

# Package: SpatialEpi (via r-universe)

October 10, 2024

**Type** Package

**Title** Methods and Data for Spatial Epidemiology

**Version** 1.2.8.9000

**Maintainer** Albert Y. Kim <albert.y.s.kim@gmail.com>

**Description** Methods and data for cluster detection and disease mapping.

**Depends** R (>= 3.0.2), sp

**License** GPL-2

**LazyData** true

**URL** <https://github.com/rudeboybert/SpatialEpi>

**BugReports** <https://github.com/rudeboybert/SpatialEpi/issues>

**Imports** Rcpp, MASS, spdep

**LinkingTo** Rcpp, RcppArmadillo

**NeedsCompilation** yes

**RoxygenNote** 7.2.2

**Suggests** rmarkdown, markdown, knitr, testthat (>= 3.0.0), ggplot2, dplyr

**Config/testthat/edition** 3

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**Repository** <https://epiverse-connect.r-universe.dev>

**RemoteUrl** <https://github.com/rudeboybert/SpatialEpi>

**RemoteRef** HEAD

**RemoteSha** 08e5da8bc822ec620ed3aae56a5ac08650595695

## Contents

bayes_cluster	2
besag_newell	5
circle	7
create_geo_objects	8
eBayes	9
EBpostdens	10
EBpostthresh	11
estimate_lambda	12
expected	13
GammaPriorCh	14
grid2latlong	15
kulldorff	16
latlong2grid	18
leglabs	20
LogNormalPriorCh	20
mapvariable	21
NYleukemia	22
NYleukemia_sf	23
pennLC	24
pennLC_sf	25
plotmap	26
polygon2spatial_polygon	27
process_MCMC_sample	29
scotland	30
scotland_sf	31
zones	32
<b>Index</b>	<b>33</b>

---

bayes_cluster	<i>Bayesian Cluster Detection Method</i>
---------------	--

---

## Description

Implementation of the Bayesian Cluster detection model of Wakefield and Kim (2013) for a study region with  $n$  areas. The prior and posterior probabilities of each of the  $n$  zones single zones being a cluster/anti-cluster are estimated using Markov chain Monte Carlo. Furthermore, the posterior probability of  $k$  clusters/anti-clusters is computed.

## Usage

```
bayes_cluster(
  y,
  E,
  population,
  sp.obj,
```

```

    centroids,
    max.prop,
    shape,
    rate,
    J,
    pi0,
    n.sim.lambda,
    n.sim.prior,
    n.sim.post,
    burnin.prop = 0.1,
    theta.init = vector(mode = "numeric", length = 0)
  )

```

### Arguments

<code>y</code>	vector of length <code>n</code> of the observed number of disease in each area
<code>E</code>	vector of length <code>n</code> of the expected number of disease in each area
<code>population</code>	vector of length <code>n</code> of the population in each area
<code>sp.obj</code>	an object of class <code>SpatialPolygons</code>
<code>centroids</code>	<code>n x 2</code> table of the (x,y)-coordinates of the area centroids. The coordinate system must be grid-based
<code>max.prop</code>	maximum proportion of the study region's population each single zone can contain
<code>shape</code>	vector of length 2 of narrow/wide shape parameter for gamma prior on relative risk
<code>rate</code>	vector of length 2 of narrow/wide rate parameter for gamma prior on relative risk
<code>J</code>	maximum number of clusters/anti-clusters
<code>pi0</code>	prior probability of no clusters/anti-clusters
<code>n.sim.lambda</code>	number of importance sampling iterations to estimate lambda
<code>n.sim.prior</code>	number of MCMC iterations to estimate prior probabilities associated with each single zone
<code>n.sim.post</code>	number of MCMC iterations to estimate posterior probabilities associated with each single zone
<code>burnin.prop</code>	proportion of MCMC samples to use as burn-in
<code>theta.init</code>	Initial configuration used for MCMC sampling

### Value

List containing `return(list( prior.map=prior.map, post.map=post.map, pk.y=pk.y))`

<code>prior.map</code>	A list containing, for each area: 1) <code>high.area</code> the prior probability of cluster membership, 2) <code>low.area</code> anti-cluster membership, and 3) <code>RR.est.area</code> smoothed prior estimates of relative risk
------------------------	--

post.map        A list containing, for each area: 1) high.area the posterior probability of cluster membership, 2) low.area anti-cluster membership, and 3) RR.est.area smoothed posterior estimates of the relative risk

pk.y            posterior probability of k clusters/anti-clusters given y for k=0,...,J

### Author(s)

Albert Y. Kim

### References

Wakefield J. and Kim A.Y. (2013) A Bayesian model for cluster detection.

### Examples

```
## Note for the NYleukemia example, 4 census tracts were completely surrounded
## by another unique census tract; when applying the Bayesian cluster detection
## model in [bayes_cluster()], we merge them with the surrounding
## census tracts yielding `n=277` areas.

## Load data and convert coordinate system from latitude/longitude to grid
data(NYleukemia)
sp.obj <- NYleukemia$spatial.polygon
population <- NYleukemia$data$population
cases <- NYleukemia$data$cases
centroids <- latlong2grid(NYleukemia$geo[, 2:3])

## Identify the 4 census tract to be merged into their surrounding census tracts
remove <- NYleukemia$surrounded
add <- NYleukemia$surrounding

## Merge population and case counts and geographical objects accordingly
population[add] <- population[add] + population[remove]
population <- population[-remove]
cases[add] <- cases[add] + cases[remove]
cases <- cases[-remove]
sp.obj <-
  SpatialPolygons(sp.obj@polygons[-remove], proj4string=CRS("+proj=longlat +ellps=WGS84"))
centroids <- centroids[-remove, ]

## Set parameters
y <- cases
E <- expected(population, cases, 1)
max.prop <- 0.15
shape <- c(2976.3, 2.31)
rate <- c(2977.3, 1.31)
J <- 7
pi0 <- 0.95
n.sim.lambda <- 10^4
n.sim.prior <- 10^5
n.sim.post <- 10^5
```

```
## (Uncomment first) Compute output
#output <- bayes_cluster(y, E, population, sp.obj, centroids, max.prop,
# shape, rate, J, pi0, n.sim.lambda, n.sim.prior, n.sim.post)
#plotmap(output$prior.map$high.area, sp.obj)
#plotmap(output$post.map$high.area, sp.obj)
#plotmap(output$post.map$RR.est.area, sp.obj, log=TRUE)
#barplot(output$pk.y, names.arg=0:J, xlab="k", ylab="P(k|y)")
```

