Package ‘ncf’

April 13, 2016

Version 1.1-7
Date 2016-04-12
Title Spatial Nonparametric Covariance Functions
Author Ottar. N. Bjornstad <onb1@psu.edu>
Maintainer Ottar N. Bjornstad <onb1@psu.edu>
Depends R (>= 2.8.0)
Description R functions for analyzing spatial (cross-)covariance: the
  nonparametric (cross-)covariance, the spline correlogram, the
  nonparametric phase coherence function, and related.
License GPL (>= 2)
URL http://ento.psu.edu/directory/onb1
NeedsCompilation no
Repository CRAN
Date/Publication 2016-04-13 09:28:18

R topics documented:

cc.offset ................................................................. 2
cor2 ........................................................................ 3
correlog ................................................................. 3
correlog.nc ............................................................. 5
ff.filter ................................................................. 7
gcdist ....................................................................... 8
lbg ................................................................. 9
lisa ................................................................. 9
lisa.nc ................................................................. 11
mantel.correlog ..................................................... 12
mantel.test ............................................................. 14
mSynch ................................................................. 15
partial.mantel.test .................................................. 17
plot.cc.offset .......................................................... 18
plot.correlog .......................................................... 19
function to calculate the distance at which the cross-correlation peaks for Sncf objects

Description

Alternative ‘Summary’ method for class "Sncf2D".

Usage

cc.offset(obj, xmax)

Arguments

obj an object of class "Sncf2D", usually, as a result of a call to Sncf2D or spline.correlog.2D.

xmax The maximum distance to consider (default is no upper limit).

Value

An matrix of class "cc.offset" is returned with columns:

- angle the cardinal angle (in degrees).
- distance the distances (in the positive direction) to the mode of the (cross-) correlation function (with 95% confidence bounds).
- correlation the correlation at the mode (with CI) for each of the cardinal angles.

See Also

Sncf2D summary.Sncf2D plot.cc.offset
cor2

Utility function

Description
Called by various functions to calculate Pearson or angular correlation matrixes.

Usage

cor2(x, y = NULL, circ=FALSE)

Arguments

x a matrix.
y an optional second matrix.
circ if TRUE, the observations are assumed to be angular (in radians), and circular correlation is used. If FALSE, Pearson product moment correlations is returned.

Details
An auxiliary function to ease the maintenance.

Value
A correlation matrix is returned.

References

correlog

Uni- and multivariate spatial correlograms

Description
correlog is the function to estimate spatial (cross-)correlograms. Either univariate or multivariate (time series) for each site can be used.

Usage
correlog(x, y, z, w = NULL, increment, resamp = 1000,
latlon = FALSE, na.rm = FALSE, quiet = FALSE)
Arguments

- **x**: vector of length n representing the x coordinates (or longitude; see latlon).
- **y**: vector of length n representing the y coordinates (or latitude).
- **z**: vector of length n or matrix of dimension n x p representing p observation at each location.
- **w**: an optional second variable with identical dimension to z (to estimate cross-correlograms).
- **increment**: increment for the uniformly distributed distance classes.
- **resamp**: the number of permutations under the null to assess level of significance.
- **latlon**: if TRUE, coordinates are latitude and longitude.
- **na.rm**: if TRUE, NA's will be dealt with through pairwise deletion of missing values.
- **quiet**: if TRUE the counter is suppressed during execution.

Details

The spatial (cross-)correlogram and Mantel (cross-)correlogram estimates the spatial dependence at discrete distance classes.

The regionwide similarity forms the reference line (the zero-line); the x-intercept is thus the distance at which object are no more similar than that expected by chance alone across the region.

If the data are univariate, the spatial dependence is measured by Moran's $I$, if it is multivariate it is measured by the centered Mantel statistic. (Use `correlog.nc` if the non-centered multivariate correlogram is desired).

Missing values are allowed – values are assumed missing at random.

Value

An object of class "correlog" is returned, consisting of the following components:

- **correlation**: the value for the moran (or Mantel) similarity.
- **mean.of.class**: the actual average of the distances within each distance class.
- **nlok**: the number of pairs within each distance class.
- **x.intercept**: the interpolate x.intercept of Epperson (1993).
- **p**: the permutation two-sided p-value for each distance-class.
- **corr0**: if a cross-correlogram is calculated, corr0 gives the empirical cross-correlation at distance zero.

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>
correlog.nc

References


See Also

plot.correlog spline.correlog correlog nc

Examples

# first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]

# z data from an exponential random field
z <- cbind(
  rmv.spa(x=x, y=y, p=2, method="exp"),
  rmv.spa(x=x, y=y, p=2, method="exp")
)

# w data from a gaussian random field
w <- cbind(
  rmv.spa(x=x, y=y, p=2, method="gaus"),
  rmv.spa(x=x, y=y, p=2, method="gaus")
)

# Spatial correlogram
fit1 <- correlog(x=x, y=y, z=z[,1], increment=2, resamp=0)
## Not run: plot(fit1)

# Mantel correlogram
fit2 <- correlog(x=x, y=y, z=z, increment=2, resamp=0)
## Not run: plot(fit2)

# Mantel cross-correlogram
fit3 <- correlog(x=x, y=y, z=z, w=w, increment=2, resamp=0)
## Not run: plot(fit3)
Description

correlog.nc is the function to estimate the non-centred (cross-)correlogram. The noncentred correlogram provides estimates of the spatial correlation for discrete distance classes. The function requires multiple observations at each location (use correlog otherwise).

Usage

correlog.nc(x, y, z, w = NULL, increment, resamp = 1000, na.rm = FALSE, latlon=FALSE, quiet = FALSE)

Arguments

x vector of length n representing the x coordinates (or longitude; see latlon).
y vector of length n representing the y coordinates (or latitude).
z a matrix of dimension n x p representing p (>1) observation at each location.
w an optional second variable with identical dimension to z (to estimate cross-correlograms).
increment increment for the uniformly distributed distance classes.
resamp the number of permutations under the null to assess level of significance.
latlon if TRUE, coordinates are latitude and longitude.
na.rm if TRUE, NA's will be dealt with through pairwise deletion of missing values.
quiet if TRUE the counter is suppressed during execution.

Details

The non-centred correlogram estimates spatial dependence at discrete distance classes. The method corresponds to the modified correlogram of Koenig & Knops(1998), but augmented to potentially estimate the cross-correlogram). The function requires multiple observations at each location. Missing values is allowed in the multivariate case (pairwise deletion will be used).

Missing values are allowed – values are assumed missing at random.

