Package 'smacpod'

May 14, 2018

Type Package

Title Statistical Methods for the Analysis of Case-Control Point Data

Version 2.0.4

Author Joshua French

Maintainer Joshua French <joshua.french@ucdenver.edu>

Description

Statistical methods for analyzing case-control point data. Methods include the ratio of kernel densities, the difference in K Functions, the spatial scan statistic, and q nearest neighbors of cases.

License GPL (>= 2)

LazyLoad yes

Depends R (>= 3.1.1)

Imports spatstat, plotrix, abind, pbapply, sp

Suggests testthat

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-14 14:48:26 UTC

R topics documented:

`circles.intersect` *Determine whether circles intersect*

Description

`circles.intersect` determines whether circles intersect with each other.

Usage

```
circles.intersect(coords, r)
```

Arguments

`coords`	A matrix of coordinates with the centroid of each circle.
`r`	A vector containing the radii of the circles. The length of `r` must equal the number of rows of `coords`.

Details

The algorithm is based on the premise that two circles intersect if, and only if, the distance between their centroids is between the sum and the difference of their radii. I have squared the respective parts of the inequality in the implemented algorithm.

Value

Returns a matrix of logical values indicating whether the circles intersect.

Author(s)

Joshua French

Examples

```
# first two circles intersect each other,
# the next two circles intersect each other
# (but not the previous ones)
# the last circles doesn't intersect any other circle
co = cbind(c(1, 2, 5, 6, 9), c(1, 2, 5, 6, 9))
r = c(1.25, 1.25, 1.25, 1.25, 1.25)
# draw circles
circles.plot(co, r)
# confirm intersections
```

```
circles.intersect(co, r)

# nested circles (don't intersect)
co = matrix(rep(0, 4), nrow = 2)
r = c(1, 1.5)
circles.plot(co, r)
circles.intersect(co, r)
```

`circles.plot` *Plot circles*

Description

`plot.circles` creates a plot with one or more circles (or adds them to an existing plot).

Usage

```
circles.plot(coords, r, add = FALSE, ..., nv = 100, border = NULL,
  ccol = NA, clty = 1, density = NULL, angle = 45, clwd = 1)
```

Arguments

`coords`	A matrix of coordinates with the centroid of each circle.
`r`	A vector containing the radii of the circles. The length of `r` must equal the number of rows of `coords`.
`add`	A logical value indicating whether the circles should be added to an existing plot. Default is `FALSE`.
`...`	Additional arguments passed to the `plot` function.
`nv`	Number of vertices to draw the circle.
`border`	A vector with the desired border of each circle. The length should either be 1 (in which case the border is repeated for all circles) or should match the number of rows of `coords`.
`ccol`	A vector with the desired color of each circle. The length should either be 1 (in which case the color is repeated for all circles) or should match the number of rows of `coords`.
`clty`	A vector with the desired line type of each circle. The length should either be 1 (in which case the line type is repeated for all circles) or should match the number of rows of `coords`.
`density`	A vector with the density for a patterned fill. The length should either be 1 (in which case the density is repeated for all circles) or should match the number of rows of `coords`. See `polygon`
`angle`	A vector with the angle of a patterned fill. The length should either be 1 (in which case the angle is repeated for all circles) or should match the number of rows of `coords`. See `polygon`
`clwd`	A vector with the desired line width of each circle. The length should either be 1 (in which case the line width is repeated for all circles) or should match the number of rows of `coords`.

Author(s)

Joshua French

See Also

draw.circle, polygon

Examples

```
co = cbind(c(1, 2, 5, 6, 9), c(1, 2, 5, 6, 9))
r = c(1.25, 1.25, 1.25, 1.25, 1.25)
# draw circles
circles.plot(co, r)
circles.plot(co, r,
   ccol = c("blue", "blue", "orange", "orange", "brown"),
   density = c(10, 20, 30, 40, 50),
   angle = c(45, 135, 45, 136, 90))
```

---

grave *Medieval Grave Site Data*

Description

This data set contains 143 observations of medieval grave site data stored as a ppp class object from the spatstat package. The data are marked as being "affected" by a tooth deformity or "unaffected" by a tooth deformity.

