Package 'echelon'

October 13, 2022

Type Package

Title The Echelon Analysis and the Detection of Spatial Clusters using Echelon Scan Method

Version 0.1.0

Author Fumio Ishioka

Maintainer Fumio Ishioka <`fishioka@okayama-u.ac.jp`>

Description Functions for the echelon analysis developed proposed by Myers et al. (1997) <doi:10.1023/A:1018518327329>, and the detection of spatial clusters using echelon scan method proposed by Kurihara (2003) <doi:10.20551/jscswabun.15.2_171>.

License GPL-3

Depends R (>= 3.3.0)

Suggests sf, spData

NeedsCompilation no

Repository CRAN

Date/Publication 2020-01-09 11:30:02 UTC

R topics documented:

`e.cluster.decision` *Echelon analysis for R*

Description

cluster detection using echelon

Usage

```
e.cluster.decision(reg_data, log.lambda)
```

Arguments

`reg_data`

`log.lambda`

`e.cluster.dendrogram` *Echelon analysis for R*

Description

draw the detected clusters on the dendrogram

Usage

```
e.cluster.dendrogram(echelon.obj, n.sim, cluster.legend.pos, cluster_reg, p_rank, para)
```

Arguments

`echelon.obj`

`n.sim`

`cluster.legend.pos`

`cluster_reg`

`p_rank`

`para`

`e.cluster.map` *Echelon analysis for R*

Description

draw the detected clusters on the map

Usage

```
e.cluster.map(x, c_separates, locs, coo, rin, p_rank, cluster_reg, n.sim, cluster.type)
```

Arguments

`x`

`c_separates`

`locs`

`coo`

`rin`

`p_rank`

`cluster_reg`

`n.sim`

`cluster.type`

`e.dendrogram` *Echelon analysis for R*

Description

draw the echelon dendrogram

Usage

```
e.dendrogram(peaks, locs, x, separates, c_separates, number, parents, pare_locs,
        progeny, symbols, col.symbols, cex.symbols, lwd, col, ens, adj.ens, col.ens,
            cex.ens, limb)
```

Arguments

`peaks`
`locs`
`x`
`separates`
`c_separates`
`number`
`parents`
`pare_locs`
`progeny`
`symbols`
`col.symbols`
`cex.symbols`
`lwd`
`col`
`ens`
`adj.ens`
`col.ens`
`cex.ens`
`limb`

`e.dendrogram.axis` *Echelon analysis for R*

Description

axis for echelon dendrogram

Usage

```
e.dendrogram.axis(main, ylab, yaxes, ylim, xaxes, xdper, dmai, peaks, x)
```

Arguments

`main`
`ylab`
`yaxes`
`ylim`
`xaxes`
`xdper`
`dmai`
`peaks`
`x`

`e.main` *Echelon analysis for R*

Description

main program

Usage

```
e.main(x, rin, T)
```

Arguments

`x`

`rin`

`T`

`e.monte.bin` *Echelon analysis for R*

Description

Monte Carlo testing based on Binomial model

Usage

```
e.monte.bin(rin,cas,pop,n.sim,K,cluster.type)
```

Arguments

`rin`

`cas`

`pop`

`n.sim`

`K`

`cluster.type`

`e.monte.poi` *Echelon analysis for R*

Description

Monte Carlo testing based on Poisson model

Usage

```
e.monte.poi(rin, cas, pop, ex, n.sim, K, cluster.type)
```

Arguments

`rin`

`cas`

`pop`

`ex`

`n.sim`

`K`

`cluster.type`

`e.profile` *Echelon analysis for R*

Description

echelon profiles

Usage

```
e.profile(peaks, parents, separates)
```

Arguments

`peaks`

`parents`

`separates`

`e.scan` *Echelon analysis for R*

Description

echelon scan based on the number of regions

Usage

```
e.scan(x, locs, peaks, c_separates, parents, K)
```

Arguments

`x`

`locs`

`peaks`

`c_separates`

`parents`

`K`

`e.scan.pop` *Echelon analysis for R*

Description

echelon scan based on a rate of population

Usage

```
e.scan.pop(x, locs, peaks, c_separates, parents, K, pop)
```

Arguments

`x`

`locs`

`peaks`

`c_separates`

`parents`

`K`

`pop`

`echebin` *Echelon spatial scan statistic based on Binomial model*

Description

`echebin` detects spatial clusters using echelon spatial scan statistic based on Binomial model.

