Package ‘scpm’

June 30, 2017

Type Package

Title An R Package for Spatial Smoothing

Version 1.0-2

Date 2017-06-30

Author Mario A. Martínez Araya [aut,cre] and Jianxin Pan [ths]

Maintainer Mario A. Martínez Araya <mma@mariomartinezaraya.com>

Description Group of functions for spatial smoothing using cubic splines and variogram maximum likelihood estimation. Also allow the inclusion of linear parametric terms and change-points for segmented smoothing splines models.

License GPL (>= 2)

Depends R (>= 2.10), stats, Matrix, RandomFields, interp

Imports geoR, methods, rgl, lattice, mvtnorm, MASS, graphics, grDevices

Enhances fields

Encoding latin1

Note This package allows to estimate spatial spline models to small to middle size datasets, for larger datasets it will depend on the memory of the system. Therefore use it carefully. Currently under development to incorporate additive components.

NeedsCompilation no

Repository CRAN

Date/Publication 2017-06-30 13:08:43 UTC

R topics documented:

`scpm-package` *'An R Package for Spatial Smoothing'*

Description

Group of functions for spatial smoothing using cubic splines and variogram maximum likelihood estimation. Also allow the inclusion of linear parametric terms and change-points for segmented smoothing splines models.

Note

This package allows to estimate spatial spline models to small to middle size datasets, for larger datasets it will depend on the memory of the system. Therefore use it carefully. Currently under development to incorporate additive components.

Author(s)

Mario A. Martínez Araya [aut,cre] and Jianxin Pan [ths]

Examples

```
require(geoR)
data(landim1, package = "geoR")
d <- as.sss(landim1, coords = NULL, coords.col = 1:2, data.col = 3:4)
##fitting spatial linear model with response A and covariate B
##Gneiting covariance function in the errors
#m0 <- scp(A ~ linear(~ B), data = d, model = "RMgneiting")
##adding a bivariate cubic spline based on the coordinates
#m1 <- scp(A ~ linear(~ B) + s2D(penalty = "cs"), data = d, model = "RMgneiting")
##plotting observed and estimated field from each model
#par(mfrow=c(2,2))
#plot(m0, what = "obs", type = "persp", main = "Model null - y")
#plot(m0, what = "fit", type = "persp", main = "Model null - fit")
#plot(m1, what = "obs", type = "persp", main = "Model alternative - y")
#plot(m1, what = "fit", type = "persp", main = "Model alternative - fit")
##plotting the estimated semivariogram from each model
#par(mfrow=c(1,2))
#Variogram(m0,main="Semivariogram - model null", ylim = c(0,0.7))
#Variogram(m1,main="Semivariogram - model alternative", ylim = c(0,0.7))
```

```
##summary of the estimated coefficients
#summary(m0)
#summary(m1)
##some information criteria
#AIC(m0)
#AIC(m1)
#AICm(m0)
#AICm(m1)
#AICc(m0)
#AICc(m1)
#BIC(m0)
#BIC(m1)
```

`A1. Create sss data` *Convert an object to the class* ‘`sss`’ *for spatial smoothing splines*

Description

Create a matrix or data.frame to a valid dataset of class ‘`sss`’ for spatial smoothing splines. Those dataset can be used later by functions s2D for tensor product (natural) cubic splines or p-splines, and scp for estimating spatial smoothing splines models.

Usage

```
as.sss(X, coords, coords.col, data.col, ...)
create.sss(coords, data, ...)
is.sss(x)
sss2df(x)
```

Arguments

`X`	a matrix or data-frame. Every row must correspond to a point location in a two-dimensional space (coordinates). Coordinates columns can be included in `X` or defined separately using the argument `coords`. Some columns can also correspond to variables measured at the different point locations.
`coords`	two-columns numeric matrix of coordinates (optional).
`data`	a data-frame containing the variables measured at the locations given by `coords`.
`coords.col`	numeric vector. The number of columns in `X` that contains the coordinates.
`data.col`	numeric vector. The number of columns in `X` that contains variables measured at the points locations.
`...`	slots elements to create a new `sss` dataset. Required slots are `data`, `coords`, `grid`, `knots`, `W`, `contract` (to be discarded in the future), and `regular`. See *Value* for an explanation about each slot requirements.
`x`	an object to check validity as member of class `sss`.