Usage

```
data(grave)
```

Format

ppp (planar point process) class object from the spatstat package.

Author(s)

Joshua French

Source

Waller, L.A. and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley.

See Also

ppp

kd *Difference of estimated K functions*

Description

kd determines the difference in estimated K functions for a set of cases and controls.

Usage

```
kd(x, case = 2, r = NULL, rmax = rmax, breaks = NULL,
  correction = c("border", "isotropic", "Ripley", "translate"),
  nlarge = 3000, domain = NULL, var.approx = FALSE, ratio = FALSE)
```

Arguments

x	A ppp object from the spatstat package with marks for the case and control groups.
case	The position of the name of the "case" group in levels(x$marks). The default is 2. x$marks is assumed to be a factor. Automatic conversion is attempted if it is not.
r	Optional. Vector of values for the argument r at which $K(r)$ should be evaluated. Users are advised *not* to specify this argument; there is a sensible default. If necessary, specify rmax.
rmax	Optional. Maximum desired value of the argument r.
breaks	This argument is for internal use only.
correction	Optional. A character vector containing any selection of the options "none", "border", "bord.modif", "isotropic", "Ripley", "translate", "translation", "rigid", "none", "good" or "best". It specifies the edge correction(s) to be applied. Alternatively correction="all" selects all options.
nlarge	Optional. Efficiency threshold. If the number of points exceeds nlarge, then only the border correction will be computed (by default), using a fast algorithm.
domain	Optional. Calculations will be restricted to this subset of the window. See Details of Kest.
var.approx	Logical. If TRUE, the approximate variance of $\hat{K}(r)$ under CSR will also be computed.
ratio	Logical. If TRUE, the numerator and denominator of each edge-corrected estimate will also be saved, for use in analysing replicated point patterns.

Details

This function relies internally on the Kest and eval.fv functions from the spatstat package. The arguments are essentially the same as the Kest function, and the user is referred there for more details about the various arguments.

Value

Returns an `fv` object. See documentation for `spatstat::Kest`.

Author(s)

Joshua French

References

Waller, L.A. and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley.

See Also

`Kest`, `eval.fv`

Examples

```
data(grave)
# kd = kd(grave)
# plot(kd)
```

`kdest` *Difference of estimated K functions*

Description

`kdest` determines the difference in estimated K functions for a set of cases and controls. Tolerance envelopes can also be produced.

Usage

```
kdest(x, case = 2, nsim = 0, level = 0.95, r = NULL, rmax = NULL,
  breaks = NULL, correction = c("border", "isotropic", "Ripley",
  "translate"), nlarge = 3000, domain = NULL, var.approx = FALSE,
  ratio = FALSE)
```

Arguments

`x`	A ppp object from the `spatstat` package with marks for the case and control groups.
`case`	The position of the name of the "case" group in `levels(x$marks)`. The default is 2. `x$marks` is assumed to be a factor. Automatic conversion is attempted if it is not.
`nsim`	An non-negative integer. Default is 0. The difference in estimated K functions will be calculated for `nsim` data sets generated under the random labeling hypothesis. These will be used to construct the tolerance envelopes.

level	Level of tolerance envelopes. Ignored if nsim is 0.
r	Optional. Vector of values for the argument r at which $K(r)$ should be evaluated. Users are advised *not* to specify this argument; there is a sensible default. If necessary, specify rmax.
rmax	Optional. Maximum desired value of the argument r.
breaks	This argument is for internal use only.
correction	Optional. A character vector containing any selection of the options "none", "border", "bord.modif", "isotropic", "Ripley", "translate", "translation", "rigid", "none", "good" or "best". It specifies the edge correction(s) to be applied. Alternatively correction="all" selects all options.
nlarge	Optional. Efficiency threshold. If the number of points exceeds nlarge, then only the border correction will be computed (by default), using a fast algorithm.
domain	Optional. Calculations will be restricted to this subset of the window. See Details of Kest.
var.approx	Logical. If TRUE, the approximate variance of $\hat{K}(r)$ under CSR will also be computed.
ratio	Logical. If TRUE, the numerator and denominator of each edge-corrected estimate will also be saved, for use in analysing replicated point patterns.