Usage

```
echebin(echelon.obj, cas, ctl, K = length(cas)/2, n.sim = 99,
        cluster.type = "high", cluster.legend.pos = "bottomleft",
        dendrogram = TRUE, cluster.info = FALSE, coo = NULL, ...)
```

Arguments

`echelon.obj`	An object of class `echelon`. See `echelon`.
`cas`	A numeric (integer) vector of case counts. `NA`s are not allowed.
`ctl`	A numeric (integer) vector for control counts. `NA`s are not allowed.
`K`	Maximum cluster size. if `K` >= 1 (integer), the cluster size is limit to less than or equal to number of regions `K`. On the other hand, if 0 < `K` < 1, the cluster size is limit to less than or equal to `K` `*` 100% of the total population.
`n.sim`	Number of Monte Carlo replications used for significance testing of detected clusters. If 0, the significance is not assessed.
`cluster.type`	A character string specifying the cluster type. If `"high"`, the detected clusters have high rates (hotspot clusters). On the other hand, If `"low"`, the detected clusters have low rates (coldspot cluster).
`cluster.legend.pos`	A location of the legend on the dendrogram. (See the help for `legend`)
`dendrogram`	Logical. if TRUE, draw an echelon dendrogram with detected clusters.
`cluster.info`	Logical. if TRUE, return the result of echelon scan statistic.
`coo`	An array of the (x,y)-coordinates of the region centroid to draw a cluster map.
`...`	Related to dendrogram drawing. (See the help for `echelon`)

Value

`clusters`	Each detected cluster.
`scanned.regions`	A region list of all scanning processes.
`simulated.LLR`	Monte Carlo samples of the log-likelihood ratio.

Note

`echebin` requires either `cas` and `ctl`.

Population is defined by the sum of `cas` and `ctl`.

Typical values of `n.sim` are 99, 999, 9999, ...

Author(s)

Fumio Ishioka

References

[1] Kulldorff M, Nagarwalla N. (1995). Spatial disease clusters: Detection and inference. *Statistics in Medicine*, **14**, 799–810.

[2] Kulldorff M. (1997). A spatial scan statistic. *Communications in Statistics: Theory and Methods*, **26**, 1481–1496.

See Also

`echelon` for the echelon analysis.

`echepoi` for cluster detection based on echelons using Poisson model.

Examples

```
##Hotspot detection for non-white birth of North Carolina using echelon scan

#Non-white birth from 1974 to 1984 (case data)
library(spData)
data("nc.sids")
nwb <- nc.sids$NWBIR74 + nc.sids$NWBIR79

#White birth from 1974 to 1984 (control data)
wb <- (nc.sids$BIR74 - nc.sids$NWBIR74) + (nc.sids$BIR79 - nc.sids$NWBIR79)

#Hotspot detection based on Binomial model
nwb.echelon <- echelon(x = nwb/wb, nb = ncCR85.nb, name = row.names(nc.sids))
echebin(nwb.echelon, cas = nwb, ctl = wb, K = 20,
  main = "Hgih rate clusters", ens = FALSE)
text(nwb.echelon$coord, labels = nwb.echelon$regions.name,
  adj = -0.1, cex = 0.7)

#Detected clusters and neighbors map
#XY coordinates of each polygon centroid point
NC.coo <- cbind(nc.sids$lon, nc.sids$lat)
echebin(nwb.echelon, cas = nwb, ctl = wb, K = 20,
  coo = NC.coo, dendrogram = FALSE)


##Detected clusters map
#Here is an example using the sf class "sf"
nwb.clusters <- echebin(nwb.echelon, cas = nwb,
   ctl = wb, K = 20, dendrogram = FALSE)
MLC <- nwb.clusters$clusters[[1]]
Secondary <- nwb.clusters$clusters[[2]]
cluster.col <- rep(0,times=length(nwb))
cluster.col[MLC$regionsID] <- 2
cluster.col[Secondary$regionsID] <- 3
```

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf"))
plot(nc$geometry, col = cluster.col,
main = "Detected high rate clusters")
text(st_coordinates(st_centroid(st_geometry(nc))),
  labels = nc$CRESS_ID, cex =0.75)
legend("bottomleft",
c(paste("1- p-value:", MLC$p),
  paste("2- p-value:", Secondary$p)),
  text.col = c(2,3))
```

`echelon` *Echelon analysis for spatial data*

Description

`echelon` divides study area into structural entities consisting of peaks or foundations, say *'echelons'*, based on neighbor information, and draw its dendrogram.

Usage

