Package 'smerc'

November 12, 2015

Type Package

Title Statistical Methods for Regional Counts

Version 0.2.2

Date 2015-11-12

Author Joshua French

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

Description
Provides statistical methods for the analysis of data areal data, with a focus on cluster detection.

License GPL (>= 2)

LazyLoad yes

Imports SpatialTools, fields, parallel, maps, smacpod, igraph, spdep, utils

Suggests testthat, sp

NeedsCompilation no

Repository CRAN

Date/Publication 2015-11-12 19:00:05

R topics documented:

`color.clusters` *Color clusters*

Description

`color.clusters` is a simple helper function that makes it easier to color clusters of regions produced by an appropriate method, e.g., `scan.test` or `uls.test`. Regions/clusters that are not part of any cluster have no color.

Usage

```
color.clusters(x, col = 2:(length(x$clusters) + 1))
```

Arguments

`x`	An object of class scan produced by a function such as `scan.test`.
`col`	A vector of colors to color the clusters in `x`. Should have same length as the number of clusters in `x`.

Value

Returns a vector with colors for each region/centroid for the data set used to construct `x`.

Author(s)

Joshua French

Examples

```
data(nydf)
coords = with(nydf, cbind(longitude, latitude))
out = scan.test(coords = coords, cases = floor(nydf$cases),
                 pop = nydf$pop, alpha = 0.12, lonlat = TRUE,
                 nsim = 49)
data(nypoly)
library(sp)
plot(nypoly, col = color.clusters(out))
```

flex.test *Flexibly Shaped Spatial Scan Test*

Description

flex.test performs the flexibly shaped spatial scan test of Tango and Takahashi (2005).

Usage

```
flex.test(coords, cases, pop, w, k = 10, ex = sum(cases)/sum(pop) * pop,
  type = "poisson", nsim = 499, alpha = 0.1, nreport = nsim + 1,
  lonlat = FALSE, parallel = TRUE)
```

Arguments

coords	An $n \times 2$ matrix of centroid coordinates for the regions.
cases	The number of cases in each region.
pop	The population size of each region.
w	An $n \times n$ adjacenty matrix for the regions in the study area.
k	An integer indicating the maximum number of regions to inclue in a potential cluster. Default is 10
ex	The expected number of cases for each region. The default is calculated under the constant risk hypothesis.
type	The type of scan statistic to implement. Default is "poisson". Alternative is "bernoulli".
nsim	The number of simulations from which to compute p-value.
alpha	The significance level to determine whether a cluster is signficant. Default is 0.05.
nreport	The frequency with which to report simulation progress. The default is nsim+ 1, meaning no progress will be displayed.
lonlat	If lonlat is TRUE, then the great circle distance is used to calculate the inter-centroid distance. The default is FALSE, which specifies that Euclidean distance should be used.
parallel	A logical indicating whether the test should be parallelized using the parallel::mclapply function. Default is TRUE. If TRUE, no progress will be reported.

Details

The test is performed using the spatial scan test based on the Poisson test statistic and a fixed number of cases. 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).

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.
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

Tango, T., & Takahashi, K. (2005). A flexibly shaped spatial scan statistic for detecting clusters. International journal of health geographics, 4(1), 11. Kulldorff, M. (1997) A spatial scan statistic. Communications in Statistics – Theory and Methods 26, 1481-1496.

Examples

```
data(nydf)
data(nyw)
coords = with(nydf, cbind(longitude, latitude))
out = flex.test(coords = coords, cases = floor(nydf$cases),
                w = nyw, k = 3,
                pop = nydf$pop, nsim = 49,
                alpha = 0.12, lonlat = TRUE)

data(nypoly)
library(sp)
plot(nypoly, col = color.clusters(out))
```

flex.zones *Determine zones for flexibly shaped spatial scan test*

Description

flex.zones determines the unique zones to consider for the flexibly shaped spatial scan test of Tango and Takahashi (2005).

Usage

```
flex.zones(coords, w, k = 10, lonlat = FALSE)
```

Arguments

coords	The number of cases in each region.
w	The binary spatial adjacency matrix.
k	The maximum number of regions to include in a zone.
lonlat	A logical indioating whether the coordinates are longitude/latitude. If so, the great circle distance is used in computing the nearest/neighbor distance matrix.

Value

Returns a list of zones to consider for clustering. Each element of the list contains a vector with the location ids of the regions in that zone.