Value

An object of class "correlog" is returned, consisting of the following components:
correlation the value for the moran (or Mantel) similarity.
mean.of.class the actual average of the distances within each distance class.
nlok the number of pairs within each distance class.
x.intercept the interpolate x.intercept of Epperson (1993).
p the permutation p-value for each distance-class.
corr0 if a cross-correlogram is calculated, corr0 gives the empirical within-patch cross-correlation.
ff.filter

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

References


See Also

plot.correlog correlog

Examples

#first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]

#z data from an exponential random field
z <- cbind(rmvnspa(x=x, y=y, p=2, method="exp"),
           rmvnspa(x=x, y=y, p=2, method="exp")
)

#w data from a gaussian random field
w <- cbind(rmvnspa(x=x, y=y, p=2, method="gaus"),
           rmvnspa(x=x, y=y, p=2, method="gaus")
)

#noncentered (Mantel) correlogram
fit1 <- correlog.nc(x=x, y=y, z=z, increment=2, resamp=500)
## Not run: plot.correlog(fit1)

ff.filter

Fourier filter for c'fns

Description

Fourier filter to ensure positive semi-definite correlation functions. Called by various functions.

Usage

ff.filter(x)
Arguments

x a vector.

Value

A vector is returned whose Fourier-transform has no non-negative coefficients.

See Also

Sncf

---

gcdist  

*Greater circle distance*

Description

Greater circle distance function to calculate spatial distance from lat-long data. Called by various functions.

Usage

gcdist(x1, y1, x2, y2)

Arguments

x1 longitude 1.

x2 longitude 2.

y1 latitude 1.

y2 latitude 2.

Details

Function for great circle distance – due to T. Keitt.

Value

The distance in km is returned

Author(s)

Tim H. Keitt
**lm**  

---

**Spatio-temporal data panel of Larch Budmoth defoliation**

**Description**

This is the data in Bjornstad et al. (2002).

**Usage**

```r
data(lbm)
```

**Format**

A data-frame with 135 rows and 40 columns. The first two are the x- and y-coordinates (in m), the following 38 represents the defoliation in years 1961 through 1998.

**References**


---

**lisa**  

---

**Local indicator of spatial association**

**Description**

lisa is a function to estimate the local indicators of spatial association. The function assumes univariate data at each location. For multivariate data use `lisa.nc`

**Usage**

```r
lisa(x, y, z, neigh, resamp, latlon=FALSE, quiet = FALSE)
```

**Arguments**

- `x` vector of length n representing the x coordinates (or latitude; see latlon).
- `y` vector of length n representing the y coordinates (or longitude).
- `z` vector of n representing the observation at each location.
- `neigh` neighborhood size.
- `resamp` number of resamples under the NULL to generate p-values
- `latlon` if TRUE, coordinates are latitude and longitude.
- `quiet` if TRUE the counter is suppressed during execution.
Details

This is the function to estimate the local indicators of spatial association modified from Anselin (1995). The statistic is the average autocorrelation within a neighborhood.

Value

An object of class "lisa" is returned, consisting of the following components:

correlation  the autocorrelation within the neighborhood (neigh) of each observation measured using Moran’s I.

p            the permutation two-sided p-value for each observation.

mean         the mean of the observations inside each neighborhood.

n            the number of observations within each neighborhood.

dmean        the actual mean distance within each neighborhood.

z            the original observations

coord        a list with the x and y coordinates.

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

References


See Also

plot.lisa

Examples

# first generate some sample data
  x <- expand.grid(1:20, 1:5)[,1]
  y <- expand.grid(1:20, 1:5)[,2]

# z data from an exponential random field
  z <- rmvnspa(x=x, y=y, p=2, method="gaus")

# lisa analysis
  fit1 <- lisa(x=x, y=y, z=z, neigh=3, resamp=500)
  # Not run: plot.lisa(fit1, neigh.mean=FALSE)
**Description**

`lisa.nc` is a function to estimate the (noncentred) multivariate local indicators of spatial association. The function requires multiple observations at each location. For single observations at each location use `lisa`.

**Usage**

```r
lisa.nc(x, y, z, neigh, na.rm = FALSE, resamp=1000, 
latlon = FALSE, quiet = FALSE)
```

**Arguments**

- `x` vector of length `n` representing the x coordinates (or latitude; see `latlon`).
- `y` vector of length `n` representing the y coordinates (or longitude).
- `z` a matrix of dimension `n x p` representing `p (>1)` observation at each location.
- `neigh` neighborhood size.
- `resamp` number of resamples under the NULL to generate p-values.
- `latlon` Nif TRUE, coordinates are latitude and longitude.
- `na.rm` if TRUE, NA’s will be dealt with through pairwise deletion of missing values.
- `quiet` if TRUE the counter is suppressed during execution.

**Details**

This is the function to estimate the (non-centered) local indicators of spatial association modified form Anselin (1995). ‘correlation’ is the average correlation within a neighborhood. The function requires multiple observations at each location.

Missing values are allowed – values are assumed missing at random, and pairwise complete observations will be used.

**Value**

An object of class "lisa.nc" is returned, consisting of the following components:

- `correlation` the mean correlation within the neighborhood (neigh).
- `p` the permutation two-sided p-value for each distance-class.
- `n` the number of pairs within each neighborhood.
- `dmean` the actual mean of distance within each neighborhood.
- `coord` a list with the x and y coordinates.
mantel.correlog

Description

mantel.correlog is the function to calculate a Mantel (cross-)correlogram. The function requires two (or three) matrices.

Usage

mantel.correlog(dmat, zmat, wmat = NULL, increment, resamp = 1000, quiet = FALSE)

Arguments

- **dmat**: a matrix representing distance.
- **zmat**: a matrix representing similarity.
- **wmat**: an optional third matrix of similarities to calculate a Mantel cross-correlograms.
- **increment**: increment for the uniformly distributed distance classes.
- **resamp**: the number of permutations under the null to assess level of significance.
- **quiet**: if TRUE the counter is supressed during execution.

Examples

```r
# first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]

# z data from an exponential random field
z <- cbind(
  rmvnormspa(x=x, y=y, p=2, method="exp"),
  rmvnormspa(x=x, y=y, p=2, method="exp")
)

# lisa.nc analysis
fit1 <- lisa.nc(x=x, y=y, z=z, neigh=3)
## Not run: plot.lisa.nc(fit1)
```
Details

The function calculates Mantel (cross-)correlograms at discrete distance classes from two (or three) matrices. The first is the matrix of distances and the second is a matrix of similarities. The optional third matrix is an additional similarity matrix to be used to calculate a Mantel cross-correlogram. Missing values are allowed – values are assumed missing at random.

