Package ‘csrplus’

April 12, 2015

Title Methods to Test Hypotheses on the Distribution of Spatial Point Processes

Date 2015-04-02

Version 1.03-0

Author Sarah Smith <`smirks2@hotmail.com`>

Maintainer Sarah Smith <`smirks2@hotmail.com`>

Description Includes two functions to evaluate the hypothesis of complete spatial randomness (csr) in point processes. The function 'mwin' calculates quadrat counts to estimate the intensity of a spatial point process through the moving window approach proposed by Bailey and Gatrell (1995). Event counts are computed within a window of a set size over a fine lattice of points within the region of observation. The function 'pielou' uses the nearest neighbor test statistic and asymptotic distribution proposed by Pielou (1959) to compare the observed point process to one generated under csr. The value can be compared to that given by the more widely used test proposed by Clark and Evans (1954).

Depends R (>= 3.0.2)

License GPL-2

Imports sp

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2015-04-12 22:58:23

R topics documented:

`csrplus-package` *Methods to Test Hypotheses on the Distribution of Spatial Point Processes*

Description

Includes two functions to evaluate the hypothesis of complete spatial randomness (csr) in point processes. The function 'mwin' calculates quadrat counts to estimate the intensity of a spatial point process through the moving window approach proposed by Bailey and Gatrell (1995). Event counts are computed within a window of a set size over a fine lattice of points within the region of observation. The function 'pielou' uses the nearest neighbor test statistic and asymptotic distribution proposed by Pielou (1959) to compare the observed point process to one generated under csr. The value can be compared to that given by the more widely used test proposed by Clark and Evans (1954).

Details

Package:	csrplus
Type:	Package
Version:	1.0
Date:	2015-04-02
License:	GPL-2

Author(s)

Sarah Smith <smirks2@hotmail.com>

`boundary` *Boundary for California Earthquake Data*

Description

The dataset contains the coordinates for the boundary of the "quake" dataset. The boundary is defined as the convex hull containing all of the observed events.

Usage

```
data(boundary)
```

Format

A data frame with 16 observations on the following 2 variables.

`Longitude`	a numeric vector containing the longitude values
`Latitude`	a numeric vector containing the latitude values

Source

The boundary was obtained using the "chull()" function.

Examples

```
data(boundary)
plot(boundary, main="Boundary", xlab="Latitude", ylab="Longitude", type="l")
```

`mwin` *Moving Window Approach to Quadrat Counts*

Description

Calculates quadrat counts to estimate the intensity of a spatial point process through the moving window approach proposed by Bailey and Gatrell (1995). Event counts are computed within a window of a set size over a fine lattice of points within the region of observation.

Usage

```
mwin(xcoord, ycoord, boundaryx, boundaryy, gridx, gridy, windowsizel, windowsizew)
```

Arguments

`xcoord`	a vector containing the x-coordinates of the observed point process
`ycoord`	a vector containing the y-coordinates of the observed point process
`boundaryx`	a vector containing the x-coordinates of the boundary
`boundaryy`	a vector containing the y-coordinates of the boundary
`gridx`	integer for the number of column lattice points
`gridy`	integer for the number of row lattice points
`windowsizel`	integer for the length of the window
`windowsizew`	integer for the width of the window

Details

The function first constructs a rectangular space based on the maximum and minimum values of the boundary. It then places a lattice over the rectangle, with a number of points in the lattice defined by the user. The "point.in.polygon" function determines whether or not a specific lattice point falls within the region of observation. If so, the function counts the number of events in a window of a specified size centered on that point. If not, the function moves on to the next lattice point. The function returns the location of lattice points within the region of observation in addition to the quadrat count at each point.

Value

xgrid a vector of the x-coordinates of the lattice points within the boundary

ygrid a vector of the y-coordinates of the lattice points within the boundary

quadrat a vector containing the number of events within each window sampled along the lattice

Source

Cressie, Noel. "Chapter 8: Spatial Point Patterns." Statistics for Spatial Data. Revised ed. New York: John Wiley and Sons, 1993. N. pag. Print.

Gatrell, Anthony C. "Chapter 3: Introductory Methods for Point Patterns." Interactive Spatial Data Analysis. By Trevor C. Bailey. N.p.: Routledge, 1995. N. pag. Print.

