Package 'ncf'

January 29, 2020

R topics documented:

cc.offset	*Function to calculate the distance at which the cross-correlation peaks for Sncf objects*

Description

Alternative 'summary' method for class "Sncf2D".

Usage

```
cc.offset(object, xmax = NULL)
```

Arguments

object	an object of class "Sncf2D", usually, as a result of a call to Sncf2D or spline.correlog2D.
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

---

circ.cor2	*Circular correlation*

Description

A vectorized function to calculate a correlation matrix for panels of data.

Usage

```
circ.cor2(x, y = NULL)
```

Arguments

x	a matrix.
y	an optional second matrix.

Details

Missing values are not allowed.

Value

A correlation matrix is returned.

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 8.2, World Scientific Press, Singapore.

---

cor2   *Utility function*

Description

Called by various functions to calculate Pearson or angular correlation matrices.

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 auxilliary function to ease the maintenance.

Value

A correlation matrix is returned.

References

Jammalamadaka, S. Rao and SenGupta, A. (2001). Topics in Circular Statistics, Section 8.2, World Scientific Press, Singapore.

---

correlog   *Uni- and multivariate spatial correlograms*

Description

`correlog` is the function to estimate spatial (cross-)correlograms. Either univariate or multivariate (time seres) 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 region-wide 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 *centred* 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>

References

Bjornstad, O.N., Ims, R.A. & Lambin, X. (1999) Spatial population dynamics: Analysing patterns and processes of population synchrony. Trends in Ecology and Evolution, 11, 427-431. https://doi.org/10.1016/S0169-5347(99)01677-8

Bjornstad, O.N. & Falck, W. (2001) Nonparametric spatial covariance functions: estimation and testing. Environmental and Ecological Statistics, 8:53-70. https://doi.org/10.1023/A:1009601932481

Epperson, B.K. (1993) Recent advances in correlation studies of spatial patterns of genetic variation. Evolutionary Biology, 27, 95-155. https://doi.org/10.1007/978-1-4615-2878-4_4

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(
  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")
  )

# 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)
```

---

correlog.nc   *Non-centered spatial (cross-)correlogram*

Description

correlog.nc is the function to estimate the non-centered (cross-)correlogram. The non-centered 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.
na.rm	If TRUE, NA's will be dealt with through pairwise deletion of missing values.
latlon	If TRUE, coordinates are latitude and longitude.
quiet	If TRUE, the counter is suppressed during execution.

Details

The non-centered 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.

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

References

Bjornstad, O.N., Ims, R.A. & Lambin, X. (1999) Spatial population dynamics: Analysing patterns and processes of population synchrony. Trends in Ecology and Evolution, 11, 427-431. https://doi.org/10.1016/S0169-5347(99)01677-8

Koenig, W.D. & Knops, J.M.H. (1998) Testing for spatial autocorrelation in ecological studies. Ecography, 21, 423-429. https://doi.org/10.1111/j.1600-0587.1998.tb00407.x

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(
  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")
  )

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

---

gather *Utility function*

Description

Called by various functions to calculate various intercepts.

Usage

```
gather(u, v, w, moran, df, xpoints, filter, fw)
```

Arguments

u	a vector.
v	a vector.
w	a vector.
moran	a matrix.
df	a scalar.
xpoints	a vector.
filter	a logical.
fw	a scalar

Details

An auxiliary function to ease maintenance.

Value

A list is returned.

---

gcdist   *Great-circle distance*

Description

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

Usage

```
gcdist(x, y)
```

Arguments

x   vector of longitudes.

y   vector of latitudes.

Value

The distance in km is returned

---

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

Description

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

Usage

```
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

Bjornstad, O.N., Peltonen, M., Liebhold, A.M., and Baltensweiler, W. (2002) Waves of larch budmoth outbreaks in the European Alps. Science, 298, 1020-1023. https://doi.org/10.1126/science.1075182

---

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

```
lisa(x, y, z, neigh, 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	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 form 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 neighborhooddistance within 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

Anselin, L. 1995. Local indicators of spatial association - LISA. Geographical Analysis 27:93-115.
<https://doi.org/10.1111/j.1538-4632.1995.tb00338.x>

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 <- rmvn.spa(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(fit1, neigh.mean=FALSE)
```

---

lisa.nc   *Non-centered indicators of spatial association*

Description

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

Usage

```
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.
na.rm	If TRUE, NA's will be dealt with through pairwise deletion of missing values.
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 (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" 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.

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

References

Anselin, L. 1995. Local indicators of spatial association - LISA. Geographical Analysis 27:93-115. https://doi.org/10.1111/j.1538-4632.1995.tb00338.x

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 <- cbind(
  rmvn.spa(x = x, y = y, p = 2, method = "exp"),
  rmvn.spa(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(fit1)
```

---

mantel.correlog	*Mantel (cross-)correlograms*

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 suppressed during execution.