Value

data a data-frame containing the variables measured at the locations given by `coords`.

coords a matrix containing the two columns of observed coordinates for the data.

grid a grid matrix containing the two columns of coordinates.

knots a named list with the design points (knots) in every coordinate. Equivalent to a `grid.list` object.

W a spatial incidence matrix. If `contract=TRUE` it is W_{ij}, otherwise W_{ji}.

contract logical. The same value as the argument `contract`.

regular logical. If the coordinates are observed at regular points it is `TRUE`, `FALSE` otherwise (missing coordinates in any direction).

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

A2. Define linear terms

Linear components of the mean of the model

Description

Define parametric components of the mean as linear terms.

Usage

```
linear(formula, data = NULL, contrasts = NULL, intercept = FALSE)
```

Arguments

`formula`	formula expression. A formula expression as described in formula.
`data`	data frame. Where to search for the covariates?
`contrasts`	character. A contrast method for factor covariates. Default to '`contr.treatment`'.
`intercept`	logical. `TRUE` to include an intercept term, `FALSE` otherwise (default).

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

A3. Define unknown changes
Changes in the pattern of response

Description

Define unknown changes in the pattern of response to be estimated.

Usage

```
cp(x, psi, data = NULL, groups = NULL, contrasts = NULL,
        only.UV = FALSE)
```

Arguments

x	numeric vector. Covariate over which range define unknown change-points.
psi	numeric vector. Starting values for the change-points.
data	data frame. Where to search for the covariate?
groups	not used at the moment. To be implemented.
contrasts	character. A contrast method for factor covariates in groups. Default to 'contr.treatment'.
only.UV	logical. Not required.

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

A4. Define bivariate Spline
Bivariate spline

Description

Define a bivariate spline using tensor products or thin plate splines.

Usage

```
s2D(data = NULL, penalty = c("none", "cs", "ps", "tps"),
        is.X = c("none", "tensor", "tps"), intercept = TRUE,
        ps.order = 2, aniso.angle = 0, aniso.ratio = 1,
        env = .GlobalEnv, ...)
```

Arguments

`data`	sss object. Data of class sss generated by as.sss or create.sss.
`penalty`	character. Type of spline to use and penalty to define. One of '`cs`' or '`ps`' for cubic splines based on tensor products, '`tps`' for thin plate splines or '`none`'. '`cs`' define the penalty based on roughness matrices of natural cubic splines while '`ps`' define the penalty based on differences of order `ps.order`. See scp. If `penalty="none"` then no spline nor penalty are defined and the model for the spatial surface is defined by `is.X`.
`is.X`	character. Model for the spatial surface. One of '`tensor`', '`tps`' or '`none`'. Only required if `penalty="none"`. See details.
`intercept`	logical. Define whether to include an intercept or not. Default to `TRUE`.
`ps.order`	integer. Order for differences if `penalty = "ps"`.
`aniso.angle`	numeric. Angle for geometric anisotropy.
`aniso.ratio`	numeric. Ratio between $[0, 1]$ for geometric anisotropy.
`env`	environment. Where to search for data if `data=NULL`.
`...`	additional arguments. Not required.

Details

Note that `is.X` is only needed if `penalty="none"`. By defining `is.X="none"` it only define an intercept (if `intercept=TRUE`), `is.X="tps"` defines an intercept, coordinate 1, and coordinate 2 as covariates, while `is.X="tensor"` defines also the interaction coordinate 1*coordinate 2. See scp.

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

`A5. Estimate the model` *Spatial smoothing with unknown changes in the pattern of response*

Description

Fit a spatial semiparametric model based on splines including unknown changes in the pattern of response.

