Package 'CRTspat'

January 21, 2026

Title Workflow for Cluster Randomised Trials with Spillover

Version 1.4.0

Maintainer Thomas Smith <Thomas-a.Smith@unibas.ch>

Description Design, workflow and statistical analysis of Cluster Randomised Trials of (health) interventions where there may be spillover between the arms (see <https://thomasasmith.github.io/index.html>).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

VignetteBuilder knitr

Additional_repositories https://inla.r-inla-download.org/R/stable/

Imports ggplot2, stats, utils, MASS, tidyr, magrittr, dplyr, OOR, lme4, sf, Matrix, spatstat.geom, spatstat.random, TSP, gee, rstan, sp, loo

Suggests knitr, rmarkdown, INLA, testthat (>= 2.0.0)

Config/testthat/edition 2

Depends R (>= 3.5.0)

NeedsCompilation no

Author Thomas Smith [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-3650-9381>), Lea Multerer [ctb], Mariah Silkey [ctb]

Repository CRAN

Date/Publication 2026-01-21 19:10:26 UTC

Contents

aggregateCRT 2
anonymize_site 3
coef.CRTanalysis 4

compute_distance 4
compute_mesh 6
CRTanalysis 7
CRTpower 11
CRTsp 13
CRTwrite 15
fitted.CRTanalysis 16
latlong_as_xy 17
plotCRT 18
predict.CRTanalysis 20
randomizeCRT 21
readdata 22
residuals.CRTanalysis 23
simulateCRT 24
specify_buffer 27
specify_clusters 28
summary.CRTanalysis 29
summary.CRTsp 30

Index **31**

`aggregateCRT` *Aggregate data across records with duplicated locations*

Description

`aggregateCRT` aggregates data from a `"CRTsp"` object or trial data frame containing multiple records with the same location, and outputs a list of class `"CRTsp"` containing single values for each location, for both the coordinates and the auxiliary variables.

Usage

```
aggregateCRT(trial, auxiliaries = NULL)
```

Arguments

`trial`	An object of class `"CRTsp"` containing locations (x,y) and variables to be summed
`auxiliaries`	vector of names of auxiliary variables to be summed across each location

Details

Variables that in the trial dataframe that are not included in `auxiliaries` are retained in the output algorithm `"CRTsp"` object, with the value corresponding to that of the first record for the location in the input data frame

Value

A list of class `"CRTsp"`

Examples

```
{
trial <- readdata('example_site.csv')
trial$base_denom <- 1
aggregated <- aggregateCRT(trial, auxiliaries = c("RDT_test_result","base_denom"))
}
```

---

anonymize_site *Anonymize locations of a trial site*

Description

anonymize_site transforms coordinates to remove potential identification information.

Usage

```
anonymize_site(trial, ID = NULL, latvar = "lat", longvar = "long")
```

Arguments

trial	"CRTsp" object or trial data frame with co-ordinates of households
ID	name of column used as an identifier for the points
latvar	name of column containing latitudes in decimal degrees
longvar	name of column containing longitudes in decimal degrees

Details

The coordinates are transformed to support confidentiality of information linked to households by replacing precise geo-locations with transformed co-ordinates which preserve distances but not positions. The input may have either lat long or x,y coordinates. The function first searches for any lat long co-ordinates and converts these to x,y Cartesian coordinates. These are then are rotated by a random angle about a random origin. The returned object has transformed co-ordinates re-centred at the origin. Centroids stored in the "CRTsp" object are removed. Other data are unchanged.

Value

A list of class "CRTsp".

Examples

```
#Rotate and reflect test site locations
transformedTestlocations <- anonymize_site(trial =  readdata("exampleCRT.txt"))
```

coef.CRTanalysis *Extract model coefficients*

Description

coef.CRTanalysis method for extracting model fitted values

Usage

```
## S3 method for class 'CRTanalysis'
coef(object, ...)
```

Arguments

object	CRTanalysis object
...	other arguments

Value

the model coefficients returned by the statistical model run within the CRTanalysis function

Examples

```
{example <- readdata('exampleCRT.txt')
exampleGEE <- CRTanalysis(example, method = "GEE")
coef(exampleGEE)
}
```

compute_distance *Compute distance or surround values for a cluster randomized trial*

Description

compute_distance computes distance or surround values for a cluster randomized trial (CRT)

Usage

```
compute_distance(
  trial,
  distance = "nearestDiscord",
  scale_par = NULL,
  auxiliary = NULL
)
```

Arguments

trial	an object of class "CRTsp" or a data frame containing locations in (x,y) coordinates, cluster assignments (factor cluster), and arm assignments (factor arm).
distance	the quantity(s) to be computed. Options are:

"nearestDiscord"	distance to nearest discordant location (km)
"distanceAssigned"	distance to the nearest pixel in the assigned cluster (km)
"disc"	disc
"kern"	kernel-based measure
"hdep"	Tukey half space depth
"sdep"	simplicial depth

scale_par	scale parameter equal to the disc radius in km if distance = "disc" or to the standard deviance of the kernels if distance = "kern"
auxiliary	"CRTsp" object containing external cluster and or arm assignments.