Details

The function calculates Mantel (cross-)correlograms at discrete distance classes from two (or three) matrixes. 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

```
# 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")
  )

# 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 = dmat, zmat = zmat, increment = 2, quiet = TRUE,
                        resamp = 0)
## Not run: plot(fit1)

# Mantel cross-correlogram
fit2 <- mantel.correlog(dmat = dmat, 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	similarity/distance matrix 1
M2	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 treated through pairwise deletion.

Value

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

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

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

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")
  )

# 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

```
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 suppressed 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

Bjornstad, O.N., Ims, R.A. & Lambin, X. (1999) Spatial population dynamics: Analysing patterns and processes of population synchrony. Trends in Ecology and Evolution, 11, 427-431. https://doi.org/10.1016/S0169-5347(99)01677-8

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	similarity/distance matrix 1
M2	similarity/distance matrix 2
M3	similarity/distance matrix 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   the vector of two-sided p-value under randomization (of M1).

Author(s)

Ottar N. Bjornstad <onb1@psu.edu>

References

Legendre, P., and L. Legendre. 1998. Numerical Ecology, 2nd edition. Elsevier, Amsterdam

See Also

mantel.test

Examples

```
# 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

```
## S3 method for class 'cc.offset'
plot(x, dmax = NULL, inches = NULL, ...)
```

Arguments

x	an object of class "cc.offset", usually, as a result of applying cc.offset to an object of class Sncf2D.
dmax	the maximal distance for radial plot. If NULL, the maximum distance in the data will be used.
inches	the size of the symbols.If NULL, default is 0.1.
...	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

```
## 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

```
## 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. Above mean values are signified by red circles. Below mean values are signified by black squares.

If a permutation test was performed, observations for which the associated LISA statistic is 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.Sncf   *Plots nonparametric spatial correlation-functions*

Description

'plot' method for class "Sncf".

Usage

```
## S3 method for class 'Sncf'
plot(x, ylim = c(-1, 1), add = FALSE, ...)
```

Arguments

x	an object of class "Sncf", usually, as a result of a call to `Sncf` (or `Sncf.srf`).
ylim	limits for the y-axis (default: -1, 1).
add	If TRUE the plot is added on to the previous graph.
...	other arguments

Value

A plot of the nonparametric spatial covariance function (with CI's if boostrapps are available)

See Also

[Sncf](), [Sncf.srf]()

---

plot.Sncf.cov   *Plots nonparametric spatial covariance-functions*

Description

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

Usage

```
## S3 method for class 'Sncf.cov'
plot(x, ...)
```

Arguments

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

Value

A plot of the nonparametric spatial covariance function (with CI's if boostrapps are available)

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, ylim = c(-1, 1), 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.
ylim	limits for the y-axis (default: -1, 1).
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.
...	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 a spline correlogram*

Description

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

Usage

```
## S3 method for class 'spline.correlog'
plot(x, ylim = c(-1, 1), ...)
```

Arguments

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

ylim   limits for the y-axis (default: -1, 1).

...   other arguments

Value

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

See Also

[spline.correlog](), [summary.spline.correlog]()

---

print.mSynch   *Print function for mSynch objects*

Description

'print' method for class "mSynch".

Usage

```
## 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

---

print.Sncf   *Print function for Sncf objects*

Description

'print' method for class "Sncf".

Usage

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

Arguments

x   an object of class "Sncf", usually, as a result of a call to Sncf or related).

...   other arguments

Value

The function-call is printed to screen.

See Also

Sncf

---

print.Sncf2D   *Print function for Sncf2D objects*

Description

'print' method for class "Sncf2D".

Usage

```
## S3 method for class 'Sncf2D'
print(x, ...)
```

Arguments

x   an object of class "Sncf2D", usually, as a result of a call to Sncf2D or spline.correlog2D).

...   other arguments

Value

The function-call is printed to screen.

See Also

Sncf2D

---

print.spline.correlog *Print function for spline.correlog objects*

Description

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

Usage

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

Arguments

x   an object of class "spline.correlog", usually, as a result of a call to spline.correlog or related).

...   other arguments

Value

The function-call is printed to screen.

See Also

spline.correlog

---

rmvn.spa   *Simulate spatial data*

Description

Function to generate spatially autocorrelated random normal variates using the eigendecomposition method. Spatial covariance can follow either and 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

Ripley, B.D. (1987). Stochastic Simulation. Wiley.