Usage

```
scp(formula, data, initial = NULL, contrasts = NULL,
        model = exponential, fix.nugget = FALSE, fix.kappa = FALSE,
        nugget.tol = 1e-15, angle = 0, ratio = 1, use.reml = FALSE,
        use.profile = TRUE, chMaxiter = 20, control = list())
```

Arguments

`formula`	formula. An expression to specify the model to fit. See *2. Mean model*.
`data`	sss class object. A dataset object generated by any of the commands as.sss or create.sss.
`initial`	named list. The starting values for the covariance parameters of the model. If `initial=NULL` then it is used an internal grid search to define the starting values.
`contrasts`	character. A contrast method for factor covariates. Default to '`contr.treatment`'.
`model`	character. Name of the semivariogram model to estimate for the spatial dependence. See *Semivariogram Model*.
`fix.nugget`	logical or numeric. If `FALSE` the nugget τ^2 is estimated. If `fix.nugget` is a numeric value then the nugget τ^2 is set to the value defined for `fix.nugget`.
`fix.kappa`	logical or numeric vector. If `FALSE` the parameters κ are estimated. If `fix.kappa` is a numeric vector then κ is set to the values of the vector defined for `fix.kappa`.
`nugget.tol`	numeric. Threshold for microscale spatial variations to define the nugget effect. Default to `1.0e-15`. Do not modify unless know what is being doing.
`angle`	numeric. Angle for geometric anisotropy. Note that this overwrites any specification for `aniso.angle` in s2D.
`ratio`	numeric. Ratio between $[0,1]$ for geometric anisotropy. Note that this overwrites any specification for `aniso.ratio` in s2D.
`use.reml`	logical. For using REML estimation set to `TRUE`, for ML estimation set to `FALSE` (default).
`use.profile`	logical. For profiling set to `TRUE` (default).
`chMaxiter`	integer. Maximum number of iterations for the loop estimating changes in the pattern of response.
`control`	named list. Options to control the optimization. See argument `control` in command optim.

1. Semiparametric model

Assume that the response variable admit the trend surface model

$$Y(s) = a^T b + g(s) + \epsilon(s)$$

where a is a known vector of covariates and b their coefficients; $g(s)$ is a deterministic bivariate spline and $\epsilon(s)$ is a Gaussian spatial process (GSP) with mean zero and covariance depending only on the distance h and given by $Cov(\epsilon(s+h), \epsilon(s))$. This model is also called a *trend surface model*. Given n observed locations $s_1, \ldots, s_n \in S \subset \Re^2$ in a two-dimensional space, then the model is

$$Y = Ab + g + \epsilon$$

where $Y = (Y(s_1), \ldots, Y(s_n))^T$, A is the known matrix of covariates, $g = (g(s_1), \ldots, g(s_n))^T$ and $\epsilon = (\epsilon(s_1), \ldots, \epsilon(s_n))^T$. The covariance matrix is given by $Cov(\epsilon, \epsilon) = \Sigma = \sigma^2 R$ with R a valid correlation matrix. Thus $Y \sim N_n(\mu, \Sigma)$ where $\mu = Ab + g$ and the likelihood function is $L(b, g, \sigma^2, \theta; Y) = (2\pi)^{-n/2} |\Sigma|^{-1/2} \exp\{-\frac{1}{2}(Y-\mu)^T \Sigma^{-1} (Y-\mu)\}$ with θ the parameters that define the correlation matrix R.

2. Mean model

It can be defined by the commands:

`linear` that defines the covariates in the matrix A. Note that more than one `linear` command can be defined. See linear.

`cp` defines changes in the pattern of response by including the covariates $(z_d - \psi_d^{(0)}) \times 1\{z_d > \psi_d^{(0)}\}$ and $-1\{z_d > \psi_d^{(0)}\}$ for $d = 1, \ldots, G$ into the matrix A. Note that more than one `cp` command can be defined. See cp.

`s2D` define the bivariate spline g. Note that only one `s2D` command can be defined. See s2D.