Value

An object of class "correlog" is returned, consisting of the following components:

- **correlation**: the value for the moran (or Mantel) similarity.
- **mean.of.class**: the actual average of the distances within each distance class.
- **nlok**: the number of pairs within each distance class.
- **x.intercept**: the interpolate x.intercept of Epperson (1993).
- **p**: the permutation two-sided p-value for each distance-class.
- **corr0**: if a cross-correlogram is calculated, corr0 gives the empirical cross-correlation at distance zero.

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

See Also

- `plot.correlog`

Examples

```r
# first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]

# z data from an exponential random field
z <- cbind(
  rmvnormspa(x=x, y=y, p=2, method="exp"),
  rmvnormspa(x=x, y=y, p=2, method="exp")
)

# w data from a gaussian random field
w <- cbind(
  rmvnormspa(x=x, y=y, p=2, method="gaus"),
  rmvnormspa(x=x, y=y, p=2, method="gaus")
)

# Make distance and similarity matrices
zmat=cor(t(z))
wmat=cor(t(w))
dmat <- sqrt(outer(x,x, "-")^2+outer(y,y, "+")^2)
```
# Mantel correlogram
fit1 <- mantel.correlog(dmat=dm, zmat=zmat, increment=2, quiet=TRUE, resamp=0)
## Not run: plot(fit1)

# Mantel cross-correlogram
fit2 <- mantel.correlog(dmat=dm, zmat=zmat, wmat=wmat, increment=2, quiet=TRUE, resamp=0)
## Not run: plot(fit2)

---

mantel.test  ManTEL Test

Description

A simple function to do a permutation-based Mantel test. The data can either be two distance/similarity matrices or (x, y, z) data.

Usage

mantel.test(M1 = NULL, M2 = NULL, x = NULL, y = NULL, z = NULL,
            resamp = 1000, latlon = FALSE, quiet = FALSE)

Arguments

M1, M2  similarity/distance matrix 1, and similarity/distance matrix 2
x  vector of length n representing the x coordinates (or longitude; see latlon).
y  vector of length n representing the y coordinates (or latitude).
z  matrix of dimension n x p representing p observation at each location.
resamp  the number of resamples for the null distribution.
latlon  if TRUE, coordinates are latitude and longitude.
quiet  if TRUE the counter is suppressed during execution.

Details

Typical usages are

mantel.test(M1, M2, x = NULL, y = NULL, z = NULL,
            resamp = 1000, latlon = FALSE, quiet = FALSE)

mantel.test(x, y, z, M1=NULL, M2=NULL, resamp = 1000,
            latlon = FALSE, quiet = FALSE)

Missing values are are treated through pairwise deletion.
mSynch

Value

An object of class "Mantel" is returned, consisting of a list with two components:

- correlation is the value for the Mantel correlation.
- p the randomization-based two-sided p-value.

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

Examples

```r
# first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]
# z data from an exponential random field
z <- cbind(
  rmvn.spa(x=x, y=y, p=2, method="exp"),
  rmvn.spa(x=x, y=y, p=2, method="exp")
)

# the Mantel test
mantel.test(x=x, y=y, z=z[,1], resamp=500)
```

mSynch

the mean (cross-)correlation (with bootstrapp CI) for a panel of spatiotemporal data

Description

mSynch is the function to estimate the mean (cross-)correlation in a spatiotemporal dataset as discussed in Bjornstad et al. (1999). The function requires multiple observations at each location.

Usage

```r
mSynch(x, y = NULL, resamp = 1000, na.rm = FALSE,
circ = FALSE, quiet = FALSE)
```

Arguments

- **x** matrix of dimension n x p representing p observation at each location (i.e. each row is a time series).
- **y** optional matrix of dimension m x p representing p observation at each location (i.e. each row is a time series). If provided, the mean cross-correlation between the two panels is computed.
- **resamp** the number of resamples for the bootstrap or the null distribution.
na.rm if TRUE, NA's will be dealt with through pairwise deletion of missing values for each pair of time series – it will dump if any one pair has less than two (temporally) overlapping observations.
circ if TRUE, the observations are assumed to be angular (in radians), and circular correlation is used.
quiet if TRUE the counter is supressed during execution.

Details
Missing values are allowed – values are assumed missing at random.
The circ argument computes a circular version of the Pearson's product moment correlation (see cor2).

Value
An object of class "mSynch" is returned, consisting of a list with two components:

real the regional average correlation.
boot A vector of bootstrap resamples.

Author(s)
Ottar N. Bjornstad <onb1@psu.edu>

References

See Also
print.mSynch

Examples
#first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]
#z data from an exponential random field
z <- cbind(
  rmvn.spa(x=x, y=y, p=2, method="exp"),
  rmvn.spa(x=x, y=y, p=2, method="exp")
)

#mean correlation analysis
fit1 <- mSynch(x=z, resamp=500)
print(fit1)
partial.mantel.test

Partial Mantel test

Description

A simple function to calculate permutation-based partial mantel tests for three matrices, the partial mantel test is calculated to test for relationships between M1 and M2 (M3) controlling for M3 (M2). syntax and logic follows Legendre and Legendre (1998) pp 557-558

Usage

partial.mantel.test(M1, M2, M3, resamp = 1000, method = "pearson", quiet = FALSE)

Arguments

M1, M2, M3 similarity/distance matrices 1, 2 and 3.
resamp the number of resamples for the null distribution.
method The method to be used for calculating the correlations.
quiet if TRUE the counter is suppressed during execution.

Details

Missing values are treated through pairwise deletion. The method must be one of pearson (default), spearman or kendall

Value

An object of class "partial.Mantel" is returned, consisting of a list with two components:

MantelR the vector of observed Mantel and partial Mantel correlations.
p is the vector of two-sided p-value under randomization (of M1).

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

References


See Also

mantel.test
Examples

```r
# first generate some sample data and dissimilarity matrices
x <- rnorm(10)
y <- rnorm(10)
z <- rnorm(10)
M1 <- sqrt(outer(x, x, "-")^2)
M2 <- sqrt(outer(y, y, "-")^2)
M3 <- sqrt(outer(z, z, "-")^2)

partial.mantel.test(M1=M1, M2=M2, M3=M3, resamp=500)
```

---

**plot.cc.offset**  
Plots the *cc.offset* summary of the anisotropic spatial correlation-functions

**Description**

'plot' method for class "cc.offset".

**Usage**

```r
## S3 method for class 'cc.offset'
plot(x, xmax = 0, ...)
```

**Arguments**

- `x`  
  an object of class "cc.offset", usually, as a result of applying `cc.offset` to an object of class `Sncf2D`.

- `xmax`  
  The maximal distance to be plotted on the x-axis. If set to zero the maximum distance in the data will be used.

