Package 'smerc'

April 20, 2018

Type Package

Title Statistical Methods for Regional Counts

Version 0.4.5

Date 2018-04-18

Author Joshua French

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

Description Implements statistical methods for analyzing the counts of areal data, with a focus on the detection of spatial clusters and clustering.

License GPL (>= 2)

LazyLoad yes

Imports SpatialTools, fields, parallel, maps, smacpod, spdep, matrixStats, sp

Suggests testthat, SpatialEpi

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-04-19 23:47:22 UTC

R topics documented:

bn.test 2
casewin 4
color.clusters 5
dmst.test 6
dmst.zones 8
dweights 9
flex.test 11
flex.zones 13
mlf.test 14
mlf.zones 16
nnpop 17

nydf 18
nypoly 19
nyw 20
plot.scan 20
plot.tango 21
scan.stat 22
scan.test 23
scan.zones 25
tango.stat 26
tango.test 27
uls.test 29
uls.zones 31

Index **32**

`bn.test` *Besag-Newell Test*

Description

`bn.test` implements the Besag-Newell test of Besag and Newell (1991) for finding disease clusters.

Usage

```
bn.test(coords, cases, pop, cstar, alpha = 0.1, lonlat = FALSE,
  noc = TRUE, modified = FALSE)
```

Arguments

`coords`	An $n \times 2$ matrix of centroid coordinates for the regions.
`cases`	The number of cases observed in each region.
`pop`	The population size associated with each region.
`cstar`	A non-negative integer indicating the minimum number of cases to include in each window.
`alpha`	The significance level to determine whether a cluster is signficant. Default is `0.10`.
`lonlat`	The default is `FALSE`, which specifies that Euclidean distance should be used.If `lonlat` is `TRUE`, then the great circle distance is used to calculate the inter-centroid distance.
`noc`	A logical value indicating whether all significant clusters should be returned (`FALSE`) or only the non-overlapping clusters (`TRUE`) arranged in order of significance. The default is `TRUE`.
`modified`	A logical value indicating whether a modified version of the test should be performed. The original paper recommends computing the p-value for each cluster as `1 - ppois(cstar - 1, lambda = expected)`. The modified version replaces `cstar` with `cases`, the observed number of cases in the region, and computes the p-value for the cluster as `1 - ppois(cases - 1, lambda = ex)`. The default is `modified = FALSE`.

Value

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

`locids`	The location ids of regions in a significant cluster.
`coords`	The centroid of the initial region.
`r`	The maximum radius of the cluster (in terms of intercentroid distance from the starting region).
`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.
`tstat`	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.
`w`	The adjacency matrix of the cluster.

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

Author(s)

Joshua French

References

Besag, J. and Newell, J. (1991). The detection of clusters in rare diseases, Journal of the Royal Statistical Society, Series A, 154, 327-333.

See Also

`scan.stat`, `plot.scan`, `scan.test`, `flex.test`, `dmst.test`, `uls.test`, `mlf.test`

Examples

```
data(nydf)
data(nyw)
coords = with(nydf, cbind(x, y))
out = bn.test(coords = coords, cases = nydf$cases,
              pop = nydf$pop, cstar = 6,
              alpha = 0.1)
plot(out)

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

`casewin` *Determine case windows (circles)*

Description

`casewin` determines the case windows (circles) for the Besag-Newell method.

Usage

```
casewin(d, cases, cstar)
```

Arguments

`d`	An $n \times n$ square distance matrix containing the intercentroid distance between the n region centroids.
`cases`	A vector of length n containing the observed number of cases for the n region centroids.
`cstar`	A non-negative integer indicating the minimum number of cases to include in each window.

Details

Using the distances provided in `d`, for each observation, the nearest neighbors are included in an increasingly large window until at least `cstar` cases are included in the window. Each row of `d` is matched with the same position in `cases`.

Value

Returns the indices of the regions in each case window as a list. For each element of the list, the indices are ordered from nearest to farthest from each centroid (and include the starting region).

Author(s)

Joshua French

References

Besag, J. and Newell, J. (1991). The detection of clusters in rare diseases, Journal of the Royal Statistical Society, Series A, 154, 327-333.

Examples

```
data(nydf)
coords = as.matrix(nydf[,c("longitude", "latitude")])
d = sp::spDists(coords, longlat = FALSE)
cwins = casewin(d, cases = nydf$cases, cstar = 6)
```

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))
```

