class (or svar.bin) object. Typically an output of the function svarisonp.
dk	dimension of the kappa function (should be greater than or equal to the dimension of the spatial process `ncol(esv$data$x)`).
nx	number of discretization nodes. Defaults to `min(nesv - 1, 50)`, where `nesv` is the number of semivariogram estimates.
rmax	maximum lag considered in the discretization (range of the fitted variogram on output).
min.contrib	minimum number of contributing pairs (pilot estimates with a lower number are ignored).
method	string indicating the WLS fitting method to be used (e.g. `method = "cressie"`). See "Details" below.
iter	maximum number of interations of the WLS algorithm (used only if `method == "cressie"`).
tol	absolute convergence tolerance (used only if `method == "cressie"`).

Details

The fit is done using a (possibly iterated) weighted least squares criterion, minimizing:

$$WLS(\theta) = \sum_i w_i[(\hat{\gamma}(h_i)) - \gamma(\theta; h_i)]^2.$$

The different options for the argument `method` define the WLS algorithm used:

`"cressie"` The default method. The procedure is iterative, with $w_i = 1$ (OLS) used for the first step and with the weights recalculated at each iteration, following Cressie (1985), until convergence:

$$w_i = N(h_i)/\gamma(\hat{\theta}; h_i)^2,$$

where $N(h_i)$ is the (equivalent) number of contributing pairs in the estimation at lag h_i.

`"equal"` Ordinary least squares: $w_i = 1$.

`"npairs"` $w_i = N(h_i)$.

`"gstat"` The default fitting method in **gstat**: $w_i = N(h_i)/h_i^2$.

Function solve.QP of **quadprog** package is used to solve the quadratic programming problem. If `nx` and/or `dim(esv)` are large, this function may fail with error message "matrix D in quadratic function is not positive definite!".

Value

Returns the fitted variogram model, an object of class `sb.iso` (extends svarmod) with an additional component `fit` containing:

u	vector of lags/distances.
sv	vector of pilot semivariogram estimates.
fitted.sv	vector of fitted semivariances.

wls	value of the WLS objective function.
method	string indicating the WLS fitting method used.
iter	number of WLS iterations (if `method == "cressie"`).

References

Ball, J.S. (2000) Automatic computation of zeros of Bessel functions and other special functions. *SIAM Journal on Scientific Computing*, **21**, 1458-1464.

Cressie, N. (1985) Fitting variogram models by weighted least squares. *Mathematical Geology*, **17**, 563-586.

Cressie, N. (1993) *Statistics for Spatial Data*. New York. Wiley.

Fernandez Casal R., Gonzalez Manteiga W. and Febrero Bande M. (2003) Flexible Spatio-Temporal Stationary Variogram Models, *Statistics and Computing*, **13**, 127-136.

Gorsich, D.J. and Genton, M.G. (2004) On the discretization of nonparametric covariogram estimators. *Statistics and Computing*, **14**, 99-108.

Shapiro, A. and Botha, J.D. (1991) Variogram fitting with a general class of conditionally non-negative definite functions. *Computational Statistics and Data Analysis*, **11**, 87-96.

See Also

`svarmod.sb.iso`, `disc.sb`.

`grid.par` *Grid parameters (S3 class "grid.par")*

Description

Defines a full regular (spatial) grid. Constructor function of the `grid.par`-`class`.

Usage

```
grid.par(n, min, max = min + (n - 1) * lag,
  lag = (max - min)/(n - 1), dimnames = names(min))
```

Arguments

n	integer vector; number of nodes in each dimension.
min	vector; minimun coordinates values.
max	vector; maximum coordinates values (optional).
lag	vector; lag in each dimension (optional).
dimnames	character vector; names used to label the dimensions.

Details

All parameters must have the same length. Only one of the arguments `max` or `lag` must be specified.

S3 'version' on the `GridTopology-class` of the **sp** package.

Value

Returns an object of class `grid.par`, a list with the arguments as components and an additional component `$nd = length(n)`.

See Also

`data.grid`.

Examples

```
grid.par(n = c(100, 100), min = c(-10, 42), max = c(-7.5, 44))
grid.par(n = c(100, 100), min = c(-10, 42), lag = c(0.03, 0.02))
```

`h.cv` *Cross-validation methods for bandwidth selection*

Description

Selects the bandwidth of a local polynomial kernel (regression, density or variogram) estimator using (standart or modified) CV, GCV or CV-MASE criteria.