3. Covariance model and nugget effect

Given a distance h define $u = ||T^{1/2}_{\texttt{angle, ratio}} h|| = (h^T T_{\texttt{angle,ratio}} h)^{1/2} \in \Re$ where $T_{\texttt{angle, ratio}}$ is a rotation matrix for geometric anisotropy. The errors are given by the process $\epsilon(s) = \eta(s) + \xi(s)$ where ξ is a GSP with mean zero and covariance

$$\begin{aligned} Cov(\xi(s), \xi(s+h)) &= C_\xi(u; \phi, \kappa) \\ &= \sigma_0^2 \rho_\xi(u; \phi, \kappa) \end{aligned}$$

with $\rho_\xi(u; \phi, \kappa)$ the correlation function; and η is a nugget effect with covariance

$$\begin{aligned} Cov(\eta(s), \eta(s+h)) &= C_\eta(u; \tau^2, \texttt{tol.nugget}) \\ &= \tau^2 \rho_\eta(u; \texttt{tol.nugget}) \end{aligned}$$

with correlation function $\rho_\eta(u; \texttt{tol.nugget}) = 1\{u < \texttt{tol.nugget}\}$. Therefore the covariance of the process ϵ is given by

$$\begin{aligned} Cov(\epsilon(s), \epsilon(s+h)) &= C_\epsilon(u; \sigma^2, \theta, \texttt{tol.nugget}) \\ &= \sigma^2 \rho_\epsilon(u; \theta, \texttt{tol.nugget}) \end{aligned}$$

with correlation function given by

$$\rho_\epsilon(u; \theta, \texttt{tol.nugget}) = (1 - \rho_*)\rho_\eta(u; \texttt{tol.nugget}) + \rho_* \rho_\xi(u; \phi, \kappa)$$

where $\theta = (\rho_*, \phi, \kappa)^T$ are the parameters with $\rho_* = \sigma_0^2/\sigma^2$, $\sigma^2 = \tau^2 + \sigma_0^2$, and `tol.nugget` is the argument that controls the largest distance at which micro-scale variations can affect the observed outcome. By default `tol.nugget` is set to `1.0e-15`. The parameters ϕ, κ define the correlation function of the process ξ with ϕ usually called the *range parameter* and κ depending on the model selected. The semivariogram can be expressed as

$$\gamma_\epsilon(u; \sigma^2, \theta, \texttt{tol.nugget}) = \sigma^2(1 - \rho_\epsilon(u; \theta, \texttt{tol.nugget}))$$

where τ^2 is the nugget effect, σ^2 is the sill, and σ_0^2 is the partial sill. Note that when `angle = 0` and `ratio = 1` the matrix $T_{\texttt{angle,ratio}}$ is an identity matrix and $u = h$ so the correlation $\rho_\epsilon(u; \theta, \texttt{tol.nugget})$ is isotropic. Use different values for `angle` and `ratio` to define a geometric anisotropic correlation function. Then the covariance matrix $\Sigma = \sigma^2 R$ where R is the correlation matrix originated from $\rho_\epsilon(u; \theta, \texttt{tol.nugget})$. It is possible to define the argument `model=name` where `name` is one of the following: '`matern`', '`powered.exponential`', '`spherical`', '`wave`', '`exponential`', '`gaussian`', '`cubic`', '`circular`', '`gencauchy`', '`cauchy`', '`RMmatern`', '`RMwhittle`', '`RMgneiting`', and '`RMnugget`'. For `.semiVar` one of '`matern`', '`gaussian`', '`exponential`', '`power`', '`cubic`', '`penta.spherical`', '`spherical`', '`wave`', '`sin.hole`', '`pure.nugget`' and '`identity`'. By default the covariance model is set to '`exponential`' with `angle=0` and `ratio=1`.

4. Penalized maximum likelihood estimation

Estimation can be performed by maximisation with respect to b, g, σ^2, θ, and α of the penalized log likelihood

$$\ell_p(b, g, \sigma^2, \theta, \alpha) = \log(L(b, g, \sigma^2, \theta; Y)) - \frac{1}{2\sigma^2} J_\alpha(g)$$

where $J_\alpha(g) = g^T Q_\alpha g$ is the penalty and Q_α is the roughness matrix.

