Package: edgeCorr (via r-universe)

November 3, 2026

Type Package

Title Spatial Edge Correction

Version 1.0

Date 2016-03-12

Author Glenna Nightingale

Maintainer Glenna Nightingale <glenna.evans@gmail.com>

Description Facilitates basic spatial edge correction to point pattern data.

License GPL-2

Depends graphics

NeedsCompilation no

Repository https://glenna-nightingale.r-universe.dev

Date/Publication 2016-03-16 18:16:12 UTC

RemoteUrl https://github.com/cran/edgeCorr

RemoteRef HEAD

RemoteSha 17b34b7d50a7be3c15c1aae637f473683f9eeea7

Contents

edgeCorr-package 2
bec 3
reflection 4
translation 5
x 5
y 6

Index **7**

edgeCorr-package *edgeCorr*

Description

Basic spatial edge correction is facilitated. Three types are available: reflection, translation, and border edge correction. Edge correction (2,3,4) may be performed on point pattern data before analysis (2) or before constructing spatially dervied social networks (1).

Details

Package:	edgeCorr
Type:	Package
Version:	1.0
Date:	2016-02-25
License:	GPL-2

Author(s)

Glenna Nightingale
glenna.evans@gmail.com

References

1. Bayesian Spatial NBDA for Diffusion Data with Home-Base Coordinates. PLOS One. G.F. Nightingale, K.N. Laland, W. Hoppitt, P. Nightingale (2015)

2. Pairwise Interaction Point Processes for Modelling Bivariate Spatial Point Patterns in the Presence of Interaction Uncertainty. G. F. Nightingale, J. B. Illian, R. King. Journal of Environmental Statistics (2015)

3. Handbook of Spatial Point-Pattern Analysis in Ecology. T. Wiegand and K. A. Moloney (2014). CRC Press.

4. Statistical analysis and modelling of spatial point patterns. J. Illian and Antti Penttinen and Helga Stoyan and Dietrich Stoyan (2008). Wiley-Interscience.

Examples

```
data(x)
data(y)
datafile = data.frame(x[,1],y[,1])
bec(124,datafile,1,1,0.2)
bec(124,datafile,1,1,0.3)
```

bec *Border edge correction*

Description

This function facilitates border edge correction (minus sampling). This involves creating a border within the study area of a chosen width and using only the points within the border as focal points.

Social network indices
If a given spatial point pattern x is used to calculate metrics such as social network indices (Nightingale et. al, July 2015), where the close neighbours of each point is important, it may be necessary to account for edge effects or to perform edge correction. This is because the close neighbours of points at the edge of the point pattern may not all be observed, since some of these neighbouring points could actually exist outside of the point pattern.

Point process modelling
If a point pattern, x, is considered to be the realisation of a finite process P, defined only within the specified observation window, W, then the conditional intensity of P is observable within that window (Nightingale et. al, September 2015). If however, the process is unbounded, such that the data x is a partially observed realization of P, then issues of edge correction may arise since some of the points from the realization of P may fall on the edge of or outside of W. As a result of this, the conditional intensity of P may not be fully observed due to the presence of 'edge points' and 'unobserved' data points outside of W. As a result of this the conditional intensity of P may not be fully observed due to the presence of 'edge points' and 'unobserved' data points outside of W leading to systematic error in parameter estimation.

Using R notation, the reduced sample can be expressed as: $W_r = \{u \in W : B(u, r) \subset W\}$ where B(u,r) represents a disc or radius r centered at u.

Usage

```
bec(n, pointpattern,Length,Breadth, r)
```

Arguments

n	Number of points
pointpattern	Dataframe with two columns, x and y
Length	Length of survey plot
Breadth	Breadth of survey plot
r	Interaction radius

Value

A vector of the indices of the points which fall outside of the border. In addition, a plot of the survey area showing the points which fall outwith and within the border.

Author(s)

Glenna Nightingale

Examples

```
data(x)
data(y)
datafile = data.frame(x[,1],y[,1])
bec(124,datafile,1,1,0.2)
```

`reflection` *Reflection edge correction*

Description

This function creates an extra dataset of points that, when plotted are the mirror reflection of the specified edge of the original dataset.

Usage

```
reflection(pp, Length,Breadth, direction)
```

Arguments

`pp`	Dataframe with two columns labelled x and y respectively.
`Length`	Length of the study plot.
`Breadth`	Breadth of the study plot.
`direction`	The direction can be specified as either 1,2,3 or 4. 1,2,3,4 denote the right, left, top and bottom borders respectively.

Value

The xy coordinates for the additional dataset are returned.

Author(s)

Glenna Nightingale

Examples

```
data(x)
data(y)
datafile = data.frame(x[,1],y[,1])
reflection(datafile,1,1,4)
```

translation *Translation edge correction*

Description

This function creates an extra copy of the dataset in whichever direction (right, left, top or bottom) is requested.

Usage

```
translation(pp, Length,Breadth, direction)
```

Arguments

pp	Dataframe comprising of two rows labelled x and y.
Length	Length of study area
Breadth	Breadth of study area
direction	The direction can be specified as either 1,2,3 or 4. 1,2,3,4 denote the right, left, top and bottom borders respectively.

Value

The xy coordinates for the additional dataset are returned.

Author(s)

Glenna Nightingale

Examples

```
data(x)
data(y)
datafile = data.frame(x[,1],y[,1])
translation(datafile,1,1,1)
```

x *x*

Description

x coordinates for point pattern

Usage

```
data("x")
```

Examples

```
data(x)
## maybe str(x) ; plot(x) ...
```

---

y *y coordinates*

Description

y coordinates for a point pattern

Usage

```
data(y)
```

Examples

```
data(y)
## maybe str(y) ; plot(y) ...
```

Index

∗ **border**
    bec, 3
∗ **correction**
    bec, 3
    reflection, 4
    translation, 5
∗ **datasets**
    x, 5
    y, 6
∗ **edge**
    bec, 3
    edgeCorr-package, 2
    reflection, 4
    translation, 5
∗ **reflection**
    reflection, 4
∗ **translation**
    translation, 5

bec, 3

edgeCorr *(*edgeCorr-package*)*, 2
edgeCorr-package, 2

reflection, 4

translation, 5

x, 5

y, 6