- `...`  
  other arguments

**Value**

A radial 'symbol' plot results. The radius represents the distance to peak correlation (the mode) of the correlation function (in the positive direction). The size of the symbol represents the magnitude of the correlation at the mode for the given cardinal direction.

**See Also**

`cc.offset` `Sncf2D` `plot.Sncf2D`
plot.correlog

Plots spatial correlograms

Description

'plot' method for class "correlog".

Usage

```r
## S3 method for class 'correlog'
plot(x, ...)  
```

Arguments

- `x`: an object of class "correlog", usually, as a result of a call to `correlog` or `correlog.nc`.
- `...`: other arguments

Value

A spatial or Mantel (cross-correlogram) is plotted.

If a permutation test was performed, values significant at a nominal (two-sided) 5%-level will be represented by filled circles and non-significant values by open circles.

See Also

correlog correlog.nc

plot.lisa

Plots local indicators of spatial association

Description

'plot' method for class "lisa".

Usage

```r
## S3 method for class 'lisa'
plot(x, neigh.mean = FALSE, add=FALSE, inches=0.2, ...)  
```

Arguments

- `x`: an object of class "lisa", usually, as a result of a call to `lisa`.
- `neigh.mean`: if TRUE size of symbols represents average observation in each neighborhood, if FALSE size of symbols represents the original observation
- `add`: if TRUE, a lisa-plot will be added to a pre-existing plot
- `inches`: scales the size of the symbols
- `...`: other arguments
Value

A bubble-plot of observations against spatial coordinates is produced. Below mean values are signified by squares. Above mean values are signified by circles.

If a permutation test was performed, observations for which the associated LISA statistic is positive and significant at a nominal (two-sided) 5%-level will be represented by filled symbols and non-significant values by open symbols. Thus spatial hot-spots are represented by red filled circles and cold-spots by black filled squares.

See Also

lisa lisa.nc

plot.lisa.nc

Plots multivariate local indicators of spatial association

Description

'plot' method for class "lisa.nc".

Usage

## S3 method for class 'lisa.nc'
plot(x, ctr = FALSE, add=FALSE, inches=0.2, ...)

Arguments

x
an object of class "lisa.nc", usually, as a result of a call to lisa.nc.

ctr
if TRUE correlations will be centered before plotting (zero-sized symbols represents the average correlation), if FALSE the observed correlations are used (zero-sized symbols represents zero within neighborhood correlation)

add
if TRUE, a lisa-plot will be added to a pre-existing plot

inches
scales the size of the symbols

... other arguments

Value

A bubble-plot of the multivariate LISA against spatial coordinates is produced. Negative (or below mean) within-neighborhood correlations are signified by squares. Positive (or above mean) values are signified by circles.

If a permutation test was performed, values significant at a nominal (two-sided) 5%-level will be represented by filled symbols and non-significant values by open symbols. Thus areas of significant positive (or above-average) correlation have filled red circles and areas of significant negative (or below-average) correlation have filled black squares.

See Also

lisa lisa.nc
plot.Sncf

Plots nonparametric spatial correlation-functions

Description

'plot' method for class "Sncf".

Usage

## S3 method for class 'Sncf'
plot(x, xmax = 0, text = TRUE, add = FALSE, ...)

Arguments

x
an object of class "Sncf", usually, as a result of a call to Sncf (or Sncf.srf).
xmax
The maximal distance to be plotted on the x-axis. If set to zero the maximum
distance in the data will be used.
text
if TRUE a summary of the regional average correlation makes the plot title; the
y-label summarizes the y-intercept; the x-label summarizes the x-intercept and
the cbar-intercept (denoted "r:").
add
if TRUE the plot is added on to the previous graph.
...
other arguments

Value

A plot of the correlation function against distance is produced. 95% pointwise confidence (or null)
envelopes are superimposed.

See Also

Sncf Sncf.srf summary.Sncf

plot.Sncf.cov

Plots nonparametric spatial covariance-functions

Description

'plot' method for class "Sncf.cov".

Usage

## S3 method for class 'Sncf.cov'
plot(x, xmax = 0, text = TRUE, ...)
plot.Sncf2D

Arguments

x  an object of class "Sncf.cov", usually, as a result of a call to Sncf.srf (with corr = FALSE).

xmax The maximal distance to be plotted on the x-axis. If set to zero the maximum distance in the data will be used.

text if TRUE a summary of the regional average correlation makes the plot title; the y-label summarizes the y-intercept; the x-label summarizes the x-intercept and the cbar-intercept (denoted "r:").

... other arguments

See Also

Sncf.srf plot.Sncf

plot.Sncf2D  Plots anisotropic spatial correlation-functions

Description

'plot' method for class "Sncf2D".

Usage

## S3 method for class 'Sncf2D'
plot(x, xmax = 0, text = TRUE, detail = FALSE, ...)

Arguments

x  an object of class "Sncf2D", usually, as a result of a call to Sncf2D.

xmax The maximal distance to be plotted on the x-axis. If set to zero the maximum distance in the data will be used.

detail if TRUE a separate plot is made for each direction (including confidence envelopes; see plot.Sncf for details. If FALSE all correlation functions are superimposed on the same plot.

text if TRUE (and detail = TRUE) a summary of the regional average correlation makes the plot title; the y-label summarizes the y-intercept; the x-label summarizes the x-intercept and the cbar-intercept (denoted "r:").

... other arguments

Value

A plot or panel-plot results. These represents the xy-plot of distance against spatial (cross-)correlation for each cardinal direction.

See Also

Sncf2D plot.Sncf
**plot.spline.correlog**  
*Plots spline correlograms*

**Description**

'plot' method for class "spline.correlog".

**Usage**

```r
## S3 method for class 'spline.correlog'
plot(x, xmax = 0, text = TRUE, ...)
```

**Arguments**

- **x**: an object of class "spline.correlog", usually, as a result of a call to `spline.correlog`.
- **xmax**: The maximal distance to be plotted on the x-axis. If set to zero the maximum distance in the data will be used.
- **text**: if TRUE the y-label summarizes the y-intercept and the x-label summarizes the x-intercept.
- **...**: other arguments

**Value**

A plot of the spline correlogram function against distance is produced. 95% pointwise confidence (or null) envelopes are superimposed.

**See Also**

- `spline.correlog`
- `summary.spline.correlog`

---

**print.mSynch**  
*Print function for mSynch objects*

**Description**

'Print' method for class "mSynch".

**Usage**

```r
## S3 method for class 'mSynch'
print(x, verbose = FALSE, ...)
```
Arguments

x       an object of class "mSynch", usually, as a result of a call to mSynch).
verbose if TRUE a raw listing of the object is produced. if FALSE a summary list is produced
...     other arguments

Value
If verbose is FALSE a list summarizing the regional correlation is produced:

mean    the regional mean correlation.
Squantile the quantile distribution from the resampling for the regional correlation.