`dmst.test` *Dynamic minimum spanning tree scan test*

Description

`dmst.test` implements the Dynamic Minimum Spanning Tree scan test of Assuncao et al. (2006). Starting with a single region as a current zone, new candidate zones are constructed by combining the current zone with the connected region that maximizes the resulting likelihood ratio test statistic. This is procedure repeated until the population or distance upper bound are reached. The same procedure is repeated for each region. The maxima likelihood first scan test proposed by Yao et al. (2011) is an independent variant of this, but only searches from the starting region that maximizes the likelihood ratio scan statistic. 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).

Usage

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

Arguments

`coords`	An $n \times 2$ matrix of centroid coordinates for the regions.
`cases`	The number of cases observed in each region.
`pop`	The population size associated with each region.
`w`	A 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 the p-value.
`alpha`	The significance level to determine whether a cluster is signficant. Default is `0.10`.
`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.
`ubd`	The upperbound for the radius of a cluster. This should be a proportion in (0, 1]. The value is the proportion of the maximum intercentroid distance between any two locations in `coords`. See Details.
`lonlat`	The default is `FALSE`, which specifies that Euclidean distance should be used.If `lonlat` is `TRUE`, then the great circle distance is used to calculate the intercentroid distance.
`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 maximum intercentroid distance can be found by executing the command: `sp::spDists(as.matrix(coords), longla` based on the specified values of `coords` and `lonlat`.

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

Assuncao, R.M., Costa, M.A., Tavares, A. and Neto, S.J.F. (2006). Fast detection of arbitrarily shaped disease clusters, Statistics in Medicine, 25, 723-742.

Yao, Z., Tang, J., & Zhan, F. B. (2011). Detection of arbitrarily-shaped clusters using a neighbor-expanding approach: A case study on murine typhus in South Texas. International journal of health geographics, 10(1), 1.

See Also

`scan.stat`, `plot.scan`, `scan.test`, `flex.test`, `uls.test`, `bn.test`

Examples

```
data(nydf)
data(nyw)
coords = with(nydf, cbind(longitude, latitude))
## Not run:
out = dmst.test(coords = coords, cases = floor(nydf$cases),
                pop = nydf$pop, w = nyw,
                alpha = 0.12, lonlat = TRUE,
                nsim = 5, ubpop = 0.1, ubd = 0.2)
data(nypoly)
library(sp)
```

```
plot(nypoly, col = color.clusters(out))
## End(Not run)
```

`dmst.zones` *Determine zones using the dynamic minimum spanning tree scan test of Assuncao et al. (2006)*

Description

`dmst.zones` determines the zones that produce the largest test statistic using a greedy algorithm. Specifically, starting individually with each region as a starting zone, new (connected) regions are added to the current zone in the order that results in the largest likelihood ratio test statistic. This is used to implement the dynamic minimum spanning tree (dmst) scan test of Assuncao et al. (2006).

Usage

```
dmst.zones(coords, cases, pop, w, ex = sum(cases)/sum(pop) * pop,
  ubpop = 0.5, ubd = 1, lonlat = FALSE, parallel = FALSE,
  maxonly = FALSE)
```

Arguments

`coords`	An $n \times 2$ matrix of centroid coordinates for the regions.
`cases`	The number of cases observed in each region.
`pop`	The population size associated with each region.
`w`	A binary spatial adjacency matrix.
`ex`	The expected number of cases for each region. The default is calculated under the constant risk hypothesis.
`ubpop`	The upperbound of the proportion of the total population to consider for a cluster.
`ubd`	The upperbound for the radius of a cluster. This should be a proportion in (0, 1]. The value is the proportion of the maximum intercentroid distance between any two locations in `coords`. See Details.
`lonlat`	The default is `FALSE`, which specifies that Euclidean distance should be used.If `lonlat` is `TRUE`, then the great circle distance is used to calculate the intercentroid distance.
`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.
`maxonly`	A logical value indicating whether to return only the maximum test statistic across all candidate zones. Default is `FALSE`.

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).

Every zone considered must have a total population less than `ubpop * sum(pop)`. Additionally, the maximum intercentroid distance for the regions within a zone must be no more than `ubd * the maximum intercentroid distance across all regions`.