Usage

```
h.cv(bin, ...)

## S3 method for class 'bin.data'
h.cv(bin,
  objective = c("CV", "GCV", "MASE"), h.start = NULL,
  h.lower = NULL, h.upper = NULL, degree = 1,
  ncv = ifelse(objective == "GCV", 0, 1), cov.bin = NULL,
  DEalgorithm = FALSE, ...)

## S3 method for class 'bin.den'
h.cv(bin, h.start = NULL,
  h.lower = NULL, h.upper = NULL, degree = 1, ncv = 1,
  DEalgorithm = FALSE, ...)

hcv.data(bin, objective = c("CV", "GCV", "MASE"),
  h.start = NULL, h.lower = NULL, h.upper = NULL,
  degree = 1, ncv = ifelse(objective == "GCV", 0, 1),
  cov = NULL, DEalgorithm = FALSE, ...)
```

Arguments

`bin`	object used to select a method (binned data, binned density or binned semivariogram).
`...`	further arguments passed to or from other methods (e.g. parameters of the optimization routine).
`objective`	character; optimal criterion to be used ("CV", "GCV" or "MASE").
`h.start`	vector; initial values for the parameters (diagonal elements) to be optimized over. If `DEalgorithm == FALSE` (otherwise not used), defaults to `(3 + ncv) * lag`, where `lag = bin$grid$lag`.
`h.lower`	vector; lower bounds on each parameter (diagonal elements) to be optimized. Defaults to `(1.5 + ncv) * lag`.
`h.upper`	vector; upper bounds on each parameter (diagonal elements) to be optimized. Defaults to `dim(bin) * lag`.
`DEalgorithm`	logical; if TRUE, the differential evolution optimization algorithm in package **DEoptim** is used.
`ncv`	integer; determines the number of cells leaved out in each dimension. (0 to GCV considering all the data, > 0 to traditional or modified cross-validation). See "Details" bellow.
`cov.bin`	covariance matrix of the binned data. Defaults to identity.
`cov`	covariance matrix of the data. Defaults to identity (uncorrelated data).
`degree`	degree of the local polynomial used. Defaults to 1 (local linear estimation).

Details

Currently, only diagonal windows are supported.

If `ncv > 0`, estimates are computed by leaving out cells with indexes within the intervals $[x_i - ncv + 1, x_i + ncv - 1]$, at each dimension i, where x denotes the index of the estimation position. $ncv = 1$ corresponds with traditional cross-validation and $ncv > 1$ with modified CV (see e.g. Chu and Marron, 1991, for the one dimensional case). For standard GCV, set `ncv = 0` (the full data is used). For theoretical MASE, set `y = trend.teor`, `cov = cov.teor` and `ncv = 0`.

If `DEalgorithm == FALSE`, the `"L-BFGS-B"` method in `optim` is used.

Value

Returns a list containing the following 3 components:

`h`	the best (diagonal) bandwidth matrix found.
`value`	the value of the objective function corresponding to `h`.
`objective`	the criterion used.

References

Chu, C.K. and Marron, J.S. (1991) Comparison of Two Bandwidth Selectors with Dependent Errors. *The Annals of Statistics*, **19**, 1906-1918.

Francisco-Fernandez M. and Opsomer J.D. (2005) Smoothing parameter selection methods for non-parametric regression with spatially correlated errors. *Canadian Journal of Statistics*, **33**, 539-558.

See Also

locpol, locpolhcv, binning, np.svar.

Examples

```
bin <- binning(earthquakes[, c("lon", "lat")], earthquakes$mag, nbin = c(30,30))
hcv <- h.cv(bin)
lp <- locpol(bin, h = hcv$h)
## Equivalent to:
## lp <- locpolhcv(earthquakes[, c("lon", "lat")], earthquakes$mag, nbin = c(30,30))

coorvs <- coordvalues(lp)
ns <- names(coorvs)                              # dimnames(lp$grid)
image( coorvs[[1]], coorvs[[2]], lp$est, main = 'Smoothed magnitude',
            xlab = ns[1], ylab = ns[2])
contour(coorvs[[1]], coorvs[[2]], log(lp$est), add = TRUE)
with(earthquakes, points(lon, lat, pch = 20))

## Density estimation
hden <- h.cv(as.bin.den(bin))
den <- np.den(bin, h = hden$h)

image( coorvs[[1]], coorvs[[2]], log(den$est), main = 'Estimated log(density)',
            xlab = ns[1], ylab = ns[2])
contour(coorvs[[1]], coorvs[[2]], log(den$est), add = TRUE)
with(earthquakes, points(lon, lat, pch = 20))
```

---

interp *Fast linear interpolation of a regular grid*

Description

Computes a linear interpolation of multidimensional regularly gridded data.