Examples

```
# To load data corresponding to the location of earthquakes in California:
data(quake)

# To load data corresponding to the boundary:
data(boundary)

# To compute quadrat counts with a 40 x 40 lattice and 1 x 1 unit window:
m <- mwin(quake[,3], quake[,2], boundary[,1], boundary[,2], 40, 40, 1, 1)

# To plot the results (with the shading corresponding to the quadrat count):
layout(matrix(c(1,2), nc=2), widths = c(4, 1))
palette(rev(heat.colors((max(as.numeric(m$quadrat))-min(as.numeric(m$quadrat))))))
plot(m$xgrid, m$ygrid, col=m$quadrat, pch=15, cex=.8,
     xlab="X-Coordinates", ylab="Y Coordinates", main="Quadrat Count")
lines(boundary[,1], boundary[,2])
breaks <- seq(min(as.numeric(m$quadrat)), (max(as.numeric(m$quadrat))), by=1)
plot.new()
plot.window(xlim = c(0, 1),ylim = range(breaks),xaxs = "i", yaxs = "i")
rect(0, breaks[-length(breaks)],1, breaks[-1],
col = rev(heat.colors(length(breaks) - 1)))
axis(2)
```

---

pielou *Nearest Neighbor Test Statistic: Pielou*

Description

Calculates the nearest neighbor test statistic for a spatial point process based on that proposed by Pielou (1959). The test statistic is based on sample point to event distances. It is equal to sum from i=1 to n of [(pi*lambda*(x_i)^2)/(n)], where x_i is equal to the point-to-event distance, lambda is equal to the intensity of the point process, and n is equal to the number of sample point-to-event distances. The test statistic follows a normal distribution with mean 1 and variance 1/n.

Usage

```
pielou(xcoord, ycoord, points, boundaryx, boundaryy, lambda)
```

Arguments

`xcoord`	a vector containing the x-coordinates of the observed point process
`ycoord`	a vector containing the y-coordinates of the observed point process
`points`	integer for the number of events for which to calculate point-to-event distances
`boundaryx`	a vector containing the x-coordinates of the boundary
`boundaryy`	a vector containing the y-coordinates of the boundary
`lambda`	integer for the estimated intensity of the point process (likely the area of the region of observation divided by the number of events)

Details

The function begins by proposing a sample point within the region of observation. It then calculates and stores the distance from this point to the nearest event. The process is repeated for the number of sample points specified by the user. Based on the stored distances, and the input for lambda, the test statistic of sum from i=1 to n of [(pi*lambda*(x_i)^2)/(n)] is calculated.

The test statistic serves as an alternative to that proposed by Clark and Evans (1954). Instead of comparing event-to-event distances, the function makes use of sample point-to-event distances. Knowing that the test statistic follows a normal distribution of mean 1 and variance 1/n, it can be used in the calculation of a z-statistic to evaluate the hypothesis of complete spatial randomness.

In order for the selection of sample points to not bias the results, it is suggested that the function be run multiple times to obtain an average test statistic .

The test statistic should be used with caution as the function does not account for edge effects. As points along the border will have larger nearest neighbor distances, the normal approximation of the test statistic will underestimate the mean distance. When looking at event-to-point distances, it is expected that distances will be larger in a clustered process than in a process that exhibits complete spatial randomness. While the event-to-point distance will be small if the random event happens to be located in a cluster, there is a high probability that the sample point will be located in a sparsely populated region and therefore have a large nearest neighbor distance. By underestimating the mean distance, there is consequently more evidence to reject the null hypothesis of complete spatial randomness.

While the test statistic can provide a good first assessment of the null hypothesis of complete spatial randomness, it should not be relied upon as a definitive measure. More accurate conclusions can likely be drawn by comparing the observed process to simulations of a random process generated over the specific region of observation.

Value

pstat a integer for the calcualted test statistic

Source

Cressie, Noel. "Chapter 8: Spatial Point Patterns." Statistics for Spatial Data. Revised ed. New York: John Wiley and Sons, 1993. N. pag. Print.

Gatrell, Anthony C. "Chapter 3: Introductory Methods for Point Patterns." Interactive Spatial Data Analysis. By Trevor C. Bailey. N.p.: Routledge, 1995. N. pag. Print.

Examples

```
# To load data corresponding to the location of earthquakes in California:
data(quake)

# To load data corresponding to the boundary:
data(boundary)

# To compute the one hundred values of the test statistic:
p <- mat.or.vec(100,1)
for (i in 1:100) {
   p[i] <- pielou(quake[,3], quake[,2], 30, boundary[,1], boundary[,2], 7.177) }
# To compute the average test statistic:
pavg <- mean(p)
# To calculate a z-statistic to evalute the null hypothesis of complete spatial randomness:
z <- (pavg-1)/sqrt(1/30)
```

`quake` *California Earthquakes*

Description

The dataset provides the location of earthquake epicenters within California for the period 1800-2000. All earthquakes are of a magnitude 5.50 or greater.

Usage

`data(quake)`

Format

A data frame with 383 observations on the following 4 variables.

`Date`	a numeric vector giving the date in (yyymmdd) format
`Latitude`	a numeric vector giving the latitude
`Longitude`	a numeric vector giving the longitude
`Magnitude`	a numeric vector giving the magnitude

Source

"California Earthquake History and Catalogs." CA.Gov. California Department of Conservation, n.d. Web. <http://www.consrv.ca.gov/cgs/rghm/quakes/Pages/index.aspx>.

Examples

```
data(quake)
plot(quake[,3], quake[,2], main="Quake Locations", xlab = "Long", ylab = "Lat", pch=20)
```

Index