Details

This function relies internally on the Kest and eval.fv functions from the spatstat package. The arguments are essentially the same as the Kest function, and the user is referred there for more details about the various arguments.

Value

Returns a kdenv object. See documentation for spatstat::Kest.

Author(s)

Joshua French

References

Waller, L.A. and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley.

See Also

Kest, eval.fv

Examples

```
data(grave)
kd1 = kdest(grave)
plot(kd1, iso ~ r, ylab = "difference", legend = FALSE, main = "")
kd2 = kdest(grave, nsim = 9, level = 0.8)
plot(kd2)
```

kdplus.test *Global test of clustering using difference in K functions*

Description

kdplus.test performs a global test of clustering for comparing cases and controls using the method of Diggle and Chetwynd (1991). It relies on the difference in estimated K functions.

Usage

```
kdplus.test(x)
```

Arguments

x	A kdenv object from the kdest function.

Value

A list providing the observed test statistic (kdplus) and the estimate p-value pvalue.

Author(s)

Joshua French

References

Waller, L.A. and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley. Diggle, Peter J., and Amanda G. Chetwynd. "Second-order analysis of spatial clustering for inhomogeneous populations." Biometrics (1991): 1155-1163.

See Also

kdest

Examples

```
data(grave)
kdsim = kdest(grave, nsim = 9)
kdplus.test(kdsim)
```

logrr *Log ratio of spatial densities*

Description

logrr computes the log ratio of spatial density functions for cases and controls. The numerator in this ratio is related to the "cases" and the denominator to the "controls". If nsim > 0, then pointwise tolerance intervals are used to assess potential clustering of cases and controls relative to each other.

Usage

```
logrr(x, sigma = NULL, sigmacon = NULL, case = 2, nsim = 0,
  level = 0.9, alternative = "two.sided", ..., bwargs = list(),
  weights = NULL, edge = TRUE, varcov = NULL, at = "pixels",
  leaveoneout = TRUE, adjust = 1, diggle = FALSE, kernel = "gaussian",
  scalekernel = is.character(kernel), positive = FALSE, verbose = TRUE)
```

Arguments

x	Point pattern (object of class "ppp").
sigma	Standard deviation of isotropic smoothing kernel for cases. Either a numerical value, or a function that computes an appropriate value of sigma.
sigmacon	Standard deviation of isotropic smoothing kernel for controls. Default is the same as sigma.
case	The position of the name of the "case" group in levels(x$marks). The default is 2. x$marks is assumed to be a factor. Automatic conversion is attempted if it is not.
nsim	The number of simulated data sets from which to construct the tolerance intervals under the random labeling hypothesis. Default is 0 (i.e., no intervals).
level	The tolerance level used for the pointwise tolerance intervals.
alternative	The direction of the significance test to identify potential clusters using a Monte Carlo test based on the pointwise tolerance intervals. Default is "two.sided" (logrr != 0). The values "less" (logrr < 0) and "greater" (logrr > 0) are also valid.
...	Additional arguments passed to pixellate.ppp and as.mask to determine the pixel resolution, or passed to sigma if it is a function.
bwargs	A list of arguments for the bandwidth function supplied to sigma and sigmacon, if applicable.
weights	Optional weights to be attached to the points. A numeric vector, numeric matrix, an expression, or a pixel image.
edge	Logical value indicating whether to apply edge correction.
varcov	Variance-covariance matrix of anisotropic smoothing kernel. Incompatible with sigma.