Usage

```
interp(object, ...)

## S3 method for class 'grid.par'
interp(object, data, newx, ...)

## S3 method for class 'data.grid'
interp(object, data.ind = 1, newx,
   ...)

## S3 method for class 'locpol.bin'
predict(object, newx = NULL,
   hat.data = FALSE, ...)
```

Arguments

`object`	(gridded data) object used to select a method.
`...`	further arguments passed to or from other methods.
`data`	vector or array of data values.
`newx`	vector or matrix with the (irregular) locations to interpolate. Columns correspond with dimensions and rows with data.
`data.ind`	integer or character with the index or name of the data component.
`hat.data`	logical; if TRUE (and possible), the hat matrix corresponding to the (original) data is returned.

Details

`interp` methods are interfaces to the fortran routine `interp_data_grid` (in `grid_module.f90`).

`predict.locpol.bin` is an interface to the fortran routine `predict_lp` (in `lp_module.f90`).

Value

A list with two components:

`x`	interpolation locations.
`y`	interpolated values.

If `newx == NULL`, `predict.locpol.bin` returns the estimates (and optionally the hat matrix) corresponding to the data (otherwise `interp.data.grid` is called).

Note

Linear extrapolation is performed from the end nodes of the grid.

WARNING: May fail with missing values (especially if `object$locpol$ncv > 0`).

See Also

`interp.surface`.

---

kappasb — *Coefficients of an extended Shapiro-Botha variogram model*

Description

Computes the coefficients of an extended Shapiro-Botha variogram model.

Usage

```
kappasb(x, dk = 0)
```

Arguments

x	numeric vector (on which the kappa function will be evaluated).
dk	dimension of the kappa function.

Details

If dk >= 1, the coefficients are computed as:

$$\kappa_d(x) = (2/x)^{(d-2)/2}\Gamma(d/2)J_{(d-2)/2}(x)$$

where J_p is the Bessel function of order p.
If dk == 0, the coefficients are computed as:

$$\kappa_\infty(x) = e^{-x^2}$$

(corresponding to a model valid in any spatial dimension).
NOTE: some authors denote these functions as Ω_d.

Value

A vector with the coefficients of an extended Shapiro-Botha variogram model.

References

Shapiro, A. and Botha, J.D. (1991) Variogram fitting with a general class of conditionally non-negative definite functions. *Computational Statistics and Data Analysis*, **11**, 87-96.

See Also

svarmod.sb.iso, besselJ.

Examples

```
kappasb(seq(0, 6*pi, len = 10), 2)

curve(kappasb(x/5, 0), xlim = c(0, 6*pi), ylim = c(-1, 1), lty = 2)
for (i in 1:10) curve(kappasb(x, i), col = gray((i-1)/10), add = TRUE)
abline(h = 0, lty = 3)
```

---

locpol *Local polynomial estimation*

Description

Estimates a multidimensional regression function (and its first derivatives) using local polynomial kernel smoothing of linearly binned data.

Usage

```
locpol(x, ...)

## Default S3 method:
locpol(x, y, h = NULL, nbin = NULL,
  degree = 1 + as.numeric(drv), drv = FALSE,
  hat.bin = FALSE, ncv = 0, set.NA = FALSE, ...)

## S3 method for class 'bin.data'
locpol(x, h = NULL,
  degree = 1 + as.numeric(drv), drv = FALSE,
  hat.bin = FALSE, ncv = 0, ...)

## S3 method for class 'svar.bin'
locpol(x, h = NULL, degree = 1,
  drv = FALSE, hat.bin = FALSE, ncv = 0, ...)

## S3 method for class 'bin.den'
locpol(x, h = NULL,
  degree = 1 + as.numeric(drv), drv = FALSE, ncv = 0,
  ...)

locpolhcv(x, y, nbin = NULL,
  objective = c("CV", "GCV", "MASE"),
  degree = 1 + as.numeric(drv), drv = FALSE,
  hat.bin = FALSE, set.NA = FALSE,
  ncv = ifelse(objective == "GCV", 0, 1), cov = NULL,
  ...)
```