Value

Returns a list of zones to consider for clustering that includes the location id of each zone and the associated test statistic, number of cases, expected number of cases, and the population in the zone. If `maxonly = TRUE`, then only the maximum test statistic across all of these zones is returned.

Author(s)

Joshua French

References

Assuncao, R.M., Costa, M.A., Tavares, A. and Neto, S.J.F. (2006). Fast detection of arbitrarily shaped disease clusters, Statistics in Medicine, 25, 723-742.

Examples

```
data(nydf)
data(nyw)
coords = as.matrix(nydf[,c("longitude", "latitude")])
# find zone with max statistic starting from each individual region
max_zones = dmst.zones(coords, cases = floor(nydf$cases),
                       nydf$pop, w = nyw, ubpop = 0.25,
                       ubd = .25, lonlat = TRUE)
head(max_zones)
```

---

dweights *Distance-based weights*

Description

`dweights` constructs a distance-based weights matrix. The `dweights` function can be used to construct a weights matrix `w` using the method of Tango (1995), Rogerson (1999), or a basic style.

Usage

```
dweights(coords, kappa = 1, lonlat = FALSE, type = "basic",
  cases = NULL, pop = NULL)
```

Arguments

coords	An $n \times 2$ matrix of centroid coordinates for the regions.
kappa	A positive constant related to strength of spatial autocorrelation.
lonlat	The default is FALSE, which specifies that Euclidean distance should be used.If lonlat is TRUE, then the great circle distance is used to calculate the inter-centroid distance.
type	The type of weights matrix to construct. Current options are "basic", "tango", and "rogerson". Default is "basic". See Details.
cases	The number of cases observed in each region.
pop	The population size associated with each region.

Details

coords is used to construct an $n \times n$ distance matrix d.

If type = "basic", then $w_{ij} = exp(-d_{ij}/\kappa)$.

If type = "rogerson", then $w_{ij} = exp(-d_{ij}/\kappa)/\sqrt{(cases_i/pop_i * cases_j/pop_j)}$.

If type = "tango", then $w_{ij} = exp(-4 * d_{ij}^2/\kappa^2)$.

Value

Returns an $n \times n$ matrix of weights.

Author(s)

Joshua French

References

Tango, T. (1995) A class of tests for detecting "general" and "focused" clustering of rare diseases. Statistics in Medicine. 14:2323-2334.

Rogerson, P. (1999) The Detection of Clusters Using A Spatial Version of the Chi-Square Goodness-of-fit Test. Geographical Analysis. 31:130-147

See Also

tango.test

Examples

```
data(nydf)
coords = as.matrix(nydf[,c("longitude", "latitude")])
w = dweights(coords, kappa = 1)
```

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 observed in each region.
pop	The population size associated with each region.
w	A binary spatial adjacency matrix.
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". Only "poisson" is currently implemented.
nsim	The number of simulations from which to compute the p-value.
alpha	The significance level to determine whether a cluster is signficant. Default is 0.10.
nreport	The frequency with which to report simulation progress. The default is nsim+ 1, meaning no progress will be displayed.
lonlat	The default is FALSE, which specifies that Euclidean distance should be used.If lonlat is TRUE, then the great circle distance is used to calculate the inter-centroid distance.
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.

See Also

scan.stat, plot.scan, scan.test, uls.test, dmst.test, bn.test

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). The algorithm uses a breadth-first search to find all subgraphs connected to each vertex (region) in the data set of size k or less.

Usage

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

Arguments

coords	An $n \times 2$ matrix of centroid coordinates for the regions.
w	A binary spatial adjacency matrix.
k	An integer indicating the maximum number of regions to inclue in a potential cluster. Default is 10
lonlat	The default is FALSE, which specifies that Euclidean distance should be used.If lonlat is TRUE, then the great circle distance is used to calculate the inter-centroid distance.
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.