at	String specifying whether to compute the intensity values at a grid of pixel locations (at="pixels") or only at the points of x (at="points").
leaveoneout	Logical value indicating whether to compute a leave-one-out estimator. Applicable only when at="points".
adjust	Optional. Adjustment factor for the smoothing parameter.
diggle	Logical. If TRUE, use the Jones-Diggle improved edge correction, which is more accurate but slower to compute than the default correction.
kernel	The smoothing kernel. A character string specifying the smoothing kernel (current options are "gaussian", "epanechnikov", "quartic" or "disc"), or a pixel image (object of class "im") containing values of the kernel, or a function(x,y) which yields values of the kernel.
scalekernel	Logical value. If scalekernel=TRUE, then the kernel will be rescaled to the bandwidth determined by sigma and varcov: this is the default behaviour when kernel is a character string. If scalekernel=FALSE, then sigma and varcov will be ignored: this is the default behaviour when kernel is a function or a pixel image.
positive	Logical value indicating whether to force all density values to be positive numbers. Default is FALSE.
verbose	Logical value indicating whether to issue warnings about numerical problems and conditions.

Details

The plot function makes it easy to visualize the log ratio of spatial densities (if nsim = 0) or the regions where the log ratio deviates farther from than what is expected under the random labeling hypothesis (i.e., the locations of potential clustering). The shaded regions indicate the locations of potential clustering.

The two.sided alternative test assesses whether the observed ratio of log densities deviates more than what is expected under the random labeling hypothesis. When the test is significant, this suggests that the cases and controls are clustered, relative to the other. The greater alternative assesses whehter the cases are more clustered than the controls. The less alternative assesses whether the controls are more clustered than the cases. If the estimated density of the case or control group becomes too small, this function may produce warnings due to numerical underflow. Increasing the bandwidth (sigma) may help.

Value

The function produces an object of type logrrenv. Its components are similar to those returned by the density.ppp function from the spatstat package, with the intensity values replaced by the log ratio of spatial densities of f and g. Includes an array simr of dimension c(nx, ny, nsim + 1), where nx and ny are the number of x and y grid points used to estimate the spatial density. simr[,,1] is the log ratio of spatial densities for the observed data, and the remaining nsim elements in the third dimension of the array are the log ratios of spatial densities from a new ppp simulated under the random labeling hypothesis.

Author(s)

Joshua French (and a small chunk by the authors of the density.ppp) function for consistency with the default behavior of that function)

References

Waller, L.A. and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley.

Kelsall, Julia E., and Peter J. Diggle. "Kernel estimation of relative risk." Bernoulli (1995): 3-16.

Kelsall, Julia E., and Peter J. Diggle. "Non-parametric estimation of spatial variation in relative risk." Statistics in Medicine 14.21-22 (1995): 2335-2342.

Examples

```
data(grave)
r = logrr(grave)
plot(r)
r2 = logrr(grave, sigma = spatstat::bw.scott)
plot(r2)
rsim = logrr(grave, nsim = 9)
plot(rsim)
```

---

logrr.test *Global test of clustering using log ratio of spatial densities*

Description

logrr.test performs a global test of clustering for comparing cases and controls using the log ratio of spatial densities based on the method of Kelsall and Diggle (1995).

Usage

```
logrr.test(x)
```

Arguments

x	An logrrenv object from the logrr function.

Value

A list providing the observed test statistic (islogrr) and the estimated p-value (pvalue).

Author(s)

Joshua French

References

Waller, L.A. and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley.

Kelsall, Julia E., and Peter J. Diggle. "Non-parametric estimation of spatial variation in relative risk." Statistics in Medicine 14.21-22 (1995): 2335-2342.

Examples

```
data(grave)
logrrenv = logrr(grave, nsim = 9)
logrr.test(logrrenv)
```

---

nn *Determine nearest neighbors*

Description

nn determines the nearest neighbors for a set of observations based on a distance matrix.

Usage

```
nn(d, k, method = "c", self = FALSE)
```

Arguments

d	A square distance matrix for the coordinates of interest.
k	The number of neighbors to return (if method = "c") or the distance for which observations are considered neighbors (if method = "d").
method	The method of determining the neighbors. The default is "c", specifying that the k nearest neighbors (the number of neighbors) for each observation should be returned. The alternative is "d", meaning that neighbors are determined by their distance from an observation. In that case, two observations are neighbors if their separation distance is less or equal to k.
self	A logical indicating whether an observation is a neighbor with itself. The default is FALSE.