Arguments

`x`	a (data) object used to select a method.
`...`	further arguments passed to or from other methods (e.g. to `hcv.data`).
`y`	vector of data (response variable).
`h`	(full) bandwidth matrix (controls the degree of smoothing).
`nbin`	vector with the number of bins on each dimension.
`degree`	degree of the local polynomial used. Defaults to 1 (local linear estimation).
`drv`	logical; if TRUE, the matrix of estimated first derivatives is returned.
`hat.bin`	logical; if TRUE, the hat matrix of the binned data is returned.
`ncv`	integer; determines the number of cells leaved out in each dimension. Defaults to 0 (the full data is used) and it is not normally changed by the user in this setting. See "Details" below.
`set.NA`	logical. If `TRUE`, sets binning cells without data to missing.
`objective`	character; optimal criterion to be used ("CV", "GCV" or "MASE").
`cov`	covariance matrix of the data. Defaults to identity (uncorrelated data).

Details

Standard generic function with a default method (interface to the fortran routine `lp_raw`), in which argument `x` is a vector or matrix of covariates (e.g. spatial coordinates).

If parameter `nbim` is not specified is set to `rep(25, ncol(x))`.

A multiplicative triweight kernel is used to compute the weights.

If `ncv > 0`, estimates are computed by leaving out cells with indexes within the intervals $[x_i - ncv + 1, x_i + ncv - 1]$, at each dimension i, where x denotes the index of the estimation position. $ncv = 1$ corresponds with traditional cross-validation and $ncv > 1$ with modified CV (see e.g. Chu and Marron, 1991, for the one dimensional case).

Setting `set.NA = TRUE` (equivalent to `biny[binw == 0] <- NA`) may be useful for plotting the binned averages `$biny` (the hat matrix should be handle with care).

`locpolhcv` calls `hcv.data` to obtain an "optimal" bandwith (additional arguments `...` are passed to this function). Argument `ncv` is only used here at bandwith selection stage (estimation is done with all the data).

Value

Returns an S3 object of class `locpol.bin` (locpol + bin data + grid par.). A `bin.data` object with the additional (some optional) 3 components:

`est` vector or array (dimension `nbin`) with the local polynomial estimates.

`locpol` a list with 7 components:

- `degree` degree of the polinomial.
- `h` bandwidth matrix.
- `rm` residual mean.
- `rss` sum of squared residuals.
- `ncv` number of cells ignored in each direction.
- `hat` (if requested) hat matrix of the binned data.
- `nrl0` (if appropriate) number of cells with data (`binw > 0`) and missing estimate (`est == NA`).

`deriv` (if requested) matrix of first derivatives.

`locpol.svar.bin` returns an S3 object of class `np.svar` (locpol semivar + bin semivar + grid par.).

`locpol.bin.den` returns an S3 object of class `np.den` (locpol den + bin den + grid par.).

References

Chu, C.K. and Marron, J.S. (1991) Comparison of Two Bandwidth Selectors with Dependent Errors. *The Annals of Statistics*, **19**, 1906-1918.

Rupert D. and Wand M.P. (1994) Multivariate locally weighted least squares regression. *The Annals of Statistics*, **22**, 1346-1370.

See Also

`binning`, `data.grid`, `svarisonp`, `svar.bin`, `np.den`, `bin.den`.

Examples

```
bin <- binning(earthquakes[, c("lon", "lat")], earthquakes$mag, nbin = c(30,30))
hcv <- h.cv(bin)
lp <- locpol(bin, h = hcv$h)
## Equivalent to:
## lp <- locpolhcv(earthquakes[, c("lon", "lat")], earthquakes$mag, nbin = c(30,30))

coorvs <- coordvalues(lp)
ns <- names(coorvs)                           # dimnames(lp$grid)
image( coorvs[[1]], coorvs[[2]], lp$est, main = 'Smoothed magnitude',
            xlab = ns[1], ylab = ns[2])
contour(coorvs[[1]], coorvs[[2]], log(lp$est), add = TRUE)
with(earthquakes, points(lon, lat, pch = 20))
```

`np.den` *local polynomial density estimation*

Description

Estimates a multidimensional probability density function (and its first derivatives) using local polynomial kernel smoothing of linearly binned data.