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)
```

`mlf.test` *Maxima Likelihood First Scan Test*

Description

`mlf.test` implements the Maxima Likelihood First scan test of Yao et al. (2011), which is actually a special case of the Dynamic Minimum Spanning Tree of Assuncao et al. (2006). Find the single region that maximizes the likelihood ratio test statistic. Starting with this single region as a current zone, new candidate zones are constructed by combining the current zone with the connected region that maximizes the likelihood ratio test static. This procedure is repeated until the population upper bound is reached.

Usage

```
mlf.test(coords, cases, pop, w, ex = sum(cases)/sum(pop) * pop, nsim = 499,
  alpha = 0.1, nreport = nsim + 1, ubpop = 0.5, ubd = 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.
`ubd`	The upperbound for the proportion of the maximum intercentroid distance to allow for the maximum size of a zone.
`lonlat`	If lonlat is TRUE, then the great circle distance is used to calculate the intercentroid 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

Only a single cluster is ever returned because the algorithm only constructs a single sequence of starting zones, and overlapping zones are not returned. Only the zone that maximizes the likelihood ratio test statistic is returned.

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.
`w`	The adjacency matrix of the cluster.
`r`	The maximum radius of the cluster (in terms of intercentroid distance from the starting region).

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

Author(s)

Joshua French

References

Yao, Z., Tang, J., & Zhan, F. B. (2011). Detection of arbitrarily-shaped clusters using a neighbor-expanding approach: A case study on murine typhus in South Texas. International journal of health geographics, 10(1), 1.

Assuncao, R.M., Costa, M.A., Tavares, A. and Neto, S.J.F. (2006). Fast detection of arbitrarily shaped disease clusters, Statistics in Medicine, 25, 723-742.

Examples

```
data(nydf)
data(nyw)
coords = with(nydf, cbind(longitude, latitude))
out = mlf.test(coords = coords, cases = floor(nydf$cases),
                  pop = nydf$pop, w = nyw,
                  alpha = 0.12, lonlat = TRUE,
                  nsim = 10, ubpop = 0.1, ubd = 0.5)
data(nypoly)
library(sp)
plot(nypoly, col = color.clusters(out))
```

`mlf.zones` *Determine the candidate zone using the maxima likelihood first algorithm of Yao et al. (2011).*

Description

`mlf.zones` determines the most likely cluster zone obtained by implementing the maxima likelihood first scann method of Yao et al. (2011). Note that this is really just a special case of the dynamic minimum spanning tree (SMST) algorithm of Assuncao et al. (2006)

Usage

```
mlf.zones(coords, cases, pop, w, ex = sum(cases)/sum(pop) * pop,
  ubpop = 0.5, ubd = 1, lonlat = FALSE, parallel = TRUE,
  type = "pruned")
```

Arguments

`coords`	An $n \times 2$ matrix of centroid coordinates for the regions.
`cases`	The number of cases observed in each region.
`pop`	The population size associated with each region.
`w`	A binary spatial adjacency matrix.
`ex`	The expected number of cases for each region. The default is calculated under the constant risk hypothesis.
`ubpop`	The upperbound of the proportion of the total population to consider for a cluster.
`ubd`	The upperbound for the radius of a cluster. This should be a proportion in (0, 1]. The value is the proportion of the maximum intercentroid distance between any two locations in `coords`. See Details.
`lonlat`	The default is `FALSE`, which specifies that Euclidean distance should be used.If `lonlat` is `TRUE`, then the great circle distance is used to calculate the intercentroid distance.
`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.
`type`	One of `"maxonly"`, `"pruned"`, or `"all"`. Specifying `"maxonly"` returns only the maximum test statistic across all candidate zones, `"pruned"` returns information for the zone with the largest test statistic, while `"all"` returns information for all candidate zones. Default is `"pruned"`.

Details

Each step of the mlf scan test seeks to maximize the likelihood ratio test statistic used in the original spatial scan test (Kulldorff 1997). The first zone considered is the region that maximizes this likelihood ration test statistic, providing that no more than ubpop proportion of the total population is in

the zone. The second zone is the first zone and the connected region that maximizes the scan statistic, subject to the population and distance constraints. This pattern continues until no additional zones can be added due to population or distance constraints.

Every zone considered must have a total population less than `ubpop * sum(pop)` in the study area. Additionally, the maximum intercentroid distance for the regions within a zone must be no more than `ubd * the maximum intercentroid distance across all regions`.

Value

Returns a list that includes the location id of the zone and the associated test statistic, counts, expected counts, and population in the zone. If `type = "all"`, then each of these elements is a list or vector corresponding to each respective candidate zone.