Details

This function determine nearest neighbors in two ways: 1. number of neighbors or 2. distance.

If method = "c", then k specifies the total number of neighbors to return for each observation.

If method = "d", then k specifies the maximum distance for which an observation is considered a neighbor.

The function returns the neighbors for each observation.

Value

Returns a list with the nearest neighbors of each observation. For each element of the list, the indices order neighbors from nearest to farthest.

Author(s)

Joshua French

Examples

```
data(grave)
# make distance matrix
d = as.matrix(dist(cbind(grave$x, grave$y)))
# 3 nearest neighbors
nnc = nn(d, k = 3, method = "c")
# nearest neighbors within k units of each observation
nnd = nn(d, k = 200, method = "d")
```

---

noc *Determine non-overlapping clusters*

Description

Determine non-overlapping clusters from a list of potential clusters.

Usage

```
noc(x)
```

Arguments

x	A list containing the potential clusters.

Details

The function takes a list of potential clusters. Each element of the list contains a potential cluster. The potential clusters are defined by the location indices of the regions comprising the clusters. Starting with the first potential cluster, the function excludes every potential cluster that intersects the first (any potential cluster that shares indices). Moving onto the next non-overlapping cluster, the process is repeated. The function returns the indices (in the list of clusters) of the clusters that do not overlap.

Value

A vector with the list indices of the non-overlapping clusters.

Author(s)

Joshua French

Examples

```
x = list(1:2, 1:3, 4:5, 4:6, 7:8)
noc(x)
```

`plot.kdenv` *Plot a* `kdenv` *object.*

Description

Plots an object from `kdest` of class `kdenv`.

Usage

```
## S3 method for class 'kdenv'
plot(x, ..., shadecol1 = "gray56", shadecol2 = "lightgrey",
  main = "", legend = FALSE)
```

Arguments

`x`	An object of class `kdenv` produced by `kdest`.
`...`	Additional graphical parameters passed to the `plot.fv` function, which is used internally for plotting.
`shadecol1`	Shade color for max/min envelopes. Default is a dark grey.
`shadecol2`	Shade color for tolerance envelopes. Default is "lightgrey".
`main`	A main title for the plot. Default is blank.
`legend`	Logical for whether a legend should automatically be displayed. Default is `FALSE`. See Details for an explanation of the components of the plot.

Details

The solid line indicates the observed difference in the K functions for the cases and controls. The dashed line indicates the average difference in the K functions produced from the data sets simulated under the random labeling hypothesis when executing the `kdest` function.

See Also

`plot.fv`

Examples

```
data(grave)
kd1 = kdest(grave, nsim = 19, level = 0.9)
plot(kd1)
plot(kd1, legend = TRUE)
```

plot.logrrenv *Plots objects produced by the* logrr *function.*

Description

Plots objects of class logrrenv produced by the logrr function.

Usage

```
## S3 method for class 'logrrenv'
plot(x, ..., conlist = list(), main = "")
```

Arguments

x	An object of class logrrenv.
...	Additional graphical parameters passed to the image.im function. See Details.
conlist	Additional argument passed to the contour.im function.
main	A main title for the plot. Default is blank.

Details

An important aspect of this plot is the color argument (col) used for displaying the regions outside the tolerance envelopes. If NULL (the implicit default), then the default color palette used by image.im will be used. Simpler schemes, e.g., c("blue", "white", "orange") can suffice. See the examples.

See Also

plot.im, contour.im

Examples

```
data(grave)
logrrsim = logrr(grave, nsim = 9)
plot(logrrsim)
# no border or ribben (legend).  Simple color scheme.
plot(logrrsim, col = c("blue", "white", "orange"), ribbon = FALSE, box = FALSE)
# alternate color scheme
plot(logrrsim, col = topo.colors(12))
```

plot.spscan *Plots object from* spscan.test.