Author(s)

Joshua French

References

Tango, T., & Takahashi, K. (2005). A flexibly shaped spatial scan statistic for detecting clusters. International journal of health geographics, 4(1), 11.

Examples

```
data(nydf)
data(nyw)
coords = cbind(nydf$longitude, nydf$latitude)
flex.zones(coords = coords, w = nyw, k = 3, lonlat = TRUE)
```

nnpop *Determine nearest neighbors*

Description

nnpop determines the nearest neighbors for a set of observations based on the distance matrix according to a population upperbound.

Usage

```
nnpop(d, pop, ubpop)
```

Arguments

d	An $n \times n$ square distance matrix containing the intercentroid distance between the n region centroids.
pop	A vector of length n containing the population values of the n region centroids.
ubpop	A proportion between 0 and 1 containing the upperbound for the proportion of total population contained collectively among a set of nearest neighbors.

Details

This function determines the nearest neighbors of each centroid based on the intercentroid distance. The number of nearest neighbors is limited by the sum of the population values among the nearest neighbors. The set of nearest neighbors can contain no more than `ubpop * sum(pop)` members of the population. The nearest neighbors are ordered from nearest to farthest.

Value

Returns the indexes of the nearest neighbors as a list. For each element of the list, the indexes are ordered from nearest to farthest from each centroid.

Author(s)

Joshua French

Examples

```
data(nydf)
d = SpatialTools::dist1(as.matrix(nydf[,c("longitude", "latitude")]))
nnout = nnpop(d, pop = nydf$pop, ubpop = 0.5)
```

nydf *Leukemia data for 281 regions in New York.*

Description

This data set contains 281 observations related to leukeumia cases in an 8 county area of the state of New York. The data were made available in Waller and Gotway (2005) and details are provided there. These data are related to a similar data set in Waller et al. (1994). The longitude and latitude coordinates are taken from the NYleukemia data set in the SpatialEpi package for plotting purposes.

Usage

```
data(nydf)
```

Format

A data frame with 281 rows and 4 columns:

longitude The longitude of the region centroid. These are NOT the original values provided by Waller and Gotway (2005), but are the right ones for plotting correctly.

latitude The latitude of the region centroid. These are NOT the original values provided by Waller and Gotway (2005), but are the right ones for plotting correctly.

population The population (1980 census) of the region.

cases The number of leukemia cases between 1978-1982.

x The original 'longitude' coordinate provided by Waller and Gotway (2005).

y The original 'latitude coordinate provided by Waller and Gotway (2005).

Source

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

References

Waller, L.A., Turnbull, B.W., Clark, L.C., and Nasca, P. (1994) "Spatial Pattern Analysis to Detect Rare Disease Clusters" in Case Studies in Biometry, N. Lange, L. Ryan, L. Billard, D. Brillinger, L. Conquest, and J. Greenhouse (eds.) New York: John Wiley and Sons.

nypoly *SpatialPolygonsDataFrame for New York leukemia data.*

Description

A SpatialPolygonsDataFrame for the New York leukemia data in `nydf`. Note that the coordinates in the polygon have been projected to a different coordinate system (UTM, zone 18), but the order of the regions/polygons is the same as in `nydf`. This data comes from

Usage

```
data(nypoly)
```

Format

A SpatialPolygonDataFrame

Source

Bivand, R. S., Pebesma, E. J., Gomez-Rubio, V., and Pebesma, E. J. (2013). Applied Spatial Data Analysis with R, 2nd edition. New York: Springer.

nyw *Adjacency matrix for New York leukemia data.*

Description

This data set contains a 281 x 281 adjacency matrix for the New York leukemia data in `nydf`.

Usage

```
data(nyw)
```

Format

A matrix of dimension 281 x 281.

Source

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

References

Waller, L.A., Turnbull, B.W., Clark, L.C., and Nasca, P. (1994) "Spatial Pattern Analysis to Detect Rare Disease Clusters" in Case Studies in Biometry, N. Lange, L. Ryan, L. Billard, D. Brillinger, L. Conquest, and J. Greenhouse (eds.) New York: John Wiley and Sons.

plot.scan *Plots object of class* scan.

Description

Plots clusters (the centroids of the regions in each cluster) in different colors. The most likely cluster is plotted with solid red circles by default. Points not in a cluster are black open circles. The other cluster points are plotted with different symbols and colors.