Author(s)

Joshua French

References

Yao, Z., Tang, J., & Zhan, F. B. (2011). Detection of arbitrarily-shaped clusters using a neighbor-expanding approach: A case study on murine typhus in South Texas. International journal of health geographics, 10(1), 1.

Assuncao, R.M., Costa, M.A., Tavares, A. and Neto, S.J.F. (2006). Fast detection of arbitrarily shaped disease clusters, Statistics in Medicine, 25, 723-742.

Examples

```
data(nydf)
data(nyw)
coords = as.matrix(nydf[,c("x", "y")])
mlf.zones(coords, cases = floor(nydf$cases), pop = nydf$pop, w = nyw, 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, add = FALSE,
  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.
add	A logical indicating whether results should be drawn on existing map.
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)
```

---

plot.tango *Plots an object of class* tango.

Description

Plots results of tango.test. If Monte Carlo simulation was not used to produce x, then a a density plot of the (approximate) null distribution of tstat.chisq is produced, along with a vertical line for the observed tstat. If a Monte Carlo test was used to produce x, then a scatterplot of the gof.sim versus sa.sim is compared to the observed values gof and sa, respectively.

Usage

```
## S3 method for class 'tango'
plot(x, ..., obs.list = list(pch = 20), sim.list = list(pch
  = 2))
```

Arguments

x	An object of class tango to be plotted.
...	Additional graphical parameters passed to plot function.
obs.list	A list containing arguments for the points function, which is used to plot the gof and sa components, when appropriate.
sim.list	A list containing arguments for the points function, which is used to plot the gof.sim and sa.sim components, when appropriate.

See Also

tango.test

Examples

```
data(nydf)
coords = as.matrix(nydf[,c("x", "y")])
w = dweights(coords, kappa = 1)
x1 = tango.test(nydf$cases, nydf$pop, w)
plot(x1)
x2 = tango.test(nydf$cases, nydf$pop, w, nsim = 49)
plot(x2)
```

---

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, nsim = 499,
  alpha = 0.1, nreport = nsim + 1, ubpop = 0.5, lonlat = FALSE,
  parallel = TRUE, type = "poisson")
```

Arguments

coords	An $n \times 2$ matrix of centroid coordinates for the regions.
cases	The number of cases observed in each region.
pop	The population size associated with each region.
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 the p-value.
alpha	The significance level to determine whether a cluster is signficant. Default is `0.10`.

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	The default is FALSE, which specifies that Euclidean distance should be used.If lonlat is TRUE, then the great circle distance is used to calculate the intercentroid distance.
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.
type	The type of scan statistic to implement. Default is "poisson". Only "poisson" is currently implemented.

Details

The test is performed using the spatial scan test based on the Poisson test statistic and a fixed number of cases. Candidate zones 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 cflikely 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-overlappering 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 cluster (the largest intercentroid distance for regions in the cluster).
pop	The total population of the regions in the cluster.
cases	The observed number of cases in the cluster.
expected	The expected number of cases in the cluster.
smr	Standarized mortaility ratio (observed/expected) in the cluster.
rr	Relative risk in the cluster.
loglikrat	The loglikelihood ratio for the cluster (i.e., the log of the test statistic).
pvalue	The pvalue of the test statistic associated with the cluster.

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.

See Also

scan.stat, plot.scan, uls.test, flex.test, dmst.test, bn.test

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)
```

---

scan.zones *Determine zones for spatial scan test*

Description

scan.zones determines the unique zones to consider for the spatial scan test of Kulldorff (1997).

Usage

```
scan.zones(coords, pop, ubpop = 0.5, lonlat = FALSE)
```

Arguments

coords	An $n \times 2$ matrix of centroid coordinates for the regions.
pop	The population size associated with each region.
ubpop	The upperbound of the proportion of the total population to consider for a cluster.
lonlat	The default is FALSE, which specifies that Euclidean distance should be used.If lonlat is TRUE, then the great circle distance is used to calculate the inter-centroid distance.

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

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

Examples

```
data(nydf)
coords = cbind(nydf$longitude, nydf$latitude)
scan.zones(coords = coords, pop = nydf$pop, ubpop = 0.1, lonlat = TRUE)
```

---

tango.stat *Tango's statistic*

Description

tango.stat computes Tango's index (Tango, 1995), including both the goodness-of-fit and spatial autocorrelation components. See Waller and Gotway (2005).