Description

Plots object of class scan from spscan.test.

Usage

```
## S3 method for class 'spscan'
plot(x, ..., nv = 100, border = NULL, ccol = NULL,
  clty = NULL, clwd = NULL)
```

Arguments

x	An object of class spscan.
...	Additional graphical parameters passed to the plot.ppp function.
nv	The number of verticies to draw the cluster circles. Default is 100.
border	The border color of the circle. Default is NULL, meaning black.
ccol	Fill color of the circles. Default is NULL, indicating empty.
clty	Line type of circles. Default is NULL, indicting lty = 1.
clwd	Line width of circles. Default is NULL, indicating lwd = 2 for the most likely cluster and lwd = 1 for the rest.

Details

If border, ccol, clty, or clwd are specified, then the length of these vectors must match nrow(x$coords).

See Also

plot.ppp, draw.circle

Examples

```
data(grave)
out = spscan.test(grave, case = 2, alpha = 0.1)
plot(out, chars = c(1, 20), main = "most likely cluster")
```

qnn.test *q Nearest Neighbors Test*

Description

qnn.test calculates statistics related to the q nearest neighbors method of comparing case and control point patterns under the random labeling hypothesis.

Usage

```
qnn.test(x, q = 5, nsim = 499, case = 2, longlat = FALSE)
```

Arguments

x	A ppp object from the spatstat package with marks for the case and control groups.
q	A vector of positive integers indicating the values of q for which to do the q nearest neighbors test.
nsim	The number of simulations from which to compute p-value.
case	The position of the name of the "case" group in levels(x$marks). The default is 2.
longlat	A logical value indicating whether Euclidean distance (FALSE) or Great Circle (WGS84 ellipsoid, FALSE) should be used. Default is FALSE, i.e., Euclidean distance.

Value

Returns a list with the following components:

qsum	A dataframe with the number of neighbors (q), test statistic (Tq), and p-value for each test.
consum	A dataframe with the contrasts (contrast), test statistic (Tcon), and p-value (pvaluecon) for the test of contrasts.

Author(s)

Joshua French

References

Waller, L.A., and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley.

Cuzick, J., and Edwards, R. (1990). Spatial clustering for inhomogeneous populations. Journal of the Royal Statistical Society. Series B (Methodological), 73-104.

Alt, K.W., and Vach, W. (1991). The reconstruction of "genetic kinship" in prehistoric burial complexes-problems and statistics. Classification, Data Analysis, and Knowledge Organization, 299-310.

Examples

```
data(grave)
qnn.test(grave, q = c(3, 5, 7, 9, 11, 13, 15))
```

---

spdensity *Kernel smoothed spatial density of point pattern*

Description

spdensity computes a kernel smoothed spatial density function from a point pattern. This function is basically a wrapper for density.ppp. The density.ppp function computes the spatial intensity of a point pattern; the spdensity function scales the intensity to produce a true spatial density.

Usage

```
spdensity(x, sigma = NULL, ..., weights = NULL, edge = TRUE,
  varcov = NULL, at = "pixels", leaveoneout = TRUE, adjust = 1,
  diggle = FALSE, kernel = "gaussian", scalekernel = is.character(kernel),
  positive = FALSE, verbose = TRUE)
```

Arguments

x	Point pattern (object of class "ppp").
sigma	Standard deviation of isotropic smoothing kernel. Either a numerical value, or a function that computes an appropriate value of sigma.
...	Additional arguments passed to pixellate.ppp and as.mask to determine the pixel resolution, or passed to sigma if it is a function.
weights	Optional weights to be attached to the points. A numeric vector, numeric matrix, an expression, or a pixel image.
edge	Logical value indicating whether to apply edge correction.
varcov	Variance-covariance matrix of anisotropic smoothing kernel. Incompatible with sigma.
at	String specifying whether to compute the intensity values at a grid of pixel locations (at="pixels") or only at the points of x (at="points").
leaveoneout	Logical value indicating whether to compute a leave-one-out estimator. Applicable only when at="points".
adjust	Optional. Adjustment factor for the smoothing parameter.
diggle	Logical. If TRUE, use the Jones-Diggle improved edge correction, which is more accurate but slower to compute than the default correction.
kernel	The smoothing kernel. A character string specifying the smoothing kernel (current options are "gaussian", "epanechnikov", "quartic" or "disc"), or a pixel image (object of class "im") containing values of the kernel, or a function(x,y) which yields values of the kernel.