---

besag\_newell

*Besag-Newell Cluster Detection Method*


---

## Description

Besag-Newell cluster detection method. There are differences with the original paper and our implementation:

- we base our analysis on  $k$  cases, rather than  $k$  other cases as prescribed in the paper.
- we do not subtract 1 from the *accumulated numbers of other cases* and *accumulated numbers of others at risk*, as was prescribed in the paper to discount selection bias
- $M$  is the total number of areas included, not the number of additional areas included. i.e.  $M$  starts at 1, not 0.
- $p$ -values are not based on the original value of  $k$ , rather the actual number of cases observed until we view  $k$  or more cases. Ex: if  $k = 10$ , but as we consider neighbors we encounter 1, 2, 9 then 12 cases, we base our  $p$ -values on  $k = 12$
- we do not provide a Monte-Carlo simulated  $R$ : the number of tests that attain significance at a fixed level  $\alpha$

The first two and last differences are because we view the testing on an area-by-area level, rather than a case-by-case level.

## Usage

```
besag_newell(geo, population, cases, expected.cases = NULL, k, alpha.level)
```

## Arguments

geo	an $n \times 2$ table of the (x,y)-coordinates of the area centroids
population	aggregated population counts for all $n$ areas
cases	aggregated case counts for all $n$ areas
expected.cases	expected numbers of disease for all $n$ areas
k	number of cases to consider
alpha.level	alpha-level threshold used to declare significance

**Details**

For the population and cases tables, the rows are bunched by areas first, and then for each area, the counts for each strata are listed. It is important that the tables are balanced: the strata information are in the same order for each area, and counts for each area/strata combination appear exactly once (even if zero).

**Value**

List containing

clusters	information on all clusters that are $\alpha$ -level significant, in decreasing order of the $p$ -value
p.values	for each of the $n$ areas, $p$ -values of each cluster of size at least $k$
m.values	for each of the $n$ areas, the number of areas need to observe at least $k$ cases
observed.k.values	based on m.values, the actual number of cases used to compute the $p$ -values

**Note**

The clusters list elements are themselves lists reporting:

location.IDs.included	ID's of areas in cluster, in order of distance
population	population of cluster
number.of.cases	number of cases in cluster
expected.cases	expected number of cases in cluster
SMR	estimated SMR of cluster
p.value	$p$ -value

**Author(s)**

Albert Y. Kim

**References**

Besag J. and Newell J. (1991) The Detection of Clusters in Rare Diseases *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, **154**, 143–155

**Examples**

```
## Load Pennsylvania Lung Cancer Data
data(pennLC)
data <- pennLC$data

## Process geographical information and convert to grid
geo <- pennLC$geo[,2:3]
geo <- latlong2grid(geo)
```

```
## Get aggregated counts of population and cases for each county
population <- tapply(data$population,data$county,sum)
cases <- tapply(data$cases,data$county,sum)

## Based on the 16 strata levels, computed expected numbers of disease
n.strata <- 16
expected.cases <- expected(data$population, data$cases, n.strata)

## Set Parameters
k <- 1250
alpha.level <- 0.05

# not controlling for stratas
results <- besag_newell(geo, population, cases, expected.cases=NULL, k,
                        alpha.level)

# controlling for stratas
results <- besag_newell(geo, population, cases, expected.cases, k, alpha.level)
```

---

circle

*Compute cartesian coordinates of a cluster center and radius*

---

### Description

This function is used for plotting purposes

### Usage

```
circle(geo, cluster.center, cluster.end)
```

### Arguments

geo	A $n \times 2$ table of the x-coordinate and y-coordinates of the centroids of each area
cluster.center	The area index (an integer between 1 and $n$ ) indicating the center of the circle
cluster.end	The area index (an integer between 1 and $n$ ) indicating the area at the end of the circle

### Value

cluster.radius A data frame that you can plot

### Author(s)

Albert Y. Kim

**Examples**

```
data(pennLC)
geo <- pennLC$geo[,2:3]
plot(geo,type='n')
text(geo,labels=1:nrow(geo))
lines( circle(geo, 23, 46), col = "red" )
```

---

create_geo_objects	<i>Create geographical objects to be used in Bayesian Cluster Detection Method</i>
--------------------	--

---

**Description**

This internal function creates the geographical objects needed to run the Bayesian cluster detection method in `bayes_cluster()`. Specifically it creates all single zones based data objects, where single zones are the *zones* defined by Kulldorff (1997).