See Also
msynch

---

rmvn.spa Simulate spatial data

Description
Function to generate spatially autocorrelated random normal variates using the eigendecomposition method. Spatial covariance can follow either an exponential or Gaussian model.

Usage
rmvn.spa(x, y, p, method = "exp", nugget = 1)

Arguments

x       vector of length n representing the x coordinates (or latitude; see latlon).
y       vector of length n representing the y coordinates (or longitude).
p       the range of the spatial models.
method  correlation function "exp" (exponential) or "gaus" (gaussian). exponential is the default.
nugget  correlation at the origin (defaults to one)

Details
A target covariance matrix A between the n units is generated by calculating the distances between the locations and thereafter evaluating the covariance function in each pairwise distance. A vector, Z, of spatially correlated normal data with the target covariance is subsequently generated using the eigendecomposition method (Ripley, 1987).
Value

a vector of spatially correlated random normal variates with zero mean and unit variance is returned

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

References


See Also

mSynch

---

**Sncf**

Nonparametric (cross-)correlation function for spatio-temporal data

Description

Sncf is the function to estimate the nonparametric (cross-)correlation function using a smoothing spline as an equivalent kernel. The function requires multiple observations at each location (use **spline.correlog** otherwise).

Usage

```r
Sncf(x, y, z, w = NULL, df = NULL, type = "boot", resamp = 1000,
      npoints = 300, save = FALSE, filter = FALSE, fw = 0, max.it = 25,
      xmax = FALSE, na.rm = FALSE, latlon = FALSE, circ = FALSE, quiet = FALSE)
```

Arguments

- **x**: vector of length n representing the x coordinates (or longitude; see latlon).
- **y**: vector of length n representing the y coordinates (or latitude).
- **z**: matrix of dimension n x p representing p observation at each location.
- **w**: an optional second matrix of dimension n x p for species 2 (to estimate the spatial cross-correlation function).
- **df**: degrees of freedom for the spline. Default is \( \sqrt{n} \).
- **type**: takes the value "boot" (default) to generate a bootstrap distribution or "perm" to generate a null distribution for the estimator.
- **resamp**: the number of resamples for the bootstrap or the null distribution.
- **npoints**: the number of points at which to save the value for the spline function (and confidence envelope / null distribution).
- **save**: if TRUE the whole matrix of output from the resampling is saved (an resamp x npoints dimensional matrix).
filter  if TRUE the Fourier filter method of Hall and coworkers is applied to ensure positive semidefiniteness of the estimator. (more work may be needed on this.)
fw  if filter is TRUE, it may be useful to truncate the function at some distance w sets the truncation distance. when set to zero no truncation is done.
max.it  the maximum iteration for the Newton method used to estimate the intercepts.
xmax  if FALSE the max observed in the data is used. Otherwise all distances greater than xmax is omitted.
na.rm  if TRUE, NA's will be dealt with through pairwise deletion of missing values for each pair of time series – it will dump if any one pair has less than two (temporally) overlapping observations.
latlon  if TRUE, coordinates are latitude and longitude.
circ  if TRUE, the observations are assumed to be angular (in radians), and circular correlation is used.
quiet  if TRUE the counter is supressed during execution.

Details

Missing values are allowed – values are assumed missing at random.
The circ argument computes a circular version of the Pearson’s product moment correlation (see cor2). This option is to calculate the ‘nonparametric phase coherence function’ (Grenfell et al. 2001)

Value

An object of class "Sncf" is returned, consisting of the following components:
real  The list of estimates from the data.
$cbar  the regional average correlation.
$x.intercept  the lowest value at which the function is = 0. If correlation is initially negative, the distance is given as negative.
$e.intercept  the lowest value at which the function 1/e.
$y.intercept  the extrapolated value at x=0 (nugget).
$cbar.intercept  distance at which regional average correlation is reach.
$predicted$x  the x-axes for the fitted covariance function.
$predicted$y  the values for the covariance function.
boot  A list with the analogous output from the bootstrap or null distribution.
$summary  gives the full vector of output for the x.intercept, y.intercept, e.intercept, cbar.intercept, cbar and a quantile summary for the resampling distribution.
$boot  if save=TRUE the full raw matrices from the resampling is saved.
max.distance  the maximum spatial distance considered.

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>
References


See Also

summary.Sncf plot.Sncf Sncf2D Sncf.srf

Examples

#first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]
#z data from an exponential random field
z <- cbind(
   rmvn.spa(x=x, y=y, p=2, method="exp"),
   rmvn.spa(x=x, y=y, p=2, method="exp")
)
#w data from a gaussian random field
w <- cbind(
   rmvn.spa(x=x, y=y, p=2, method="gaus"),
   rmvn.spa(x=x, y=y, p=2, method="gaus")
)
#multivariate nonparametric covariance function
fit1 <- Sncf(x=x, y=y, z=z, resamp = 0)
## Not run: plot.Sncf(fit1)
summary.Sncf(fit1)

#multivariate nonparametric cross-covariance function
fit2 <- Sncf(x=x, y=y, z=z, w=w, resamp = 0)
## Not run: plot.Sncf(fit2)
summary.Sncf(fit2)
Nonparametric (Cross-)Covariance Function from stationary random fields

Description

Sncf.srf is the function to estimate the nonparametric for spatio-temporal data from fully stationary random fields (i.e. marginal expectation and variance identical for all locations; use Sncf otherwise).

Usage

Sncf.srf(x, y, z, w = NULL, avg = NULL, avg2 = NULL, corr = TRUE, df = NULL, type = "boot", resamp = 0, npoints = 300, save = FALSE, filter = FALSE, fw = 0, max.it = 25, xmax = FALSE, jitter = FALSE, quiet = FALSE)

Arguments

x vector of length n representing the x coordinates (or longitude; see latlon).
y vector of length n representing the y coordinates (or latitude).
z matrix of dimension n x p representing p observation at each location.
w an optional second matrix of dimension n x p for variable 2 (to estimate the spatial cross-correlation function).
avg supplies the marginal expectation of the Markov random field; if TRUE the sample mean (across the markovian field) is used.
avg2 Optionally supplies the marginal expectation of the Markov random field for optional variable 2; if TRUE the sample mean is used.
corr if TRUE, the covariance function is standardized by the marginal variance (across the markovian field) to return a correlation function (alternatively the covariance function is returned).
df degrees of freedom for the spline. Default is sqrt(n).
type takes the value "boot" (default) to generate a bootstrap distribution or "perm" to generate a null distribution for the estimator
resamp the number of resamples for the bootstrap or the null distribution.
npoints the number of points at which to save the value for the spline function (and confidence envelope / null distribution).
save if TRUE the whole matrix of output from the resampling is saved (an resamp x npoints dimensional matrix).
filter if TRUE the Fourier filter method of Hall and coworkers is applied to ensure positive semidefiniteness of the estimator. (more work may be needed on this.)
fw if filter is TRUE, it may be useful to truncate the function at some distance w sets the truncation distance. when set to zero no truncation is done.
max.it  the maximum iteration for the Newton method used to estimate the intercepts.
xmax  if FALSE the max observed in the data is used. Otherwise all distances greater than xmax is omitted.
jitter  if TRUE, jitters the distance matrix, to avoid problems associated with fitting the function to data on regular grids
quiet  if TRUE the counter is supressed during execution.