5. Penalties

Depending on the type of spline assumed for g the penalty is defined differently depending on the roughness matrix Q_α which is given by:

Tensor product spline. Given $\tau_{1,1}, \ldots, \tau_{1,K_1}$ and $\tau_{2,1}, \ldots, \tau_{2,K_2}$ the design points in each coordinate then

$$Q_\alpha = \alpha_1 I_{K_2} \otimes Q_1 + \alpha_2 Q_2 \otimes I_{K_1}$$

where Q_1, Q_2 are unidimensional roughness matrices from the design points in each coordinate and α_1, α_2 are smoothing parameters in each direction.

Thin plate spline. Given the n locations, $Q_\alpha = \alpha E$ where α is the smoothing parameter and the $n \times n$ matrix E has elements $E_{i,j} = \vartheta(||s_i - s_j||)$ for $i, j = 1, \ldots, n$ where

$$\vartheta(u) = \begin{cases} \frac{1}{16\pi} \times u^2 \log(u^2) & , u > 0 \\ 0 & , \text{otherwise.} \end{cases}$$

6. Mixed model representation

The spline can be written as $g = X\beta + Zr$ with β and r the coefficients and X and Z design matrices conveniently defined. Then for the observed responses the model can be expressed as a the mixed model

$$Y = Ab + X\beta + Zr + \epsilon$$

where $r \sim Normal(0, I_V)$ with V the number of columns in Z. Then, $Y \sim N_n(\mu_m, \Sigma)$ where $\mu_m = Ab + X\beta$ and $\Sigma = \sigma^2 R$; and $Y|r \sim N_n(\mu, V)$ where $\mu = Ab + X\beta + Zr$ and $V = ZZ^T + \Sigma$. Let us denote $\vartheta = (b, \beta, \sigma^2, \theta, \alpha)^T$, then the conditional log-likelihood of the model is given by

$$\ell(\vartheta|r) \propto -\frac{1}{2} \left\{ \log|\Sigma| + (Y - \mu)^T \Sigma^{-1} (Y - \mu) \right\}$$

and the marginal log-likelihood is given by

$$\ell(\vartheta) \propto -\frac{1}{2} \left\{ \log|V| + (Y - Ab - X\beta)^T V^{-1} (Y - Ab - X\beta) \right\}.$$

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

A6. Obtain approximation
Linear approximation to a spline.

Description

Approximation to a spatial semiparametric model based on a bivariate spline.

Usage

```
## S4 method for signature 'sssFit'
scpApproximate(object, tol)
```

Arguments

`object`	an object of class `sssFit` from command `scp`.
`tol`	numeric. Numeric tolerance to use for some inversion of matrices. Default to `.Machine$double.neg.eps*1.0e-10`.

Details

`scpApproximate` compute an approximation to the spatial semiparametric model obtained from `scp`. This command update the fitted values and fitted spline in the input object of class `sssFit`. Then we can use the command `plot` for plotting the approximated semiparametric model.

Value

This command return an object of class `sssFit`.

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

B1. Testing surface *Testing the surface model*

Description

Test the model for the surface of response. The null hypothesis is assumed as a linear model defined by the coordinates while the alternative hypothesis is assumed a bivariate spline (tensor product or thin-plate spline).

Usage

```
## S4 method for signature 'sssFit'
testSurface(object, tol)
```

Arguments

`object`	an object of class `sssFit` from command `scp`.
`tol`	numeric. Numeric tolerance to use for some inversion of matrices. Default to `.Machine$double.neg.eps*1.0e-10`.