**Usage**

```
create_geo_objects(max.prop, population, centroids, sp.obj)
```

**Arguments**

max.prop	maximum proportion of study region's population each single zone can contain
population	vector of length n of the population of each area
centroids	n x 2 table of the (x,y)-coordinates of the area centroids. The coordinate system must be grid-based
sp.obj	object of class <code>SpatialPolygons</code> (See <a href="#">SpatialPolygons-class</a> ) representing the study region

**Value**

overlap	list with two elements: 1. <code>presence</code> which lists for each area all the single zones it is present in and 2. <code>cluster.list</code> for each single zone its component areas
cluster.coords	n.zones x 2 matrix of the center and radial area of each single zone

**Author(s)**

Albert Y. Kim

**References**

Wakefield J. and Kim A.Y. (2013) A Bayesian model for cluster detection. *Biostatistics*, **14**, 752–765.



**Examples**

```

data(pennLC)
max.prop <- 0.15
population <- tapply(pennLC$data$population, pennLC$data$county, sum)
centroids <- latlong2grid(pennLC$geo[, 2:3])
sp.obj <- pennLC$spatial.polygon
output <- create_geo_objects(max.prop, population, centroids, sp.obj)
## number of single zones
nrow(output$cluster.coords)

```

eBayes

*Empirical Bayes Estimates of Relative Risk***Description**

The computes empirical Bayes estimates of relative risk of study region with  $n$  areas, given observed and expected numbers of counts of disease and covariate information.

**Usage**

```
eBayes(Y, E, Xmat = NULL)
```

**Arguments**

Y	a length $n$ vector of observed cases
E	a length $n$ vector of expected number of cases
Xmat	$n \times p$ dimension matrix of covariates

**Value**

A list with 5 elements:

RR	the ecological relative risk posterior mean estimates
RRmed	the ecological relative risk posterior median estimates
beta	the MLE's of the regression coefficients
alpha	the MLE of negative binomial dispersion parameter
SMR	the standardized mortality/morbidity ratio $Y/E$

**References**

Clayton D. and Kaldor J. (1987) Empirical Bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics*, **43**, 671–681

**Examples**

```

data(scotland)
data <- scotland$data
x <- data$AFF
Xmat <- cbind(x,x^2)
results <- eBayes(data$cases,data$expected,Xmat)
scotland.map <- scotland$spatial.polygon
mapvariable(results$RR, scotland.map)

```

---

EBpostdens

*Produce plots of empirical Bayes posterior densities when the data  $Y$  are Poisson with expected number  $E$  and relative risk  $\theta$ , with the latter having a gamma distribution with known values  $\alpha$  and  $\beta$ , which are estimated using empirical Bayes.*

---

**Description**

This function produces plots of empirical Bayes posterior densities which are gamma distributions with parameters  $(\alpha+Y, (\alpha+E*\mu)/\mu)$  where  $\mu = \exp(x \beta)$ . The SMRs are drawn on for comparison.

**Usage**

```

EBpostdens(
  Y,
  E,
  alpha,
  beta,
  Xrow = NULL,
  lower = NULL,
  upper = NULL,
  main = ""
)

```

**Arguments**

Y	observed disease counts
E	expected disease counts
alpha	x
beta	x
Xrow	x
lower	x
upper	x
main	x

**Value**

A plot containing the gamma posterior distribution

**Author(s)**

Jon Wakefield

**Examples**

```
data(scotland)
Y <- scotland$data$cases
E <- scotland$data$expected
ebresults <- eBayes(Y,E)
EBpostdens(Y[1], E[1], ebresults$alpha, ebresults$beta, lower=0, upper=15,
            main="Area 1")
```

---

EBpostthresh	<i>Produce the probabilities of exceeding a threshold given a posterior gamma distribution.</i>
--------------	---

---

**Description**

This function produces the posterior probabilities of exceeding a threshold given a gamma distributions with parameters  $(\alpha+Y, (\alpha+E*\mu)/\mu)$  where  $\mu = \exp(x \text{ beta})$ . This model arises from Y being Poisson with mean  $\theta$  times E where  $\theta$  is the relative risk and E are the expected numbers. The prior on  $\theta$  is gamma with parameters  $\alpha$  and  $\beta$ . The parameters  $\alpha$  and  $\beta$  may be estimated using empirical Bayes.

**Usage**

```
EBpostthresh(Y, E, alpha, beta, Xrow = NULL, rrthresh)
```

**Arguments**

Y	observed disease counts
E	expected disease counts
alpha	x
beta	x
Xrow	x
rrthresh	x

**Value**

Posterior probabilities of exceedence are returned.

**Author(s)**

Jon Wakefield

**See Also**[eBayes\(\)](#)**Examples**

```

data(scotland)
Y <- scotland$data$cases
E <- scotland$data$expected
ebresults <- eBayes(Y,E)
#Find probabilities of exceedence of 3
thresh3 <- EBpostthresh(Y, E, alpha=ebresults$alpha, beta=ebresults$beta, rrthresh=3)
mapvariable(thresh3, scotland$spatial.polygon)

```

estimate\_lambda

*Estimate lambda values***Description**

Internal function to estimate values of lambda needed for MCMC\_simulation and prior probability of k clusters/anti-clusters for k=0,...,J

**Usage**

```
estimate_lambda(n.sim, J, prior.z, overlap, pi0)
```

**Arguments**

n.sim	number of importance sampling iterations
J	maximum number of clusters/anti-clusters to consider
prior.z	prior probability of each single zone
overlap	output of <a href="#">create_geo_objects()</a> : list with two elements: presence which lists for each area all the single zones it is present in and cluster_list for each single zone its component areas
pi0	prior probability of no clusters

**Value**

estimates of lambda and prior.j

**References**

Wakefield J. and Kim A.Y. (2013) A Bayesian model for cluster detection. *Biostatistics*, **14**, 752–765.

---

expected	<i>Compute Expected Numbers of Disease</i>
----------	--

---

**Description**

Compute the internally indirect standardized expected numbers of disease.