Details

if corr = F, an object of class "Sncf.cov" is returned. Otherwise the class is "Sncf".

Sncf.srf is a function to estimate the nonparametric (cross-)covariance function (as discussed in Bjornstad and Bascompte 2001) for data from a fully stationary random fields. I have found it useful to estimate the (cross-)covariance functions in synthetic data.

Value

an object of class "Sncf" (or "Sncf.cov") is returned see Sncf for details.

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

References


See Also

Sncf summary.Sncf plot.Sncf plot.Sncf cov

Examples

#first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]

#z data from an exponential random field
z <- cbind(
  rmvn.spa(x=x, y=y, p=2, method="exp"),
  rmvn.spa(x=x, y=y, p=2, method="exp")
)

#w data from a gaussian random field
w <- cbind(
  rmvn.spa(x=x, y=y, p=2, method="gaus"),
  rmvn.spa(x=x, y=y, p=2, method="gaus")
)

#multivariate nonparametric covariance function
Sncf2D

Description

Sncf2D is the function to estimate the anisotropic nonparametric correlation function in 8 (or arbitrary) directions (North - Southeast). Correlation functions are calculated for each different bearing. The function requires multiple observations at each location. (use `spline.correlog.2D` otherwise).

Usage

```r
Sncf2D(x, y, z, w = NULL, df = NULL, type = "boot", resamp = 1000,
       npoints = 300, save = FALSE, max.it = 25, xmax = FALSE, na.rm = FALSE,
       jitter = FALSE, quiet = FALSE,
       angle = c(0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5))
```

Arguments

- **x** vector of length n representing the x coordinates.
- **y** vector of length n representing the y coordinates.
- **z** matrix of dimension n x p representing p observation at each location.
- **w** an optional second matrix of dimension n x p for variable 2 (to estimate spatial or lagged cross-correlation functions).
- **df** degrees of freedom for the spline. Default is sqrt(n).
- **type** takes the value "boot" (default) to generate a bootstrap distribution or "perm" to generate a null distribution for the estimator
- **resamp** the number of resamples for the bootstrap or the null distribution.
- **npoints** the number of points at which to save the value for the spline function (and confidence envelope / null distribution).
- **save** if TRUE the whole matrix of output from the resampling is saved (an resamp x npoints dimensional matrix).
- **max.it** the maximum iteration for the Newton method used to estimate the intercepts.
\textbf{xmax} \quad \text{if FALSE the max observed in the data is used. Otherwise all distances greater than xmax is omitted.}

\textbf{na.rm} \quad \text{if TRUE, NA's will be dealt with through pairwise deletion of missing values for each pair of time series – it will dump if any one pair has less than two (temporally) overlapping observations.}

\textbf{jitter} \quad \text{if TRUE, jitters the distance matrix, to avoid problems associated with fitting the function to data on regular grids}

\textbf{quiet} \quad \text{if TRUE the counter is suppressed during execution.}

\textbf{angle} \quad \text{specifies number of cardinal directions and angles for which to calculate correlation functions. Default are 8 directions between 0 and 180.}

\textbf{Details}

Correlation functions are calculated on projected distances onto the different bearings so ALL data are used for each direction. The (obsolete?) oldncf2D used the alternative of slicing up the data like pieces of a pie.

Latitude-longitude coordinates can NOT be used.

Missing values are allowed - values are assumed missing at random.

I have implemented an optional argument: \textit{jitter} if TRUE this jitters the distance matrix, to avoid some problems I've had with spline-smoothing data from regular grid-data.

\textbf{Value}

An object of class "Sncf2D" is returned, consisting of a list of estimates for each cardinal direction:

- \texttt{real} \quad \text{The list of estimates from the data.}
- \texttt{cbar} \quad \text{the regional average correlation.}
- \texttt{x.intercept} \quad \text{the lowest value at which the function is = 0. If correlation is initially negative, the distance is given as negative.}
- \texttt{e.intercept} \quad \text{the lowest value at which the function 1/e.}
- \texttt{y.intercept} \quad \text{the extrapolated value at x=0 (nugget).}
- \texttt{cbar.intercept} \quad \text{distance at which regional average correlation is reached.}
- \texttt{predictedx} \quad \text{the x-axes for the fitted covariance function.}
- \texttt{predictedy} \quad \text{the values for the covariance function.}
- \texttt{boot} \quad \text{A list with the analogous output from the bootstrap or null distribution.}
- \texttt{summary} \quad \text{gives the full vector of output for the x.intercept, y.intercept, e.intercept, cbar.intercept, and the cbar and a quantile summary for the resampling distribution.}
- \texttt{boot} \quad \text{if save=TRUE the full raw matrices from the resampling is saved.}
- \texttt{angle} \quad \text{A vector with the cardinal directions.}
- \texttt{max.distance} \quad \text{the maximum spatial distance.}
Note

The function to estimate the anisotropic nonparametric (cross-)correlation function in arbitrary directions. In particular it was developed to calculate the lagged cross-correlation function (Bjornstad et al. 2002).

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

References


See Also

`summary.Nsncf2D, plot.Nsncf2D, cc.offset, Sncf, spline.correlog.2D`

Examples

```r
# first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]
# z data from an exponential random field
z <- cbind(
  rmvn.spa(x=x, y=y, p=2, method="exp"),
  rmvn.spa(x=x, y=y, p=2, method="exp")
)
# anisotropic nonparametric covariance function at 30 and 60 degrees
fit1 <- Sncf2D(x=x, y=y, z=z, resamp = 0, angle=c(30, 60))
# Not run: plot.Sncf2D(fit1)
summary.Sncf2D(fit1)

# What distance is the peak in correlation
cc.offset(fit1)
```

---

**spatial.plot**

Simple wrapper around symbols to visualize spatial data

**Description**

Spatial.plot is a quick function to visualize spatial data using bubble plots

**Usage**

```r
spatial.plot(x, y, z, ctr=TRUE, add=FALSE, inches=.2, ...)
```
spatial.plot

Arguments

x vector of length n representing the x coordinates.

y vector of length n representing the y coordinates.

z vector of length n representing the observation at each location.

ctr if TRUE observations will be centered before plotting (zero-sized symbols repre-
sents average observations), if FALSE the original observations are used

add if TRUE, a lisa-plot will be added to a pre-existing plot

inches scales the size of the symbols

... other arguments

details

This is a simple function to visualize spatial data. Positive (or above average) observations are
shown by red circles, Negative (or below average) observations are shown as black squares. For
hot/coldspot analysis using Local indicators of spatial association use lisa.