scalekernel	Logical value. If scalekernel=TRUE, then the kernel will be rescaled to the bandwidth determined by sigma and varcov: this is the default behaviour when kernel is a character string. If scalekernel=FALSE, then sigma and varcov will be ignored: this is the default behaviour when kernel is a function or a pixel image.
positive	Logical value indicating whether to force all density values to be positive numbers. Default is FALSE.
verbose	Logical value indicating whether to issue warnings about numerical problems and conditions.

Value

This function produces the spatial density of x as an object of class im from the spatstat package.

Author(s)

Joshua French

References

Waller, L.A. and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley.

See Also

density.ppp

Examples

```
data(grave)
contour(spdensity(grave))
```

---

spscan.test *Spatial Scan Test*

Description

spscan.test performs the spatial scan test of Kulldorf (1997) for case/control point data.

Usage

```
spscan.test(x, case = 2, nsim = 499, alpha = 0.1, maxd = NULL,
  cl = NULL, longlat = FALSE)
```

Arguments

x	A ppp object from the spatstat package with marks for the case and control groups.
case	The position of the name of the "case" group in levels(x$marks). The default is 2. x$marks is assumed to be a factor. Automatic conversion is attempted if it is not.
nsim	The number of simulations from which to compute the p-value. A non-negative integer. Default is 499.
alpha	The significance level to determine whether a cluster is signficiant. Default is 0.1.
maxd	The radius of the largest possible cluster to consider. Default is NULL, i.e., no limit.
cl	A cluster object created by makeCluster, or an integer to indicate number of child-processes (integer values are ignored on Windows) for parallel evaluations.
longlat	A logical value indicating whether Euclidean distance (FALSE) or Great Circle (WGS84 ellipsoid, FALSE) should be used. Default is FALSE, i.e., Euclidean distance.

Details

The test is performed using the random labeling hypothesis. The windows are circular and extend from the observed data locations. The clusters returned are non-overlapping, ordered from most significant to least significant. The first cluster is the most likely to be a cluster. If no significant clusters are found, then the most likely cluster is returned (along with a warning).

Setting cl to a positive integer MAY speed up computations on non-Windows computers. However, parallelization does have overhead cost, and there are cases where parallelization results in slower computations.

Value

Returns a list of length two of class scan. The first element (clusters) is a list containing the significant, non-ovlappering clusters, and has the the following components:

coords	The centroid of the significant clusters.
r	The radius of the window of the clusters.
pop	The total population in the cluser window.
cases	The observed number of cases in the cluster window.
expected	The expected number of cases in the cluster window.
smr	Standarized mortaility ratio (observed/expected) in the cluster window.
rr	Relative risk in the cluster window.
propcases	Proportion of cases in the cluster window.
loglikrat	The loglikelihood ratio for the cluster window (i.e., the log of the test statistic).
pvalue	The pvalue of the test statistic associated with the cluster window.

The second element of the list is the centroid coordinates. This is needed for plotting purposes.

Author(s)

Joshua French

References

Kulldorff M., Nagarwalla N. (1995) Spatial disease clusters: Detection and Inference. Statistics in Medicine 14, 799-810.

Kulldorff, M. (1997) A spatial scan statistic. Communications in Statistics – Theory and Methods 26, 1481-1496.

Waller, L.A. and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley.

Examples

```
data(grave)
out = spscan.test(grave, nsim = 99)
plot(out, chars = c(1, 20), main = "most likely cluster")
# get warning if no significant cluster
out2 = spscan.test(grave, alpha = 0.001, nsim = 99)
```

Index