**Usage**

```
expected(population, cases, n.strata)
```

**Arguments**

population	a vector of population counts for each strata in each area
cases	a vector of the corresponding number of cases
n.strata	number of strata considered

**Details**

The population and cases vectors must be *balanced*: all counts are sorted by area first, and then within each area the counts for all strata are listed (even if 0 count) in the same order.

**Value**

expected.cases a vector of the expected numbers of disease for each area

**Author(s)**

Albert Y. Kim

**References**

Elliot, P. et al. (2000) *Spatial Epidemiology: Methods and Applications*. Oxford Medical Publications.

**Examples**

```
data(pennLC)
population <- pennLC$data$population
cases <- pennLC$data$cases
## In each county in Pennsylvania, there are 2 races, gender and 4 age bands
## considered = 16 strata levels
pennLC$data[1:16,]
expected(population, cases, 16)
```

---

`GammaPriorCh`*Compute Parameters to Calibrate a Gamma Distribution*

---

**Description**

Compute parameters to calibrate the prior distribution of a relative risk that has a gamma distribution.

**Usage**

```
GammaPriorCh(theta, prob, d)
```

**Arguments**

<code>theta</code>	upper quantile
<code>prob</code>	upper quantile
<code>d</code>	degrees of freedom

**Value**

List containing

<code>a</code>	shape parameter
<code>b</code>	rate parameter

**Author(s)**

Jon Wakefield

**See Also**

`LogNormalPriorCh`

**Examples**

```
param <- GammaPriorCh(5, 0.975, 1)
curve(dgamma(x, shape=param$a, rate=param$b), from=0, to=6, n=1000, ylab="density")
```

---

`grid2latlong`*Convert Coordinates from Grid to Latitude/Longitude*

---

**Description**

Convert geographic coordinates from Universal Transverse Mercator system to Latitude/Longitude.

**Usage**

```
grid2latlong(input)
```

**Arguments**

<code>input</code>	A data frame with columns named <code>x</code> and <code>y</code> of the UTM coordinates to convert or an $n \times 2$ matrix of grid coordinates or an object of class <code>SpatialPolygons</code> (See <a href="#">SpatialPolygons-class</a> )
--------------------	---

**Details**

Longitude/latitudes are not a grid-based coordinate system: latitudes are equidistant but the distance between longitudes varies.

**Value**

Either a data frame with the corresponding longitude and latitude, or a `SpatialPolygons` object with the coordinates changed.

**Note**

Rough conversion of US lat/long to km (used by GeoBUGS): (see also [forum.swarthmore.edu/dr.math/problems/longandlat.h](http://forum.swarthmore.edu/dr.math/problems/longandlat.h))  
Radius of earth:  $r = 3963.34$  (equatorial) or  $3949.99$  (polar) mi = 6378.2 or 6356.7 km, which implies: km per mile = 1.609299 or 1.609295 a change of 1 degree of latitude corresponds to the same number of km, regardless of longitude.  $arclength=r\theta$ , so the multiplier for coord `y` should probably be just the radius of earth. On the other hand, a change of 1 degree in longitude corresponds to a different distance, depending on latitude. (at N pole, the change is essentially 0. at the equator, use equatorial radius. Perhaps for U.S., might use an "average" latitude, 30 deg is roughly Houston, 49deg is most of N bdry of continental 48 states.  $0.5(30+49)=39.5$  deg. so use  $r$  approx  $6378.2\sin(51.5)$ )

**Author(s)**

Lance A. Waller

**Examples**

```

coord <- data.frame(rbind(
# Montreal, QC
c(-6414.30, 5052.849),
# Vancouver, BC
c(-122.6042, 45.6605)
))

grid2latlong(coord)

```

kulldorff

*Kulldorff Cluster Detection Method***Description**

Kulldorff spatial cluster detection method for a study region with  $n$  areas. The method constructs *zones* by consecutively aggregating nearest-neighboring areas until a proportion of the total study population is included. Given the observed number of cases, the likelihood of each zone is computed using either binomial or poisson likelihoods. The procedure reports the zone that is the *most likely cluster* and generates significance measures via Monte Carlo sampling. Further, *secondary clusters*, whose Monte Carlo p-values are below the  $\alpha$ -threshold, are reported as well.

**Usage**

```

kulldorff(
  geo,
  cases,
  population,
  expected.cases = NULL,
  pop.upper.bound,
  n.simulations,
  alpha.level,
  plot = TRUE
)

```

**Arguments**

geo	an $n \times 2$ table of the (x,y)-coordinates of the area centroids
cases	aggregated case counts for all $n$ areas
population	aggregated population counts for all $n$ areas
expected.cases	expected numbers of disease for all $n$ areas
pop.upper.bound	the upper bound on the proportion of the total population each zone can include
n.simulations	number of Monte Carlo samples used for significance measures
alpha.level	alpha-level threshold used to declare significance
plot	flag for whether to plot histogram of Monte Carlo samples of the log-likelihood of the most likely cluster



**Details**

If `expected.cases` is specified to be `NULL`, then the binomial likelihood is used. Otherwise, a Poisson model is assumed. Typical values of `n.simulations` are 99, 999, 9999

**Value**

List containing:

<code>most.likely.cluster</code>	information on the most likely cluster
<code>secondary.clusters</code>	information on secondary clusters, if none <code>NULL</code> is returned
<code>type</code>	type of likelihood
<code>log.lkhd</code>	log-likelihood of each zone considered
<code>simulated.log.lkhd</code>	<code>n.simulations</code> Monte Carlo samples of the log-likelihood of the most likely cluster

**Note**

The `most.likely.cluster` and `secondary.clusters` list elements are themselves lists reporting:

<code>location.IDs.included</code>	ID's of areas in cluster, in order of distance
<code>population</code>	population of cluster
<code>number.of.cases</code>	number of cases in cluster
<code>expected.cases</code>	expected number of cases in cluster
<code>SMR</code>	estimated SMR of cluster
<code>log.likelihood.ratio</code>	log-likelihood of cluster
<code>monte.carlo.rank</code>	rank of <code>lkhd</code> of cluster within Monte Carlo simulated values
<code>p.value</code>	Monte Carlo $p$ -value

**Author(s)**

Albert Y. Kim

**References**

SatScan: Software for the spatial, temporal, and space-time scan statistics <https://www.satscan.org/> Kulldorff, M. (1997) A spatial scan statistic. *Communications in Statistics: Theory and Methods*, **26**, 1481–1496. Kulldorff M. and Nagarwalla N. (1995) Spatial disease clusters: Detection and Inference. *Statistics in Medicine*, **14**, 799–810.