Usage

```
np.den(x, ...)

## Default S3 method:
np.den(x, nbin = NULL, h = NULL,
   degree = 1 + as.numeric(drv), drv = FALSE, ncv = 0,
   ...)

## S3 method for class 'bin.den'
np.den(x, h = NULL,
   degree = 1 + as.numeric(drv), drv = FALSE, ncv = 0,
   ...)

## S3 method for class 'bin.data'
np.den(x, h = NULL,
   degree = 1 + as.numeric(drv), drv = FALSE, ncv = 0,
   ...)

## S3 method for class 'svar.bin'
np.den(x, h = NULL,
   degree = 1 + as.numeric(drv), drv = FALSE, ncv = 0,
   ...)
```

Arguments

x	a (data) object used to select a method.
...	further arguments passed to or from other methods.
nbin	vector with the number of bins on each dimension.
h	(full) bandwidth matrix (controls the degree of smoothing).
degree	degree of the local polynomial used. Defaults to 1 (local linear estimation).
drv	logical; if TRUE, the matrix of estimated first derivatives is returned.
ncv	integer; determines the number of cells leaved out in each dimension. Defaults to 0 (the full data is used) and it is not normally changed by the user in this setting. See "Details" below.

Details

Standard generic function with a default method (interface to the fortran routine `lp_data_grid`), in which argument `x` is a vector or matrix of covariates (e.g. spatial coordinates). In this case, the data are binned (calls `bin.den`) and the local fitting procedure is applied to the scaled bin counts (calls `np.den.bin.den`).

If parameter `nbim` is not specified is set to `rep(25, ncol(x))`.

A multiplicative triweight kernel is used to compute the weights.

If `ncv > 1`, estimates are computed by leaving out cells with indexes within the intervals $[x_i - ncv + 1, x_i + ncv - 1]$, at each dimension i, where x denotes the index of the estimation position.

Value

Returns an S3 object of class `np.den` (locpol den + bin den + grid par.). A `bin.den` object with the additional (some optional) 3 components:

est	vector or array (dimension `nbin`) with the local polynomial density estimates.
locpol	a list with 6 components: • `degree` degree of the polinomial. • `h` bandwidth matrix. • `rm` residual mean (of the escaled bin counts). • `rss` sum of squared residuals (of the escaled bin counts). • `ncv` number of cells ignored (in each dimension).
deriv	(if requested) matrix of first derivatives.

References

Wand, M.P. and Jones, M.C. (1995) *Kernel Smoothing*. Chapman and Hall, London.

See Also

`bin.den`, `binning`, `data.grid`.

Examples

```
bin <- binning(earthquakes[, c("lon", "lat")], earthquakes$mag, nbin = c(30,30))
hden <- h.cv(as.bin.den(bin))
den <- np.den(bin, h = hden$h)
## Equivalent to:
## den <- np.den(earthquakes[, c("lon", "lat")], h = hden$h, nbin = c(30,30))

coorvs <- coordvalues(bin)
ns <- names(coorvs)
image( coorvs[[1]], coorvs[[2]], log(den$est), main = 'Estimated log(density)',
            xlab = ns[1], ylab = ns[2])
contour(coorvs[[1]], coorvs[[2]], log(den$est), add = TRUE)
with(earthquakes, points(lon, lat, pch = 20))
```

---

np.svar — *Local polynomial estimation of the semivariogram*

Description

Estimates a multidimensional semivariogram (and its first derivatives) using local polynomial kernel smoothing of linearly binned semivariances.

Usage

```
np.svar(x, ...)

## Default S3 method:
np.svar(x, y, h = NULL, maxlag = NULL,
  nlags = NULL, minlag = maxlag/nlags, degree = 1,
  drv = FALSE, hat.bin = FALSE, ncv = 0, ...)

## S3 method for class 'svar.bin'
np.svar(x, h = NULL, degree = 1,
  drv = FALSE, hat.bin = FALSE, ncv = 0, ...)

svarisonp(x, y, h = NULL, maxlag = NULL, nlags = NULL,
  minlag = maxlag/nlags, degree = 1, drv = FALSE,
  hat.bin = FALSE, ncv = 0, ...)

svarisohcv(x, y, maxlag = NULL, nlags = NULL,
  minlag = maxlag/nlags,
  objective = c("CV", "GCV", "MASE"),
  ncv = ifelse(objective == "GCV", 0, 1), cov.bin = NULL,
  ...)
```