Usage

```
## S3 method for class 'scan'
plot(x, ..., ccol = NULL, cpch = NULL, usemap = FALSE,
  mapargs = list())
```

Arguments

x	An object of class scan to be plotted.
...	Additional graphical parameters passed to plot function.
ccol	Fill color of the plotted points. Default is NULL, indicating red for the most likely cluster, and col = 3, 4, ..., up to the remaining number of clusters.
cpch	Plotting character to use for points in each cluster. Default is NULL, indicating pch = 20 for the most likely cluster and then pch = 2, 3, .., up to the remaining number of clusters.
usemap	Logical indicating whether the maps::map function should be used to create a plot background for the coordinates. Default is FALSE. Use TRUE if you have longitude/latitude coordinates.
mapargs	A list of arguments for the map function.

See Also

map

Examples

```
data(nydf)
coords = with(nydf, cbind(longitude, latitude))
out = scan.test(coords = coords, cases = floor(nydf$cases),
                pop = nydf$pop, nsim = 49,
                lonlat = TRUE, alpha = 0.12,
                parallel = FALSE)
## plot output for new york state
# specify desired argument values
mapargs = list(database = "state", region = "new york",
xlim = range(out$coords[,1]), ylim = range(out$coords[,2]))
# needed for "state" database (unless you execute library(maps))
data(stateMapEnv, package = "maps")
plot(out, usemap = TRUE, mapargs = mapargs)
```

scan.stat *Scan Statistic*

Description

scan.stat calculates the scan statistic for various distributions.

Usage

```
scan.stat(yin, ein, eout, ty, type = "poisson")
```

Arguments

yin	The sum of the response values inside the window. Generally, the sum of the cases.
ein	The expected value of the response in the window. Generally, the estimated overall risk for all regions combined, multiplied by the population size of the window.
eout	The expected value of the response outside the window.
ty	The sum of all responses in the study area. Generally, the total number of cases.
type	The type of scan statistic to implement. Currently, only "poisson" is implemented.

Value

A vector of scan statistics.

Author(s)

Joshua French

References

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

Examples

```
# statistic for most likely cluster of New York leukemia data
scan.stat(106, 62.13, 552 - 62.13, 552)
```

scan.test *Spatial Scan Test*

Description

scan.test performs the spatial scan test of Kulldorf (1997).

Usage

```
scan.test(coords, cases, pop, ex = sum(cases)/sum(pop) * pop,
  type = "poisson", nsim = 499, alpha = 0.1, nreport = nsim + 1,
  ubpop = 0.5, lonlat = FALSE, parallel = TRUE)
```

Arguments

coords	An $n \times 2$ matrix of centroid coordinates for the regions.
cases	The number of cases in each region.
pop	The population size of each region.
ex	The expected number of cases for each region. The default is calculated under the constant risk hypothesis.
type	The type of scan statistic to implement. Default is "poisson".
nsim	The number of simulations from which to compute p-value.
alpha	The significance level to determine whether a cluster is signficant. Default is 0.05.
nreport	The frequency with which to report simulation progress. The default is nsim+ 1, meaning no progress will be displayed.
ubpop	The upperbound of the proportion of the total population to consider for a cluster.
lonlat	If lonlat is TRUE, then the great circle distance is used to calculate the inter-centroid distance. The default is FALSE, which specifies that Euclidean distance should be used.
parallel	A logical indicating whether the test should be parallelized using the parallel::mclapply function. Default is TRUE. If TRUE, no progress will be reported.

Details

The test is performed using the spatial scan test based on the Poisson test statistic and a fixed number of cases. 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).

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:

locids	The location ids of regions in a significant cluster.
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.
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

Waller, L.A. and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley. Kulldorff, M. (1997) A spatial scan statistic. Communications in Statistics – Theory and Methods 26, 1481-1496.