**Examples**

```

## Load Pennsylvania Lung Cancer Data
data(pennLC)
data <- pennLC$data

## Process geographical information and convert to grid
geo <- pennLC$geo[,2:3]
geo <- latlong2grid(geo)

## Get aggregated counts of population and cases for each county
population <- tapply(data$population,data$county,sum)
cases <- tapply(data$cases,data$county,sum)

## Based on the 16 strata levels, computed expected numbers of disease
n.strata <- 16
expected.cases <- expected(data$population, data$cases, n.strata)

## Set Parameters
pop.upper.bound <- 0.5
n.simulations <- 999
alpha.level <- 0.05
plot <- TRUE

## Kulldorff using Binomial likelihoods
binomial <- kulldorff(geo, cases, population, NULL, pop.upper.bound, n.simulations,
                    alpha.level, plot)
cluster <- binomial$most.likely.cluster$location.IDs.included

## plot
plot(pennLC$spatial.polygon,axes=TRUE)
plot(pennLC$spatial.polygon[cluster],add=TRUE,col="red")
title("Most Likely Cluster")

## Kulldorff using Poisson likelihoods
poisson <- kulldorff(geo, cases, population, expected.cases, pop.upper.bound,
                    n.simulations, alpha.level, plot)
cluster <- poisson$most.likely.cluster$location.IDs.included

## plot
plot(pennLC$spatial.polygon,axes=TRUE)
plot(pennLC$spatial.polygon[cluster],add=TRUE,col="red")
title("Most Likely Cluster Controlling for Strata")

```

---

latlong2grid

---

*Convert Coordinates from Latitude/Longitude to Grid*


---

**Description**

Convert geographic latitude/longitude coordinates to kilometer-based grid coordinates.

**Usage**

```
latlong2grid(input)
```

**Arguments**

input                    either an  $n \times 2$  matrix of longitude and latitude coordinates in decimal format or an object of class SpatialPolygons

**Details**

Longitude/latitudes are not a grid-based coordinate system: latitudes are equidistant but the distance between longitudes varies.

**Value**

Either a data frame with the corresponding (x,y) kilometer-based grid coordinates, or a SpatialPolygons object with the coordinates changed.

**Note**

Rough conversion of US lat/long to km (used by GeoBUGS): (see also [forum.swarthmore.edu/dr.math/problems/longandlat.html](http://forum.swarthmore.edu/dr.math/problems/longandlat.html))  
 Radius of earth:  $r = 3963.34$  (equatorial) or  $3949.99$  (polar) mi = 6378.2 or 6356.7 km, which implies: km per mile = 1.609299 or 1.609295 a change of 1 degree of latitude corresponds to the same number of km, regardless of longitude.  $\text{arclength} = r * \theta$ , so the multiplier for coord y should probably be just the radius of earth. On the other hand, a change of 1 degree in longitude corresponds to a different distance, depending on latitude. (at N pole, the change is essentially 0. at the equator, use equatorial radius.

**Author(s)**

Lance A. Waller

**Examples**

```
## Convert coordinates
coord <- data.frame(rbind(
  # Montreal, QC: Latitude: 45deg 28' 0" N (deg min sec), Longitude: 73deg 45' 0" W
  c(-73.7500, 45.4667),
  # Vancouver, BC: Latitude: 45deg 39' 38" N (deg min sec), Longitude: 122deg 36' 15" W
  c(-122.6042, 45.6605)
))
latlong2grid(coord)
## Convert SpatialPolygon
data(pennLC)
new <- latlong2grid(pennLC$spatial.polygon)
par(mfrow=c(1,2))
plot(pennLC$spatial.polygon, axes=TRUE)
title("Lat/Long")
plot(new, axes=TRUE)
title("Grid (in km)")
```

---

leglabs	<i>Make legend labels</i>
---------	---------------------------

---

**Description**

leglabs makes character strings from the same break points. This function was copied from the soon-to-be deprecated maptools package with permission from author Roger Bivand

**Usage**

```
leglabs(vec, under = "under", over = "over", between = "-", reverse = FALSE)
```

**Arguments**

vec	vector of break values
under	character value for under
over	character value for over
between	character value for between
reverse	flag to reverse order of values, you will also need to reorder colours, see example

**Author(s)**

Roger Bivand, Nick Bearman, Nicholas Lewin-Koh

---

LogNormalPriorCh	<i>Compute Parameters to Calibrate a Log-normal Distribution</i>
------------------	--

---

**Description**

Compute parameters to calibrate the prior distribution of a relative risk that has a log-normal distribu

**Usage**

```
LogNormalPriorCh(theta1, theta2, prob1, prob2)
```

**Arguments**

theta1	lower quantile
theta2	upper quantile
prob1	lower probability
prob2	upper probability

**Value**

A list containing

mu                    mean of log-normal distribution  
sigma                 variance of log-normal distribution

**Author(s)**

Jon Wakefield

**Examples**

```
# Calibrate the log-normal distribution s.t. the 95% confidence interval is [0.2, 5]
param <- LogNormalPriorCh(0.2, 5, 0.025, 0.975)
curve(dlnorm(x,param$mu,param$sigma), from=0, to=6, ylab="density")
```

---

mapvariable

*Plot Levels of a Variable in a Colour-Coded Map*

---

**Description**

Plot levels of a variable in a colour-coded map along with a legend.