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

```
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 semi-definiteness 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 suppressed 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.
$predcited$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

Hall, P. & Patil, P. (1994) Properties of nonparametric estimators of autocovariance for stationary random fields. Probability Theory and Related Fields, 99:399-424. https://doi.org/10.1007/BF01199899

Hall, P., Fisher, N.I. & Hoffmann, B. (1994) On the nonparametric estimation of covariance functions. Annals of Statistics, 22:2115-2134. https://doi.org/10.1214/aos/1176325774

Bjornstad, O.N. & Falck, W. (2001) Nonparametric spatial covariance functions: estimation and testing. Environmental and Ecological Statistics, 8:53-70. https://doi.org/10.1023/A:1009601932481

Bjornstad, O.N., Ims, R.A. & Lambin, X. (1999) Spatial population dynamics: Analysing patterns and processes of population synchrony. Trends in Ecology and Evolution, 11:427-431. https://doi.org/10.1016/S0169-5347(99)01677-8

Bjornstad, O. N., and J. Bascompte. (2001) Synchrony and second order spatial correlation in host-parasitoid systems. Journal of Animal Ecology 70:924-933. https://doi.org/10.1046/j.0021-8790.2001.00560.x

Grenfell, B.T., Bjornstad, O.N., & Kappey, J. (2001) Travelling waves and spatial hierarchies in measles epidemics. Nature 414:716-723. https://doi.org/10.1038/414716a

See Also

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(fit1)

# multivariate nonparametric cross-covariance function
fit2 <- Sncf(x = x, y = y, z = z, w = w, resamp = 0)
## Not run: plot(fit2)
summary(fit2)
```

Sncf.srf   *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 suppressed 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

Bjornstad, O. N., and J. Bascompte. (2001) Synchrony and second order spatial correlation in host-parasitoid systems. Journal of Animal Ecology 70:924-933. https://doi.org/10.1046/j.0021-8790.2001.00560.x

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
fit1 <- Sncf.srf(x = x, y = y, z = z, avg = NULL, corr = TRUE, resamp = 0)
## Not run: plot(fit1)
summary(fit1)

# multivariate nonparametric cross-covariance function (with known
# marginal expectation of zero for both z and w
fit2 <- Sncf.srf(x = x, y = y, z = z, w = w, avg = 0, avg2 = 0, corr = FALSE,
                 resamp = 0)
## Not run: plot(fit2)
summary(fit2)
```

Sncf2D *Anisotropic nonparametric (cross-)correlation function for spatio-temporal data*

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.correlog2D](#) otherwise).

Usage

```
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.
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

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: `jitter` if TRUE this jitters the distance matrix, to avoid some problems I've had with spline-smoothing data from regular grid-data.

Value

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

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.
$predcited$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, and the cbar and a quantile summary for the resampling distribution.
$boot	If save=TRUE, the full raw matrices from the resampling is saved.
angle	a vector with the cardinal directions.
max.distance	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

Bjornstad, O. N., M. Peltonen, A. M. Liebhold, and W. Baltensweiler. 2002. Waves of larch budmoth outbreaks in the European Alps. Science 298:1020-1023. https://doi.org/10.1126/science.1075182

See Also

summary.Sncf2D, plot.Sncf2D, cc.offset , Sncf, spline.correlog2D

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")
  )
# anisotorpic 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(fit1)
summary(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

```
spatial.plot(x, y, z, ctr = TRUE, add = FALSE, inches = 0.2, ...)
```

Arguments

x	vector of length n representing the x coordinates.
y	vector of length n representing the y coordinates.
z	vector of n representing the observation at each location.
ctr	If TRUE, observations will be centered before plotting (zero-sized symbols represents 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>

References

Ripley, B.D. (1987). Stochastic Simulation. Wiley.

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 <- rmvn.spa(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 seres) 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 region-wide 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.
$predcited$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

Bjornstad, O.N. & Falck, W. (2001) Nonparametric spatial covariance functions: estimation and testing. Environmental and Ecological Statistics, 8:53-70. https://doi.org/10.1023/A:1009601932481

See Also

summary.spline.correlog, plot.spline.correlog, Sncf, spline.correlog2D, 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(
  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")
  )

# univariate spline correlogram
```

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

# multivariate spline correlogram
fit2 <- spline.correlog(x = x, y = y, z = z, resamp = 100)
## Not run: plot.spline.correlog(fit2)
summary(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(fit3)
```

---

spline.correlog2D	*Anisotropic nonparametric (cross-)correlation function for univariate spatial data*

Description

spline.correlog2D 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.correlog2D(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

Oden, N.L. and Sokal, R.R. 1986. Directional autocorrelation: an extension of spatial correlograms to two dimensions. Systematic Zoology 35: 608-617. https://doi.org/10.2307/2413120

Humston, R., Mortensen, D. and Bjornstad, O.N. 2005. Anthropogenic forcing on the spatial dynamics of an agricultural weed: the case of the common sunflower. Journal of Applied Ecology 42: 863-872. https://doi.org/10.1111/j.1365-2664.2005.01066.x

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   the quantile distribution from the resampling for the regional correlation.

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 <= 1/e.

$y   is the extrapolated value at x=0.

$cbar   is the shortest distance at which function is = 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   *Summarizing anisotropic spatial correlation-functions*

Description

'summary' method for class "Sncf2D".

Usage

```
## 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 entires as in `summary.Sncf`.

See Also

`Sncf2D`, `cc.offset`, `summary.Sncf`

---

summary.spline.correlog
                              Summarizing spline correlograms

---

Description

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

Usage

```
## 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

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 <= 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