Details

For each selected distance measure, the function first checks whether the variable is already present, and carries out the calculations only if the corresponding field is absent from the trial data frame.

If distance = "nearestDiscord" is selected the computed values are Euclidean distances assigned a positive sign for the intervention arm of the trial, and a negative sign for the control arm.

If distance = "distanceAssigned" is selected the computed values are Euclidean distances to the nearest pixel in the auxiliary "CRTsp" object.

If distance = "disc" is specified, the disc statistic is computed for each location as the number of locations within the specified radius that are in the intervention arm (Anaya-Izquierdo & Alexander(2020)). The input value of scale_par is stored in the design list of the output "CRTsp" object. Recalculation is carried out if the input value of scale_par differs from the one in the input design list. The value of the the surround calculated based on intervened locations is divided by the value of the surround calculated on the basis of all locations, so the value returned is a proportion.

If distance = "kern" is specified, the Normal curve with standard deviation scale_par is used to simulate diffusion of the intervention effect by Euclidean distance. For each location in the trial, the contributions of all intervened locations are summed. As with distance = "disc", when distance = "kern" the surround calculated based on intervened locations is divided by the value of the surround calculated on the basis of all locations, so the value returned is a proportion.

If either distance = "hdep" or distance = "sdep" is specified then both the simplicial depth and Tukey half space depth are calculated using the algorithm of Rousseeuw & Ruts(1996). The half-depth probability within the intervention cloud (di) is computed with respect to other locations in the intervention arm (Anaya-Izquierdo & Alexander(2020)). The half-depth within the half-depth within the control cloud (dc) is also computed. CRTspat returns the proportion di/(dc + di).

If an auxiliary auxiliary "CRTsp" object is specified then either distanceAssigned or nearestDiscord

(the default) is computed with respect to the assignments in the auxiliary. If the auxiliary is a grid with `design$geometry` set to `'triangle'`, `'square'` or `'hexagon'` then the distance is computed to the edge of the nearest grid pixel in the discordant arm (using a circular approximation for the perimeter) rather than to the point location itself.

Value

The input `"CRTsp"` object with additional column(s) added to the `trial` data frame with variable name corresponding to the input value of `distance`.

Examples

```
{
# Calculate the disc with a radius of 0.5 km
exampletrial <- compute_distance(trial = readdata('exampleCRT.txt'),
distance = 'disc', scale_par = 0.5)
}
```

---

compute_mesh *Create INLA mesh for spatial analysis*

Description

`compute_mesh` create objects required for INLA analysis of an object of class `"CRTsp"`.

Usage

```
compute_mesh(
  trial = trial,
  offset = -0.1,
  max.edge = 0.25,
  inla.alpha = 2,
  maskbuffer = 0.5,
  pixel = 0.5
)
```

Arguments

`trial`	an object of class `"CRTsp"` or a data frame containing locations in (x,y) coordinates, cluster assignments (factor `cluster`), and arm assignments (factor `arm`) and outcome.
`offset`	see `inla.mesh.2d` documentation
`max.edge`	see `inla.mesh.2d` documentation
`inla.alpha`	parameter related to the smoothness (see `inla` documentation)
`maskbuffer`	numeric: width of buffer around points (km)
`pixel`	numeric: size of pixel (km)

Details

compute_mesh carries out the computationally intensive steps required for setting-up an INLA analysis of an object of class "CRTsp", creating the prediction mesh and the projection matrices. The mesh can be reused for different models fitted to the same geography. The computational resources required depend largely on the resolution of the prediction mesh. The prediction mesh is thinned to include only pixels centred at a distance less than maskbuffer from the nearest point.
A warning may be generated if the Matrix library is not loaded.