**Usage**

```
mapvariable(
  y,
  spatial.polygon,
  ncut = 1000,
  nlevels = 10,
  lower = NULL,
  upper = NULL,
  main = NULL,
  xlab = NULL,
  ylab = NULL
)
```

**Arguments**

y                    variable to plot  
spatial.polygon    an object of class SpatialPolygons (See [SpatialPolygons-class](#))  
ncut                 number of cuts in colour levels to plot  
nlevels             number of levels to include in legend  
lower                lower bound of levels  
upper                upper bound of levels

main            an overall title for the plot  
 xlab            a title for the x axis  
 ylab            a title for the y axis

### Value

A map colour-coded to indicate the different levels of y

### Author(s)

Jon Wakefield, Nicky Best, Sebastien Haneuse, and Albert Y. Kim

### References

Bivand, R. S., Pebesma E. J., and Gomez-Rubio V. (2008) *Applied Spatial Data Analysis with R*. Springer Series in Statistics. E. J. Pebesma and R. S. Bivand. (2005) Classes and methods for spatial data in *R. R News*, **5**, 9–13.

### Examples

```
data(scotland)
map <- scotland$spatial.polygon
y <- scotland$data$cases
E <- scotland$data$expected
SMR <- y/E
mapvariable(SMR, map, main="Scotland", xlab="Eastings (km)", ylab="Northings (km)")
```

---

NYleukemia

*Upstate New York Leukemia Data*

---

### Description

Census tract level (n=281) leukemia data for the 8 counties in upstate New York from 1978-1982, paired with population data from the 1980 census. Note that 4 census tracts were completely surrounded by another unique census tract; when applying the Bayesian cluster detection model in [bayes\\_cluster\(\)](#), we merge them with the surrounding census tracts yielding n=277 areas.

### Usage

NYleukemia

### Format

List with 5 items:

**geo** table of the FIPS code, longitude, and latitude of the geographic centroid of each census tract

**data** table of the FIPS code, number of cases, and population of each census tract

**spatial.polygon** object of class SpatialPolygons

**surrounded** row IDs of the 4 census tracts that are completely surrounded by the

**surrounding** census tracts

## References

Turnbull, B. W. et al (1990) Monitoring for clusters of disease: application to leukemia incidence in upstate New York *American Journal of Epidemiology*, **132**, 136–143

## Examples

```
## Load data and convert coordinate system from latitude/longitude to grid
data(NYleukemia)
map <- NYleukemia$spatial.polygon
population <- NYleukemia$data$population
cases <- NYleukemia$data$cases
centroids <- latlong2grid(NYleukemia$geo[, 2:3])

## Identify the 4 census tract to be merged into their surrounding census tracts.
remove <- NYleukemia$surrounded
add <- NYleukemia$surrounding

## Merge population and case counts
population[add] <- population[add] + population[remove]
population <- population[-remove]
cases[add] <- cases[add] + cases[remove]
cases <- cases[-remove]

## Modify geographical objects accordingly
map <- SpatialPolygons(map@polygons[-remove], proj4string=CRS("+proj=longlat +ellps=WGS84"))
centroids <- centroids[-remove, ]

## Plot incidence in latitude/longitude
plotmap(cases/population, map, log=TRUE, nclr=5)
points(grid2latlong(centroids), pch=4)
```

---

NYleukemia\_sf

*Upstate New York Leukemia*

---

## Description

Census tract level (n=281) leukemia data for the 8 counties in upstate New York from 1978-1982, paired with population data from the 1980 census. Note that 4 census tracts were completely surrounded by another unique census tract; when applying the Bayesian cluster detection model in [bayes\\_cluster\(\)](#), we merge them with the surrounding census tracts yielding n=277 areas.

## Usage

NYleukemia\_sf

**Format**

An sf 'POLYGON' data frame with 281 rows and 4 variables:

**geometry** Geometric representation of 8 counties in upstate New York

**cases** Number of cases per county

**population** Population of each census tract

**censtract.FIPS** 11-digit Federal Information Processing System identification number for each county

**Source**

Turnbull, B. W. et al (1990) Monitoring for clusters of disease: application to leukemia incidence in upstate New York *American Journal of Epidemiology*, **132**, 136–143

**Examples**

```
# Static map of NY Leukemia rate per county
library(ggplot2)
## Not run:
ggplot(NYleukemia_sf) +
  geom_sf(aes(fill= cases/population)) +
  scale_fill_gradient(low = "white", high = "red")

## End(Not run)
```

---

pennLC

*Pennsylvania Lung Cancer*

---

**Description**

County-level (n=67) population/case data for lung cancer in Pennsylvania in 2002, stratified on race (white vs non-white), gender and age (Under 40, 40-59, 60-69 and 70+). Additionally, county-specific smoking rates.