```
echelon(x, nb, dendrogram = TRUE, name = NULL,
      main = NULL, ylab = NULL, yaxes = TRUE, ylim = NULL,
      xaxes = FALSE, xdper = c(0, 1), dmai = NULL,
      col = 1, lwd = 1, symbols = 4, cex.symbols = 1, col.symbols = 4,
      ens = TRUE, adj.ens = 1, cex.ens = 0.8, col.ens = 1,
      profiles = FALSE)
```

Arguments

`x`	A numeric vector of data values.
`nb`	Neighbor information data. An object of class `nb` or a weights matrix.
`name`	The region names. if NULL, it is assigned `1:length(x)`.
`dendrogram`	Logical. if TRUE, draw an echelon dendrogram.
`main`	Related to dendrogram drawing. An overall title for the dendrogram.
`ylab`	Related to dendrogram drawing. A title for the y axis.
`yaxes`	Related to dendrogram drawing. Logical. if TRUE, draw the y axis.
`ylim`	Related to dendrogram drawing. A scale of y axis given by `c(min, max)`.
`xaxes`	Related to dendrogram drawing. Logical. if TRUE, draw the x axis.
`xdper`	Related to dendrogram drawing. A display percentage of x axis. The full display percentage is given in *[0, 1]*.
`dmai`	Related to dendrogram drawing. A numerical vector of the form c(bottom, left, top, right) which gives the margin size specified in inches. Default is set to `c(0.4, 0.8, 0.3, 0.01)`.

`col`	Related to dendrogram drawing. A line color of the dendrogram.
`lwd`	Related to dendrogram drawing. A line width of the dendrogram.
`symbols`	Related to dendrogram drawing. Either an integer specifying a symbol or a single character. If integer, it is synonymous with `pch` in `par`.
`cex.symbols`	Related to dendrogram drawing. A magnification to be used for the plotting symbols.
`col.symbols`	Related to dendrogram drawing. A color to be used for the plotting symbols.
`ens`	Related to dendrogram drawing. Logical. if TRUE, draw the labels of echelon numbers.
`adj.ens`	Related to dendrogram drawing. Adjustment of the labels of the echelon numbers.(see the help for `text`("adj")).
`cex.ens`	Related to dendrogram drawing. A magnification to be used for the labels of echelon numbers.
`col.ens`	Related to dendrogram drawing. A color to be used for the labels of echelon numbers.
`profiles`	Logical. if TRUE, return the result of echelon profiles. (See [2] for the details of echelon profiles)

Value

The function `echelon` returns an object of class `echelon`. An object of class `echelon` contains the following components:

`Table`	Summary of each echelon.
`Echelons`	Regions that composes each echelon.

Note

If there are `NA` in `x`, then that is set the minimum value of `x`.

`Sf::st_read` and `spdep::poly2nb` are useful for creating the object specified in the argument `nb`.

Author(s)

Fumio Ishioka

References

[1] Myers, W.L., Patil, G.P. and Joly, K. (1997). Echelon approach to areas of concern in synoptic regional monitoring. *Environmental and Ecological Statistics*, **4**, 131–152.

[2] Kurihara, K., Myers, W.L. and Patil, G.P. (2000) Echelon analysis of the relationship between population and land cover patter based on remote sensing data. *Community ecology*, **1**, 103–122.

See Also

`echepoi` and `echebin` for cluster detection based on echelons.

Examples

```
##Echelon analysis for one-dimensional data with 25 regions
#A weights matrix
one.nb <- matrix(0,25,25)
one.nb[1,2] <- 1
for(i in 2:24) one.nb[i,c(i-1,i+1)] <- c(1,1)
one.nb[25,24] <- 1

#25 random values
one.dat <- runif(25) * 10

#Echelon analysis
echelon(x = one.dat, nb = one.nb)


##Echelon analysis for SIDS data for North Carolina
#Mortality rate per 1,000 live births from 1974 to 1984
library(spData)
data("nc.sids")
SIDS.cas <- nc.sids$SID74 + nc.sids$SID79
SIDS.pop <- nc.sids$BIR74 + nc.sids$BIR79
SIDS.rate <- SIDS.cas * 1000 / SIDS.pop

#Echelon analysis
SIDS.echelon <- echelon(x = SIDS.rate, nb = ncCR85.nb, name = row.names(nc.sids),
  symbols = 12, cex.symbols = 1.5, ens = FALSE)
text(SIDS.echelon$coord, labels = SIDS.echelon$regions.name,
  adj = -0.1, cex = 0.7)

#Echelon Profiles
echelon(x = SIDS.rate, nb = ncCR85.nb, profiles = TRUE)
```

`echepoi` *Echelon spatial scan statistic based on Poisson model*

Description

`echepoi` detects spatial clusters using echelon spatial scan statistic based on Poisson model.

Usage