Value

A bubble-plot of the spatial data is produced.

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

See Also

lisa

Examples

#first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]

#z data from an exponential random field
z <- rmvnspa(x=x, y=y, p=2, method="gaus")

#plot data
## Not run: spatial.plot(x=x, y=y, z=z, ctr=FALSE)
spline.correlog

Uni- and multivariate spline correlograms

Description

spline.correlog is the function to estimate the spline (cross-)correlogram from spatial data. Either univariate or multivariate (time series) for each site can be used.

Usage

spline.correlog(x, y, z, w = NULL, df = NULL, type = "boot", resamp = 1000, npoints = 300, save = FALSE, filter = FALSE, fw = 0, max.it = 25, xmax = FALSE, latlon = FALSE, na.rm = FALSE, quiet = FALSE)

Arguments

x vector of length n representing the x coordinates (or longitude; see latlon).
y vector of length n representing the y coordinates (or latitude).
z vector of length n or matrix of dimension n x p representing p observation at each location.
w an optional second variable with identical dimension to z (to estimate cross-correlograms).
df degrees of freedom for the spline. Default is sqrt(n).
type takes the value "boot" (default) to generate a bootstrap distribution or "perm" to generate a null distribution for the estimator.
resamp the number of resamples for the bootstrap or the null distribution.
npoints the number of points at which to save the value for the spline function (and confidence envelope / null distribution).
save if TRUE the whole matrix of output from the resampling is saved (an resamp x npoints dimensional matrix).
filter if TRUE the Fourier filter method of Hall and coworkers is applied to ensure positive semidefiniteness of the estimator.
fw if filter is TRUE, it may be useful to truncate the function at some distance w sets the truncation distance. when set to zero no truncation is done.
max.it the maximum iteration for the Newton method used to estimate the intercepts.
xmax if FALSE the max observed in the data is used. Otherwise all distances greater than xmax is omitted.
latlon if TRUE, coordinates are latitude and longitude.
na.rm if TRUE, NA's will be dealt with through pairwise deletion of missing values.
quiet if TRUE the counter is suppressed during execution.
Details

If observations are univariate the spline (cross-)correlogram represents the generalization of the spatial (cross-)correlogram; if observations are multivariate the spline (cross-)correlogram represents the generalization of the Mantel (cross-)correlogram.

The spline (cross-)correlogram differs from the spatial correlogram (and Mantel correlogram) in that it estimated spatial dependence as a continuous functions of distance (rather than binning into distance classes). The spline correlogram differs from the nonparametric (cross-)correlation function in that the zero-correlation reference line in the former corresponds to the regionwide correlation reference line in the latter. The x-intercept in the spline correlogram is the distance at which object are no more similar than that expected by-chance-alone across the region.

Missing values are allowed – values are assumed missing at random.

Value

An object of class "spline.correlog" is returned, consisting of the following components:

- real: The list of estimates from the data.
- $x.intercept: the lowest value at which the function is = 0. If correlation is initially negative, the distance is given as negative.
- $e.intercept: the lowest value at which the function 1/e.
- $y.intercept: the extrapolated value at x=0 (nugget).
- $predicted$x: the x-axes for the fitted covariance function.
- $predicted$y: the values for the covariance function.
- boot: A list with the analogous output from the bootstrap or null distribution.
- $summary: gives the full vector of output for the x.intercept, y.intercept, e.intercept, and a quantile summary for the resampling distribution.
- $boot: if save=TRUE the full raw matrices from the resampling is saved.
- max.distance: the maximum spatial distance considered.

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

References


See Also

summary.spline.correlog plot.spline.correlog Sncf spline.correlog 2D correlog
Examples

# first generate some sample data
x <- expand.grid(1:20, 1:5)[,1]
y <- expand.grid(1:20, 1:5)[,2]

# z data from an exponential random field
z <- cbind(
    rmvnNspa(x=x, y=y, p=2, method="exp"),
    rmvnNspa(x=x, y=y, p=2, method="exp")
)

# w data from a gaussian random field
w <- cbind(
    rmvnNspa(x=x, y=y, p=2, method="gaus"),
    rmvnNspa(x=x, y=y, p=2, method="gaus")
)

# univariate spline correlogram
fit1 <- spline.correlog(x=x, y=y, z=z[,1], resamp = 100)
## Not run: plot.spline.correlog(fit1)
summary.spline.correlog(fit1)

# multivariate spline correlogram
fit2 <- spline.correlog(x=x, y=y, z=z, resamp = 100)
## Not run: plot.spline.correlog(fit2)
summary.spline.correlog(fit2)

# multivariate spline cross-correlogram
fit3 <- spline.correlog(x=x, y=y, z=z, w=w, resamp = 100)
## Not run: plot.spline.correlog(fit3)
summary.spline.correlog(fit3)

spline.correlog.2D Anisotropic nonparametric (cross-)correlation function for univariate spatial data

Description

spline.correlog.2D is the function to estimate the anisotropic nonparametric correlation function in 8 (or arbitrary) directions (North - Southeast) for univariate data. Correlation functions are calculated for each different bearing. The function assumes univariate observations at each location. (use Sncf2D otherwise).

Usage

spline.correlog.2D(x, y, z, w = NULL, df = NULL, type = "boot", resamp = 1000, npoints = 300, save = FALSE, max.it = 25, xmax = FALSE, na.rm = FALSE, jitter = FALSE, quiet = FALSE, angle = c(0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5))
Arguments

- **x**: vector of length n representing the x coordinates.
- **y**: vector of length n representing the y coordinates.
- **z**: vector of length n representing the observation at each location.
- **w**: an optional second vector of length n for variable 2 (to estimate spatial or lagged cross-correlation functions).
- **df**: degrees of freedom for the spline. Default is sqrt(n).
- **type**: takes the value "boot" (default) to generate a bootstrap distribution or "perm" to generate a null distribution for the estimator.
- **resamp**: the number of resamples for the bootstrap or the null distribution.
- **npoints**: the number of points at which to save the value for the spline function (and confidence envelope / null distribution).
- **save**: if TRUE the whole matrix of output from the resampling is saved (an resamp x npoints dimensional matrix).
- **max.it**: the maximum iteration for the Newton method used to estimate the intercepts.
- **xmax**: if FALSE the max observed in the data is used. Otherwise all distances greater than xmax is omitted.
- **na.rm**: if TRUE, NA's will be dealt with through pairwise deletion of missing values for each pair of time series – it will dump if any one pair has less than two (temporally) overlapping observations.
- **jitter**: if TRUE, jitters the distance matrix, to avoid problems associated with fitting the function to data on regular grids.
- **quiet**: if TRUE the counter is suppressed during execution.
- **angle**: specifies number of cardinal directions and angles for which to calculate correlation functions. Default are 8 directions between 0 and 180.