**Usage**

pennLC

**Format**

List of 3 items

**geo** a table of county IDs, longitude/latitude of the geographic centroid of each county

**data** a table of county IDs, number of cases, population and strata information

**smoking** a table of county IDs and proportion of smokers

**spatial.polygon** an object of class SpatialPolygons



**Source**

Population data was obtained from the 2000 decennial census, lung cancer and smoking data were obtained from the Pennsylvania Department of Health website: <https://www.health.pa.gov/Pages/default.aspx>

**Examples**

```
data(pennLC)
pennLC$geo
pennLC$data
pennLC$smoking
# Map smoking rates in Pennsylvania
mapvariable(pennLC$smoking[,2], pennLC$spatial.polygon)
```

---

pennLC\_sf

*Pennsylvania Lung Cancer*


---

**Description**

County-level (n=67) population/case data for lung cancer in Pennsylvania in 2002, stratified on race (white vs non-white), gender and age (Under 40, 40-59, 60-69 and 70+). Additionally, county-specific smoking rates.

**Usage**

```
pennLC_sf
```

**Format**

An sf POLYGON data frame with 1072 rows = 67 counties x 2 race x 2 gender x 4 age bands

**county** Pennsylvania county

**cases** Number of cases per county split by strata

**population** Population per county split by strata

**race** Race (w = white and o = non-white)

**gender** Gender (f = female and m = male)

**age** Age (4 bands)

**smoking** Overall county smoking rate (not broken down by strata)

**geometry** Geometric representation of counties in Pennsylvania

## Source

Population data was obtained from the 2000 decennial census, lung cancer and smoking data were obtained from the Pennsylvania Department of Health website:<https://www.health.pa.gov/Pages/default.aspx>.

## Examples

```
library(ggplot2)
library(dplyr)
# Sum cases & population for each county
lung_cancer_rate <- pennLC_sf %>%
  group_by(county) %>%
  summarize(cases = sum(cases), population = sum(population)) %>%
  mutate(rate = cases/population)

# Static map of Pennsylvania lung cancer rates for each county
## Not run:
ggplot() +
  geom_sf(data = lung_cancer_rate, aes(fill = rate))

## End(Not run)
```

---

plotmap

*Plot Levels of a Variable in a Colour-Coded Map*

---

## Description

Plot levels of a variable in a colour-coded map.

## Usage

```
plotmap(
  values,
  map,
  log = FALSE,
  nclr = 7,
  include.legend = TRUE,
  lwd = 0.5,
  round = 3,
  brks = NULL,
  legend = NULL,
  location = "topright",
  rev = FALSE
)
```

**Arguments**

values	variable to plot
map	an object of class SpatialPolygons (See <a href="#">SpatialPolygons-class</a> )
log	boolean of whether to plot values on log scale
nclr	number of colour-levels to use
include.legend	boolean of whether to include legend
lwd	line width of borders of areas
round	number of digits to round to in legend
brks	if desired, pre-specified breaks for legend
legend	if desired, a pre-specified legend
location	location of legend
rev	boolean of whether to reverse colour scheme (darker colours for smaller values)

**Value**

A map colour-coded to indicate the different levels of values.

**Author(s)**

Albert Y. Kim

**Examples**

```
## Load data
data(scotland)
map <- scotland$spatial.polygon
y <- scotland$data$cases
E <- scotland$data$expected
SMR <- y/E
## Plot SMR
plotmap(SMR, map, nclr=9, location="topleft")
```

---

`polygon2spatial_polygon`

*Convert a Polygon to a Spatial Polygons Object*

---

**Description**

Converts a polygon (a matrix of coordinates with NA values to separate subpolygons) into a Spatial Polygons object.

**Usage**

```

polygon2spatial_polygon(
  poly,
  coordinate.system,
  area.names = NULL,
  nrepeats = NULL
)

```

**Arguments**

poly	a 2-column matrix of coordinates, where each complete subpolygon is separated by NA's
coordinate.system	the coordinate system to use
area.names	names of all areas
nrepeats	number of sub polygons for each area

**Details**

Just as when plotting with the `graphics::polygon()` function, it is assumed that each subpolygon is to be closed by joining the last point to the first point. In the matrix `poly`, NA values separate complete subpolygons. In the case with an area consists of more than one separate closed polygon, `nrepeats` specifies the number of closed polygons associated with each area.

**Value**

An object of class `SpatialPolygons` (See [SpatialPolygons-class](#) from the `sp` package).

**Author(s)**

Albert Y. Kim

**References**

Bivand, R. S., Pebesma E. J., and Gomez-Rubio V. (2008) *Applied Spatial Data Analysis with R*. Springer Series in Statistics. E. J. Pebesma and R. S. Bivand. (2005) Classes and methods for spatial data in *R. R News*, **5**, 9–13.

**Examples**

```

data(scotland)

polygon <- scotland$polygon$polygon
coord.system <- "+proj=eqc +lat_ts=0 +lat_0=0 +lon_0=0 +x_0=0 +y_0=0 "
coord.system <- paste(coord.system, "+ellps=WGS84 +datum=WGS84 +units=m +no_defs", sep = "")
names <- scotland$data$county.names
nrepeats <- scotland$polygon$nrepeats

spatial.polygon <- polygon2spatial_polygon(polygon, coord.system, names, nrepeats)

```

```

par(mfrow=c(1,2))
# plot using polygon function
plot(polygon,type='n',xlab="Eastings (km)",ylab="Northings (km)",main="Polygon File")
polygon(polygon)

# plot as spatial polygon object
plot(spatial.polygon,axes=TRUE)
title(xlab="Eastings (km)",ylab="Northings (km)",main="Spatial Polygon")

# Note that area 23 (argyll-bute) consists of 8 separate polygons
nrepeats[23]
plot(spatial.polygon[23],add=TRUE,col="red")

```

---

process\_MCMC\_sample    *Process MCMC Sample*

---

### Description

Take the output of sampled configurations from `MCMC_simulation` and produce area-by-area summaries

### Usage

```
process_MCMC_sample(sample, param, RR.area, cluster.list, cutoffs)
```

### Arguments

<code>sample</code>	list objects of sampled configurations
<code>param</code>	mean relative risk associated with each of the <code>n.zones</code> single zones considering the wide prior
<code>RR.area</code>	mean relative risk associated with each of the <code>n</code> areas considering the narrow prior
<code>cluster.list</code>	list of length <code>n.zones</code> listing, for each single zone, its component areas
<code>cutoffs</code>	cutoffs used to declare highs (clusters) and lows (anti-clusters)

### Value

<code>high.area</code>	Probability of cluster membership for each area
<code>low.area</code>	Probability of anti-cluster membership for each area
<code>RR.est.area</code>	Smoothed relative risk estimates for each area

### References

Wakefield J. and Kim A.Y. (2013) A Bayesian model for cluster detection. *Biostatistics*, **14**, 752–765.