Details

If we have defined a bivariate spline using s2D in the formula of scp then the model is an spatial semiparametric model based on splines (tensor products or thin-plate splines). In this case `testSurface` performs a test for the null hypothesis $H_0 : g = X\beta$ (linear model) against the alternative $H_1 : g = X\beta + Zr$ (spline model). When g is assumed as a thin-plate spline then this test is equivalent to test the null hypothesis H_0 : *the pattern of response in the space is a plane* against the alternative H_1 : *the pattern of response in the space is a bivariate thin-plate spline*. In one dimension this test is equivalent to a test for linearity in the pattern of response.

Value

Returns a table with the degrees of freedom, sum of squares and mean squares from different sources and the F test and its associated p-value.

Author(s)

Mario A. Martinez Araya <mma@mariomartinezaraya.com> [aut, cre]

`B2. Plotting` *Plot observed, fitted values or estimated spline from* `scp` *object.*

Description

Draw an image, perspective or levelplot of the observed, fitted values or estimated bivariate spline from the elements in a `scp` object.

Usage

```
## S4 method for signature 'sssFit,missing'
plot(x, what, type, which, col.args, col.contour,
    level.at, border, theta, phi, shade, ...)
```

Arguments

`x`	sssFit object from scp.
`what`	character. What to plot? One of '`obs`' (for observed responses), '`fit`' (for fitted values, the default) or '`g`' (for the estimated bivariate spline).
`type`	character. Which type of plot? One of '`image`' (the default), '`levelplot`', '`persp`' or '`persp3d`'.

`which`	character. Which color pattern? One of '`colorRampPalette`' (default), '`colorRamp`', '`rainbow`', '`heat.colors`', '`terrain.colors`', '`topo.colors`', or '`cm.colors`'.
`col.args`	named list. List with argument to pass to the color pattern function defined by `which`. See colorRampPalette, colorRamp, rainbow, heat.colors, '`terrain.colors`', '`topo.colors`', and '`cm.colors`'.
`col.contour`	character. Only for `type="image"`. Color for the contours.
`level.at`	character or numeric vector. Only for `type="levelplot"`. Where to draw the levels at. If character, it is the name of the function to compute where to put the levels.
`border`	character. Color of the border.
`theta, phi, shade`	numeric. See persp or persp3d.
`...`	other arguments for levelplot, image, persp or persp3d.

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

`B3. Summary` *Summary of the estimated model from* `scp` *object.*

Description

Report estimated coefficients, standard errors and t-tests for parametric effects of the semiparametric model from a `scp` object.

Usage

```
## S4 method for signature 'sssFit'
summary(object,alpha=0.05)
```

Arguments

`object`	sssFit object from scp.
`alpha`	numeric $[0, 1]$. The level for confidence intervals.

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

`B4. Information criterion`
Information criterion of the estimated model from `scp` *object.*

Description

Return the information criterion of the estimated model from a `scp` object.

Usage

```
## S4 method for signature 'sssFit'
AIC(object, k, only.criterion)
## S4 method for signature 'sssFit'
BIC(object, only.criterion)
## S4 method for signature 'sssFit'
AICm(object, k, only.criterion)
## S4 method for signature 'sssFit'
AICc(object, k, only.criterion)
## S4 method for signature 'sssFit'
BICc(object, only.criterion)
## S4 method for signature 'sssFit'
BICj(object, k, tol, only.criterion)
## S4 method for signature 'sssFit'
GIC(object, k, only.criterion)
## S4 method for signature 'sssFit'
GIChq(object, k, only.criterion)
## S4 method for signature 'sssFit'
GICpn(object, only.criterion)
## S4 method for signature 'sssFit'
GICb(object, only.criterion)
```

Arguments

`object`	sssFit object from scp.
`k`	numeric. Factor multiplying the number of parameters in each criterion. Default to `k=2`.
`tol`	numeric. Value for the tolerance in some computation of inverse matrices. By default is set to `.Machine$double.neg.eps`.
`only.criterion`	logical. If `TRUE` (the default) returns only the value of the criterion.