Arguments

`x`	a (data) object used to select a method.
`...`	further arguments passed to or from other methods.
`y`	vector of data (response variable).
`maxlag`	maximum lag. Defaults to 55% of largest lag.
`nlags`	number of lags. Defaults to 101.
`minlag`	minimun lag.
`hat.bin`	logical; if TRUE, the hat matrix of the binned semivariances is returned.
`cov.bin`	covariance matrix of the binned semivariances. Defaults to identity.
`h`	(full) bandwidth matrix (controls the degree of smoothing).
`degree`	degree of the local polynomial used. Defaults to 1 (local linear estimation).
`drv`	logical; if TRUE, the matrix of estimated first derivatives is returned.
`ncv`	integer; determines the number of cells leaved out in each dimension. Defaults to 0 (the full data is used) and it is not normally changed by the user in this setting. See "Details" below.
`objective`	character; optimal criterion to be used ("CV", "GCV" or "MASE").

Details

Currently, only isotropic semivariogram estimation is supported.

If parameter `nlags` is not specified is set to `101`.

A multiplicative triweight kernel is used to compute the weights.

Value

Returns an S3 object of class `np.svar` (locpol svar + binned svar + grid par.), extends `svar.bin`, with the additional (some optional) 3 components:

`est`	vector or array with the local polynomial semivariogram estimates.
`locpol`	a list of 6 components: • `degree` degree of the local polinomial used. • `h` smoothing matrix. • `rm` mean of residual semivariances. • `rss` sum of squared residual semivariances. • `ncv` number of cells ignored in each direction. • `hat` (if requested) hat matrix of the binned semivariances. • `nrl0` (if appropriate) number of cells with `binw > 0` and `est == NA`.
`deriv`	(if requested) matrix of estimated first semivariogram derivatives.

References

Fernandez Casal R., Gonzalez Manteiga W. and Febrero Bande M. (2003) Space-time dependency modeling using general classes of flexible stationary variogram models, *J. Geophys. Res.*, **108**, 8779, doi:10.1029/2002JD002909.

Garcia-Soidan P.H., Gonzalez-Manteiga W. and Febrero-Bande M. (2003) Local linear regression estimation of the variogram, *Stat. Prob. Lett.*, **64**, 169-179.

See Also

`svar.bin`, `data.grid`, `locpol`.

---

npsp-geoR   *Interface to package "geoR"*

Description

Utilities to interact with the **geoR** package.

Usage

```
as.variogram(x, ...)

## S3 method for class 'svar.bin'
as.variogram(x, ...)

## S3 method for class 'np.svar'
as.variogram(x, ...)

as.variomodel(m, ...)

## S3 method for class 'svarmod'
as.variomodel(m, ...)
```

Arguments

`x`	semivariogram estimate (e.g. `svar.bin` or `np.svar` object).
`m`	variogram model (e.g. `svarmod` object).
`...`	further arguments passed to or from other methods.

Details

`as.variogram` tries to convert a semivariogram estimate $\hat{\gamma}(h_i)$ to an object of the (not fully documented) **geoR**-class `variogram` (see e.g. `variog`).

`as.variomodel` tries to convert a semivariogram model $\gamma(pars; h)$ to an object of the **geoR**-class `variomodel` (see e.g. `variofit`).

See Also

variog, variofit, variomodel, svar.bin, np.svar.

---

npsp-gstat *Interface to package "gstat"*

Description

Utilities to interact with the **gstat** package.

Usage

```
as.vgm(x, ...)

## S3 method for class 'variomodel'
as.vgm(x, ...)

## S3 method for class 'svarmod'
as.vgm(x, ...)

vgm.tab.svarmod(x, h = seq(0, x$range, length = 1000),
  sill = x$sill, ...)

## S3 method for class 'sb.iso'
as.vgm(x,
  h = seq(0, x$range, length = 1000), sill = x$sill, ...)
```

Arguments

x	variogram model object (used to select a method).
...	further arguments passed to or from other methods.
h	vector of lags at which the covariogram is evaluated.
sill	sill of the covariogram (or pseudo-sill).