Examples

```
data(nydf)
coords = with(nydf, cbind(longitude, latitude))
out = scan.test(coords = coords, cases = floor(nydf$cases),
                pop = nydf$pop, nsim = 49,
                alpha = 0.12, lonlat = TRUE)
## plot output for new york state
# specify desired argument values
mapargs = list(database = "state", region = "new york",
xlim = range(out$coords[,1]), ylim = range(out$coords[,2]))
# needed for "state" database (unless you execute library(maps))
data(stateMapEnv, package = "maps")
plot(out, usemap = TRUE, mapargs = mapargs)

# a second example to match the results of Waller and Gotway (2005)
# in chapter 7 of their book (pp. 220-221).
# Note that the 'longitude' and 'latitude' used by them has
# been switched.  When giving their input to SatScan, the coords
# were given in the order 'longitude' and 'latitude'.
# However, the SatScan program takes coordinates in the order
```

```
# 'latitude' and 'longitude', so the results are slightly different
# from the example above.
coords = with(nydf, cbind(y, x))
out2 = scan.test(coords = coords, cases = floor(nydf$cases),
                 pop = nydf$pop, nsim = 49,
                 alpha = 0.5, lonlat = TRUE)
# the cases observed for the clusters in Waller and Gotway: 117, 47, 44
# the second set of results match
c(out2$clusters[[1]]$cases, out2$clusters[[2]]$cases, out2$clusters[[3]]$cases)
```

`uls.test` *Upper Level Set Spatial Scan Test*

Description

`uls.test` performs Upper Level Set (ULS) spatian scan test of Patil and Taillie (2004).

Usage

```
uls.test(coords, cases, pop, w, ex = sum(cases)/sum(pop) * pop, nsim = 499,
  alpha = 0.1, nreport = nsim + 1, ubpop = 0.5, lonlat = FALSE,
  parallel = TRUE)
```

Arguments

`coords`	An $n \times 2$ matrix of centroid coordinates for the regions.
`cases`	The number of cases in each region.
`pop`	The population size of each region.
`w`	The binary spatial adjacency matrix.
`ex`	The expected number of cases for each region. The default is calculated under the constant risk hypothesis.
`nsim`	The number of simulations from which to compute p-value.
`alpha`	The significance level to determine whether a cluster is signficant. Default is 0.05.
`nreport`	The frequency with which to report simulation progress. The default is `nsim+ 1`, meaning no progress will be displayed.
`ubpop`	The upperbound of the proportion of the total population to consider for a cluster.
`lonlat`	If lonlat is TRUE, then the great circle distance is used to calculate the inter-centroid distance. The default is FALSE, which specifies that Euclidean distance should be used.
`parallel`	A logical indicating whether the test should be parallelized using the `parallel::mclapply function`. Default is TRUE. If TRUE, no progress will be reported.

Details

The test is performed using the spatial scan test based on the Poisson test statistic and a fixed number of cases. The windows are based on the Upper Level Sets proposed by Patil and Taillie (2004). 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).

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:

`locids`	The location ids of regions in a significant cluster.
`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.
`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

Waller, L.A. and Gotway, C.A. (2005). Applied Spatial Statistics for Public Health Data. Hoboken, NJ: Wiley. Kulldorff, M. (1997) A spatial scan statistic. Communications in Statistics – Theory and Methods 26, 1481-1496.

Examples

```
data(nydf)
data(nyw)
coords = with(nydf, cbind(longitude, latitude))
out = uls.test(coords = coords, cases = floor(nydf$cases),
                  pop = nydf$pop, w = nyw,
                  alpha = 0.12, lonlat = TRUE,
                  nsim = 10, ubpop = 0.1)
## plot output for new york state
# specify desired argument values
mapargs = list(database = "state", region = "new york",
xlim = range(out$coords[,1]), ylim = range(out$coords[,2]))
# needed for "state" database (unless you execute library(maps))
data(stateMapEnv, package = "maps")
```

```
plot(out, usemap = TRUE, mapargs = mapargs)

data(nypoly)
library(sp)
plot(nypoly, col = color.clusters(out))
```

`uls.zones` *Determine sequence of ULS zones.*

Description

`uls.zones` determines the unique zones obtained by implementing the ULS (Upper Level Set) method of Patil and Taillie (2004).

Usage

```
uls.zones(cases, pop, w, ubpop = 0.5)
```

Arguments

`cases`	The number of cases in each region.
`pop`	The population size of each region.
`w`	The binary spatial adjacency matrix.
`ubpop`	The upperbound of the proportion of the total population to consider for a cluster.

Details

The zones returned must have a total population less than ubpop * the total population of all regions in the study area.

Value

Returns a list of zones to consider for clustering. Each element of the list contains a vector with the location ids of the regions in that zone.

Author(s)

Joshua French

References

Patil, G. P., and Taillie, C. (2004). Upper level set scan statistic for detecting arbitrarily shaped hotspots. Environmental and Ecological Statistics, 11(2), 183-197.

Examples

```
data(nydf)
data(nyw)
uls.zones(cases = nydf$cases, pop = nydf$population, w = nyw)
```

Index