---

`scotland`*Lip Cancer in Scotland*

---

**Description**

County-level (n=56) data for lip cancer among males in Scotland between 1975-1980

**Usage**

```
scotland
```

**Format**

List containing:

**geo** a table of county IDs, x-coordinates (eastings) and y-coordinates (northings) of the geographic centroid of each county.

**data** a table of county IDs, number of cases, population and strata information

**spatial.polygon** a Spatial Polygons class (See [SpatialPolygons-class](#)) map of Scotland

**polygon** a polygon map of Scotland (See [polygon2spatial\\_polygon\(\)](#))

**Source**

Kemp I., Boyle P., Smans M. and Muir C. (1985) Atlas of cancer in Scotland, 1975-1980, incidence and epidemiologic perspective *International Agency for Research on Cancer* **72**.

**References**

Clayton D. and Kaldor J. (1987) Empirical Bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics*, **43**, 671–681.

**Examples**

```
data(scotland)
data <- scotland$data
scotland.map <- scotland$spatial.polygon
SMR <- data$cases/data$expected
mapvariable(SMR,scotland.map)
```

---

`scotland_sf`*Lip Cancer in Scotland*

---

### Description

County-level (n=56) data for lip cancer among males in Scotland between 1975-1980

### Usage

```
scotland_sf
```

### Format

A data frame with 56 rows representing counties and 5 variables:

**geometry** Geometric representation of counties in Scotland

**cases** Number of Lip Cancer cases per county

**county.names** Scotland County name

**AFF** Proportion of the population who work in agricultural fishing and farming

**expected** Expected number of lip cancer cases

### Source

Kemp I., Boyle P., Smans M. and Muir C. (1985) Atlas of cancer in Scotland, 1975-1980, incidence and epidemiologic perspective *International Agency for Research on Cancer* **72**.

### References

Clayton D. and Kaldor J. (1987) Empirical Bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics*, **43**, 671–681.

### Examples

```
library(ggplot2)
## Not run:
ggplot() +
  geom_sf(data = scotland_sf, aes(fill= cases))

## End(Not run)
```

---

zones *Create set of all single zones and output geographical information*

---

### Description

Based on the population counts and centroid coordinates of each of  $n$  areas, output the set of  $n$ .zones single zones as defined by Kulldorff and other geographical information.

### Usage

```
zones(geo, population, pop.upper.bound)
```

### Arguments

geo	$n \times 2$ table of the (x,y)-coordinates of the area centroids
population	a vector of population counts of each area
pop.upper.bound	maximum proportion of study region each zone can contain

### Value

A list containing

nearest.neighbors	list of $n$ elements, where each element is a vector of the nearest neighbors in order of distance up until pop.upper.bound of the total population is attained
cluster.coords	$n$ .zones $\times$ 2 table of the center and the radial area for each zone
dist	$n \times n$ inter-point distance matrix of the centroids

### Author(s)

Albert Y. Kim

### References

Kulldorff, M. (1997) A spatial scan statistic. *Communications in Statistics: Theory and Methods*, **26**, 1481–1496. Kulldorff M. and Nagarwalla N. (1995) Spatial disease clusters: Detection and Inference. *Statistics in Medicine*, **14**, 799–810.

### Examples

```
data(pennLC)
geo <- pennLC$geo[,2:3]
geo <- latlong2grid(geo)
population <- tapply(pennLC$data$population, pennLC$data$county, sum)
pop.upper.bound <- 0.5
geo.info <- zones(geo, population, pop.upper.bound)
```



# Index

## \* datasets

- NYleukemia, [22](#)
- NYleukemia\_sf, [23](#)
- pennLC, [24](#)
- pennLC\_sf, [25](#)
- scotland, [30](#)
- scotland\_sf, [31](#)

bayes\_cluster, [2](#)  
bayes\_cluster(), [8](#), [22](#), [23](#)  
besag\_newell, [5](#)

circle, [7](#)  
create\_geo\_objects, [8](#)  
create\_geo\_objects(), [12](#)

eBayes, [9](#)  
eBayes(), [12](#)  
EBpostdens, [10](#)  
EBpostthresh, [11](#)  
estimate\_lambda, [12](#)  
expected, [13](#)

GammaPriorCh, [14](#)  
graphics::polygon(), [28](#)  
grid2latlong, [15](#)

kulldorff, [16](#)

latlong2grid, [18](#)  
leglabs, [20](#)  
LogNormalPriorCh, [20](#)

mapvariable, [21](#)

NYleukemia, [22](#)  
NYleukemia\_sf, [23](#)

pennLC, [24](#)  
pennLC\_sf, [25](#)  
plotmap, [26](#)

polygon2spatial\_polygon, [27](#)  
polygon2spatial\_polygon(), [30](#)  
process\_MCMC\_sample, [29](#)

scotland, [30](#)  
scotland\_sf, [31](#)  
SpatialPolygons-class, [8](#), [15](#), [21](#), [27](#), [28](#), [30](#)

zones, [32](#)