Details

The information criterion for a mixed model is defined as

$$IC = -2\ell + penalty$$

where ℓ is the log-likelihood $\ell(\vartheta)$ or conditional log-likelihood $\ell(\vartheta|r)$ (see scp). The penalty is expressed as $k \times a_0 \times \omega_{\mu_*,V}$ where $\omega_{\mu_*,V} = \omega_{\mu_*} + \omega_V$ is the (effective) number of parameters in the mean and variance and k and a_0 are factors that depend on the criterion used. Thus the information criterion can be written as

$$IC = -2\ell + k \times a_0 \times \omega_{\mu_*,V}.$$

Note that μ_* depends on the criterion being used so it can be $\mu_* = \mu_m$ or $\mu_* = \mu$. See scp.

Value

If `only.criterion=TRUE` returns the value of the criterion. If `only.criterion=FALSE` returns a list with the following elements:

logLik numeric. The log-likelihood or conditional log-likelihood (given r) of the model depending of the criterion used.

criterion numeric. The value of the information criterion.

ka0 numeric. Factors ka_0 multiplying the number of parameters. Depends on the criterion selected.

numpar numeric. The (effective) number of parameters. Depends on the criterion selected.

penalty numeric. The value of the penalty.

Author(s)

Mario Martinez Araya, `<mma@mariomartinezaraya.com>`

References

• Müller, Samuel; Scealy, J. L. and Welsh, A. H. (2013) Model Selection in Linear Mixed Models. Statist. Sci. **28**, no. 2, 135–167. doi:10.1214/12-STS410. `http://projecteuclid.org/euclid.ss/1369147909`.

`B5. Variogram` *Compute and plot the semi-variogram from* `scp` *object.*

Description

Compute and plot the semi-variogram of the semiparametric model from a `scp` object.

Usage

```
## S4 method for signature 'sssFit'
Variogram(object, distance, plot, ...)
```

Arguments

`object`	sssFit object from scp.
`distance`	numeric vector. The distances at which to compute the semi-variogram. By default is set to `NULL`.
`plot`	logical. `plot=TRUE` (the default) produce the semivariogram plot. `plot=FALSE` returns the values of the semivariogram at `distance`. See *value*.
`...`	other graphical parameters to pass.

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

`sss-class` *Class* `"sss"`

Description

Create a dataset for spatial smoothing splines. Those dataset can be used later by functions s2D for tensor product (natural) cubic splines or p-splines, and scp for estimating spatial smoothing splines models.

Usage

```
sss(...)
```

Arguments

`...`	Slots elements to be included into the `sss` dataset. Allowed slots names are `data`, `coords`, `grid`, `knots`, `W`, `contract` (to be discarded in the future), and `regular`.

Objects from the Class

Objects can be created by calls of the form `sss(...)`.

Slots

data a data-frame containing the variables measured at the locations given by `coords`.

coords a matrix containing the two columns of observed coordinates for the data.

grid a grid matrix containing the two columns of coordinates.

knots a named list with the design points (knots) in every coordinate. Equivalent to a `grid.list` object.

W a spatial incidence matrix. If `contract=TRUE` it is W_{ij}, otherwise W_{ji}.

contract logical. The same value as the argument `contract`.

regular logical. If the coordinates are observed at regular points it is `TRUE`, `FALSE` otherwise (missing coordinates in any direction).

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

---

sssFit-class *Class* "sssFit"

Description

Output object from scp that can be used with methods for post-processing.

Usage

```
sssFit(...)
```

Arguments

...	Slots elements to be included into the sssFit object. Allowed slots names are data, zV, XL, XC, XF, XS, fit, and call.

Objects from the Class

Objects of this class are created by calls to the command scp (see scp). It is also possible to define an empty object of this class by calls of the form sssFit(name) however for further use this is subject to validity of the object.

Slots

data an object of sss class containing the input data.

zV numeric vector. Response variables measured at the locations given by data@coords.

XL a named list with elements and covariates from linear command.

XC a named list with elements and covariates from cp command.

XF not implemented.

XS a named list with elements from s2D command.

fit a named list with different estimated parameters and summaries from the estimated model.

call call to the fitted model.

Author(s)

Mario Martinez Araya, <mma@mariomartinezaraya.com>

Index