Usage

```
tango.stat(cases, pop, w)
```

Arguments

cases	The number of cases observed in each region.
pop	The population size associated with each region.
w	An $n \times n$ weights matrix.

Value

Returns a list with the test statistic (`tstat`), the goodness-of-fit component (`gof`), and the spatial autocorrelation component (`sa`).

Author(s)

Joshua French

References

Tango, T. (1995) A class of tests for detecting "general" and "focused" clustering of rare diseases. Statistics in Medicine. 14:2323-2334.

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

Examples

```
data(nydf)
coords = as.matrix(nydf[,c("longitude", "latitude")])
w = dweights(coords, kappa = 1, type = "tango")
tango.stat(nydf$cases, nydf$pop, w)
```

---

tango.test *Tango's cluster detection test*

Description

`tango.test` performs a test for clustering proposed by Tango (1995). The test uses Tango's chi-square approximation for significance testing by default, but also uses Monto Carlo simulation when `nsim > 0`.

Usage

```
tango.test(cases, pop, w, nsim = 0)
```

Arguments

`cases`	The number of cases observed in each region.
`pop`	The population size associated with each region.
`w`	An $n \times n$ weights matrix.
`nsim`	The number of simulations for which to perform a Monto Carlo test of significance. Counts are simulated according to a multinomial distribution with `sum(cases)` total cases and class probabilities `pop/sum(pop)`. `sum(cases)` .

Details

The dweights function can be used to construct a weights matrix w using the method of Tango (1995), Rogerson (1999), or a basic style.

Value

Returns a list of class tango with elements:

tstat	Tango's index
tstat.chisq	The approximately chi-squared statistic proposed by Tango that is derived from tstat
dfc	The degrees of freedom of tstat.chisq
pvalue.chisq	The p-value associated with tstat.chisq
tstat.sim	The vector of test statistics from the simulated data if nsim > 0
pvalue.sim	The p-value associated with the Monte Carlo test of significance when nsim > 0

Additionally, the goodness-of-fit gof and spatial autocorrelation sa components of the Tango's index are provided (and for the simulated data sets also, if appropriate).

Author(s)

Joshua French

References

Tango, T. (1995) A class of tests for detecting "general" and "focused" clustering of rare diseases. Statistics in Medicine. 14, 2323-2334.

Rogerson, P. (1999) The Detection of Clusters Using A Spatial Version of the Chi-Square Goodness-of-fit Test. Geographical Analysis. 31, 130-147

Tango, T. (2010) Statistical Methods for Disease Clustering. Springer.

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

See Also

dweights

Examples

```
data(nydf)
coords = as.matrix(nydf[,c("x", "y")])
w = dweights(coords, kappa = 1)
results = tango.test(nydf$cases, nydf$pop, w, nsim = 49)
```

uls.test *Upper Level Set Spatial Scan Test*

Description

uls.test performs the Upper Level Set (ULS) spatial 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 observed in each region.
pop	The population size associated with each region.
w	A 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 the p-value.
alpha	The significance level to determine whether a cluster is signficant. Default is 0.10.
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	The default is FALSE, which specifies that Euclidean distance should be used.If lonlat is TRUE, then the great circle distance is used to calculate the inter-centroid distance.
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.

See Also

`scan.stat`, `plot.scan`, `scan.test`, `flex.test`, `dmst.test`, `bn.test`

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 observed in each region.
`pop`	The population size associated with each region.
`w`	A 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

bn.test, 2, *7*, *12*, *25*, *30*

casewin, 4
color.clusters, 5

dmst.test, *3*, 6, *12*, *25*, *30*
dmst.zones, 8
dweights, 9, *28*

flex.test, *3*, *7*, 11, *25*, *30*
flex.zones, 13

map, *21*
mlf.test, *3*, 14
mlf.zones, 16

nnpop, 17
nydf, 18
nypoly, 19
nyw, 20

plot.scan, *3*, *7*, *12*, 20, *25*, *30*
plot.tango, 21
points, *22*

scan.stat, *3*, *7*, *12*, 22, *25*, *30*
scan.test, *3*, *7*, *12*, 23, *30*
scan.zones, 25

tango.stat, 26
tango.test, *10*, *21*, *22*, 27

uls.test, *3*, *7*, *12*, *25*, 29
uls.zones, 31