```
echepoi(echelon.obj, cas, pop = NULL, ex = NULL, K = length(cas)/2, n.sim = 99,
        cluster.type = "high", cluster.legend.pos = "bottomleft",
        dendrogram = TRUE, cluster.info = FALSE, coo = NULL, ...)
```

Arguments

`echelon.obj`	An object of class `echelon`. See `echelon`.
`cas`	A numeric (integer) vector of case counts. `NA`s are not allowed.
`pop`	A numeric (integer) vector for population. `NA`s are not allowed.
`ex`	A numeric vector for expected cases. `NA`s are not allowed.
`K`	Maximum cluster size. if `K` >= 1 (integer), the cluster size is limit to less than or equal to number of regions `K`. On the other hand, if 0 < `K` < 1, the cluster size is limit to less than or equal to `K` * 100% of the total population.
`n.sim`	Number of Monte Carlo replications used for significance testing of detected clusters. If 0, the significance is not assessed.
`cluster.type`	A character string specifying the cluster type. If `"high"`, the detected clusters have high rates (hotspot clusters). On the other hand, If `"low"`, the detected clusters have low rates (coldspot cluster).
`cluster.legend.pos`	A location of the legend on the dendrogram. (See the help for `legend`)
`dendrogram`	Logical. if TRUE, draw an echelon dendrogram with detected clusters.
`cluster.info`	Logical. if TRUE, return the result of detected clusters for detail.
`coo`	An array of (x,y)-coordinates of the region centroid to draw a cluster map.
`...`	Related to dendrogram drawing. (See the help for `echelon`)

Value

`clusters`	Each detected cluster.
`scanned.regions`	A region list of all scanning processes.
`simulated.LLR`	Monte Carlo samples of the log-likelihood ratio.

Note

`echepoi` requires either `pop` or `ex`.

Typical values of `n.sim` are 99, 999, 9999, ...

Author(s)

Fumio Ishioka

References

[1] Kulldorff M. (1997). A spatial scan statistic. *Communications in Statistics: Theory and Methods*, **26**, 1481–1496.

[2] Ishioka F, Kawahara J, Mizuta M, Minato S, and Kurihara K. (2019) Evaluation of hotspot cluster detection using spatial scan statistic based on exact counting. *Japanese Journal of Statistics and Data Science*, **2**, 241–262.

See Also

`echelon` for the echelon analysis.

`echebin` for cluster detection based on echelons using Binomial model.

Examples

```
##Hotspot detection for SIDS data of North Carolina using echelon scan

#Mortality rate per 1,000 live births from 1974 to 1984
library(spData)
data("nc.sids")
SIDS.cas <- nc.sids$SID74 + nc.sids$SID79
SIDS.pop <- nc.sids$BIR74 + nc.sids$BIR79
SIDS.rate <- SIDS.cas * 1000 / SIDS.pop

#Hotspot detection based on Poisson model
SIDS.echelon <- echelon(x = SIDS.rate, nb = ncCR85.nb, name = row.names(nc.sids))
echepoi(SIDS.echelon, cas = SIDS.cas, pop = SIDS.pop, K = 20,
  main = "Hgih rate clusters", ens = FALSE)
text(SIDS.echelon$coord, labels = SIDS.echelon$regions.name,
  adj = -0.1, cex = 0.7)

#Detected clusters and neighbors map
#XY coordinates of each polygon centroid point
NC.coo <- cbind(nc.sids$lon, nc.sids$lat)
echepoi(SIDS.echelon, cas = SIDS.cas, pop = SIDS.pop, K = 20,
  coo = NC.coo, dendrogram = FALSE)


##Detected clusters map
#Here is an example using the sf class "sf"
SIDS.clusters <- echepoi(SIDS.echelon, cas = SIDS.cas,
  pop = SIDS.pop, K = 20, dendrogram = FALSE)
MLC <- SIDS.clusters$clusters[[1]]
Secondary <- SIDS.clusters$clusters[[2]]
cluster.col <- rep(0,times=length(SIDS.rate))
cluster.col[MLC$regionsID] <- 2
cluster.col[Secondary$regionsID] <- 3

library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf"))
plot(nc$geometry, col = cluster.col,
main = "Detected high rate clusters")
text(st_coordinates(st_centroid(st_geometry(nc))),
  labels = nc$CRESS_ID, cex =0.75)
legend("bottomleft",
  c(paste("1- p-value:", MLC$p),
  paste("2- p-value:", Secondary$p)),
  text.col = c(2,3))
```

`nlimb` *Echelon analysis for R*

Description

auxiliary of echelon profiles

Usage

```
nlimb(x, k2)
```

Arguments

`x`

`k2`

Index