Details

see Sncf2D

Value

An object of class "Sncf2D" is returned. See Sncf2D for details.

Note

The function to estimate the UNIvariate anisotropic nonparametric (cross-)correlation function in arbitrary directions. In particular it was developed to calculate the univariate lagged cross-correlation function used in (Humston et al. 2005). Note that this 2D spline correlogram does the anisotropic analysis NOT by doing the angle-with-tolerance-wedge-style of Oden and Sokal (1986) but by projecting the the spatial coordinates of all locations on a sequence of cardinal angles (a la Sncf2D). Hence, all data points are used every time, it is only their relative distances that are changed. For example 0,0 and 0, 10 are distance zero in the zero-degree direction but at distance 10 in the 90-degree direction.
References


See Also

Sncf2D

summary.Sncf

Summarizing nonparametric spatial correlation-functions

Description

‘Summary’ method for class "Sncf".

Usage

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

Arguments

object an object of class "Sncf", usually, as a result of a call to Sncf (or Sncf.srf).
...
other arguments

Value

A list summarizing the nonparametric (cross-)covariance function is returned.

Regional.synch the regional mean (cross-)correlation.
Squantile estimates
the quantile distribution from the resampling for the regional correlation.
a vector of benchmark statistics:
$x$ is the lowest value at which the function is $= 0$. If correlation is initially negative, the distance calculated appears as a negative measure.
$e$ is the lowest value at which the function is $<= 1/e$.
$y$ is the extrapolated value at $x=0$.
$cbar$ is the shortest distance at which function is $= \text{regional mean correlation}$.
quantiles A matrix summarizing the quantiles in the bootstrap (or null) distributions of the benchmark statistics.

See Also

Sncf plot.Sncf
**summary.Sncf2D**

*Summary* method for class "Sncf2D".

**Usage**

```r
## S3 method for class 'Sncf2D'
summary(object, ...)
```

**Arguments**

- `object`: an object of class "Sncf2D", usually, as a result of a call to Sncf2D.
- `...`: other arguments

**Value**

A list summarizing the Nonparametric covariance function in each cardinal direction results, each with the entries as in `summary.Sncf`.

**See Also**

`Sncf2D cc.offset summary.Sncf`

---

**summary.spline.correlog**

*Summary* method for class "spline.correlog".

**Usage**

```r
## S3 method for class 'spline.correlog'
summary(object, ...)
```

**Arguments**

- `object`: an object of class "spline.correlog", usually, as a result of a call to `spline.correlog`.
- `...`: other arguments

**See Also**

`Sncf2D cc.offset summary.Sncf spline.correlog`
Value

A list summarizing spline correlograms is returned.

estimates a vector of benchmark statistics:
$x$ is the lowest value at which the function is $= 0$. If correlation is initially negative, the distance calculated appears as a negative measure.
$e$ is the lowest value at which the function is $\leq 1/e$.
$y$ is the extrapolated value at $x=0$.
quantiles A matrix summarizing the quantiles in the bootstrap (or null) distributions of the benchmark statistics.

See Also

spline.correlog plot.spline.correlog
Index

*Topic **datasets**
  lbm, 9

*Topic **misc**
  cor2, 3
  ff.filter, 7
  gcdist, 8

*Topic **regression**
  cc.offset, 2
  plot.cc.offset, 18
  plot.Sncf, 21
  plot.Sncf.cov, 21
  plot.Sncf2D, 22
  plot.spline.correlog, 23
  print.mSynch, 23
  rmvn.spa, 24
  Sncf, 25
  Sncf.srf, 28
  Sncf2D, 30
  spline.correlog.2D, 36
  summary.Sncf, 38
  summary.Sncf2D, 39
  summary.spline.correlog, 39

*Topic **smooth**
  cc.offset, 2
  plot.cc.offset, 18
  plot.Sncf, 21
  plot.Sncf.cov, 21
  plot.Sncf2D, 22
  plot.spline.correlog, 23
  print.mSynch, 23
  rmvn.spa, 24
  Sncf, 25
  Sncf.srf, 28
  Sncf2D, 30
  spline.correlog, 34
  spline.correlog.2D, 36
  summary.Sncf, 38
  summary.Sncf2D, 39
  summary.spline.correlog, 39

*Topic **spatial**
  correlog, 3
  correlog.nc, 5
  lisa, 9
  lisa.nc, 11
  mantel.correlog, 12
  mantel.test, 14
  mSynch, 15
  partial.mantel.test, 17
  plot.correlog, 19
  plot.lisa, 19
  plot.lisa.nc, 20
  Sncf, 25
  spatial.plot, 32
  spline.correlog, 34
  cc.offset, 2, 18, 32, 39
  cor2, 3, 16, 26
  correlog, 3, 6, 7, 19, 35
  correlog.nc, 4, 5, 5, 19
  ff.filter, 7
  gcdist, 8
  lbm, 9
  lisa, 9, 12, 19, 20, 33
  lisa.nc, 9, 11, 20
  mantel.correlog, 12
  mantel.test, 14, 17
  mSynch, 15, 24, 25
  partial.mantel.test, 17
  plot.cc.offset, 2, 18
  plot.correlog, 5, 7, 13, 19
  plot.lisa, 10, 19
  plot.lisa.nc, 12, 20
  plot.Sncf, 21, 22, 27, 29, 38
  plot.Sncf.cov, 21, 29
  plot.Sncf2D, 18, 22, 32
plot.spline.correlog, 23, 35, 40
print.mSynch, 16, 23
rmvn.spa, 24

Sncf, 8, 21, 25, 28, 29, 32, 35, 38
Sncf.srf, 21, 22, 27, 28, 38
Sncf2D, 2, 18, 22, 27, 30, 36–39
spatial.plot, 32
spline.correlog, 5, 23, 25, 34, 39, 40
spline.correlog.2D, 30, 32, 35, 36
summary.Sncf, 21, 27, 29, 38, 39
summary.Sncf2D, 2, 32, 39
summary.spline.correlog, 23, 35, 39