Details

Tries to convert a variogram object to vgm (variogramModel-class of **gstat** package). S3 generic function.

as.vgm.variomodel tries to convert an object of class variomodel defined in **geoR** (interface to as.vgm.variomodel defined in **gstat**).

vgm.tab.svarmod converts a svarmod object to a variogramModel-class object of type "Tab" (one-dimensional covariance table).

as.vgm.sb.iso is an alias of vgm.tab.svarmod.

See Also

vgm, svarmod.

`sv` *Evaluate a semivariogram model*

Description

Evaluates an `svarmod` object `x` at lags `h` (S3 generic function).

Usage

```
sv(x, h, ...)

## Default S3 method:
sv(x, h, ...)

## S3 method for class 'svarmod'
sv(x, h, ...)

## S3 method for class 'sb.iso'
sv(x, h, ...)
```

Arguments

`x`	variogram model (`svarmod` object).
`h`	vector (isotropic case) or matrix of lags values.
`...`	further arguments passed to or from other methods.

Value

A vector of semivariance values $\gamma(h_i)$.

See Also

`covar`

`svar.bin` *Linear binning of semivariances*

Description

Creates a `svar.bin` (binned semivar. + grid parameters) object with linearly binned semivariances.

Usage

```
svar.bin(x, ...)

## Default S3 method:
svar.bin(x, y, maxlag = NULL,
  nlags = NULL, minlag = maxlag/nlags,
  estimator = c("classical", "modulus"), ...)

svariso(x, y, maxlag = NULL, nlags = NULL,
  minlag = maxlag/nlags,
  estimator = c("classical", "modulus"), ...)
```

Arguments

`x`	a (data) object used to select a method.
`...`	further arguments passed to or from other methods.
`estimator`	character, estimator name (e.g. "classical"). See "Details" below.
`y`	vector of data (response variable).
`maxlag`	maximum lag. Defaults to 55% of largest lag.
`nlags`	number of lags. Defaults to 101.
`minlag`	minimun lag.

Details

Currently, only isotropic semivariogram estimation is supported.

If parameter `nlags` is not specified is set to `101`.

Value

Returns an S3 object of class `svar.bin` (extends `bin.data`), a `data.grid` object with the following 4 components:

`biny` array (dimension `nlags`) with the binned semivariances.

`binw` array (dimension `nlags`) with the bin counts (weights).

`grid` a `grid.par-class` object with the grid parameters.

`data` a list with 3 components:

- `x` argument `x`.
- `y` argument `y`.
- `med` (weighted) mean of the (binned) semivariances.

`svar` a list of 2 components:

- `type` character, type of estimation (e.g. "isotropic").
- `estimator` character, estimator name (e.g. "classical").

See Also

`svarisonp`, `np.svar`, `data.grid`, `binning`, `locpol`.

svarmod	*Define a (semi)variogram model*

Description

Defines a variogram model specifying the parameter values. Constructor function of the `svarmod-class`.

Usage

```
svarmod(model, type = "isotropic", par = NA,
  nugget = NULL, sill = NULL, range = NULL)

svarmod.sb.iso(dk, x, z, nu, range, sill = nu)

svarmodels(type = "isotropic")
```

Arguments

`model`	string indicating the variogram family (see Details below).
`type`	string indicating the type of variogram, e.g. "isotropic".
`par`	vector of variogram parameters.
`nugget`	nugget value c_0.
`sill`	variance σ^2 or sill of the variogram (NA for unbounded variograms).
`range`	range (practical range or scale parameter) of the variogram (NA for unbounded variograms; maybe a vector for anisotropic variograms).
`dk`	dimension of the kappa function.
`x`	discretization nodes.
`z`	jumps (of the spectral distibution) at the discretization nodes.
`nu`	parameter ν_0 (can be thought of as the sill).

Value

`svarmod` returns an `svarmod-class` object, a list with function arguments as components.

`svarmod.sb.iso` returns an S3 object of `class` `sb.iso` (extends `svarmod`) corresponding to a 'non-parametric' isotropic Shapiro-Botha model.

`svarmodels` returns a named character vector with the available models of the corresponding `type` (when appropriate, component values could be used as `cov.model` argument in **geoR** routines and component names as `model` argument in **gstat** routines).

Note

`svarmod` does not check the consistency of the parameter values.

References

Shapiro, A. and Botha, J.D. (1991) Variogram fitting with a general class of conditionally non-negative definite functions. *Computational Statistics and Data Analysis*, **11**, 87-96.

See Also

`sv`, `covar`.

Index