Value

list

- prediction Data frame containing the prediction points and covariate values
- A projection matrix from the observations to the mesh nodes.
- Ap projection matrix from the prediction points to the mesh nodes.
- indexs index set for the SPDE model
- spde SPDE model
- pixel pixel size (km)

Examples

```
{

# low resolution mesh for test dataset
library(Matrix)
example <- readdata('exampleCRT.txt')
exampleMesh=compute_mesh(example, pixel = 0.5)
}
```

CRTanalysis *Analysis of cluster randomized trial with spillover*

Description

CRTanalysis carries out a statistical analysis of a cluster randomized trial (CRT).

Usage

```
CRTanalysis(
  trial,
  method = "GEE",
  distance = "nearestDiscord",
  scale_par = NULL,
  cfunc = "L",
  link = "logit",
  numerator = "num",
```

```
  denominator = "denom",
  excludeBuffer = FALSE,
  alpha = 0.05,
  baselineOnly = FALSE,
  baselineNumerator = "base_num",
  baselineDenominator = "base_denom",
  personalProtection = FALSE,
  clusterEffects = TRUE,
  spatialEffects = FALSE,
  pixel = 0.1,
  control = NULL,
  requireMesh = FALSE,
  inla_mesh = NULL,
  verbose = FALSE
)
```

Arguments

`trial`	an object of class `"CRTsp"` or a data frame containing locations in (x,y) coordinates, cluster assignments (factor `cluster`), and arm assignments (factor `arm`) and outcome data (see details).
`method`	statistical method with options:

`"EMP"`	simple averages of the data
`"T"`	comparison of cluster means by t-test
`"GEE"`	Generalised Estimating Equations
`"LME4"`	Generalized Linear Mixed-Effects Models
`"INLA"`	Integrated Nested Laplace Approximation (INLA)
`"MCMC"`	Markov chain Monte Carlo using `"stan"`

`distance`	Measure of distance or surround with options:

`"nearestDiscord"`	distance to nearest discordant location (km)
`"disc"`	disc
`"kern"`	surround based on sum of normal kernels
`"hdep"`	Tukey half space depth
`"sdep"`	simplicial depth

`scale_par`	numeric: pre-specified value of the spillover parameter or disc radius for models where this is fixed (`cfunc = "R"`).
`cfunc`	transformation defining the spillover function with options:

"Z"	arm effects not considered	reference model
"X"	spillover not modelled	the only valid value of cfunc for methods "EMP", "T" and "GEE"
"L"	inverse logistic (sigmoid)	the default for "INLA" and "MCMC" methods
"P"	inverse probit (error function)	available with "INLA" and "MCMC" methods
"D"	diffusion model	only available with the "MCMC" method
"S"	piecewise linear	only available with the "MCMC" method
"E"	estimation of scale factor	only available with distance = "disc" or distance = "kern"
"R"	rescaled linear	

link link function with options:

"logit"	(the default). numerator has a binomial distribution with denominator denominator.
"log"	numerator is Poisson distributed with an offset of log(denominator).
"cloglog"	numerator is Bernoulli distributed with an offset of log(denominator).
"identity"	The outcome is numerator/denominator with a normally distributed error function.

numerator	string: name of numerator variable for outcome
denominator	string: name of denominator variable for outcome data (if present)
excludeBuffer	logical: indicator of whether any buffer zone (records with buffer=TRUE) should be excluded from analysis
alpha	numeric: confidence level for confidence intervals and credible intervals
baselineOnly	logical: indicator of whether required analysis is of effect size or of baseline only
baselineNumerator	string: name of numerator variable for baseline data (if present)
baselineDenominator	string: name of denominator variable for baseline data (if present)
personalProtection	logical: indicator of whether the model includes local effects with no spillover
clusterEffects	logical: indicator of whether the model includes cluster random effects
spatialEffects	logical: indicator of whether the model includes spatial random effects (available only for method = "INLA" or for method = "MCMC")
pixel	numeric: size of pixel in km for spatial model (used for method = "MCMC")
control	list: control options to be passed to the statistical fitting function (available only for method = "MCMC")
requireMesh	logical: indicator of whether spatial predictions are required (available only for method = "INLA")
inla_mesh	string: name of pre-existing INLA input object created by compute_mesh()
verbose	logical: indicator of whether progress indicators, stan code, and the result summary should be returned

Details

CRTanalysis is a wrapper for the statistical analysis packages: gee, INLA, rstan, and the t.test function of package stats.

The wrapper does not provide an interface to the full functionality of these packages. It is specific for typical analyses of cluster randomized trials with geographical clustering. Further details are provided in the vignette.

The key results of the analyses can be extracted using a summary() of the output list. The model_object in the output list is the usual output from the statistical analysis routine, and can be also be inspected with summary(), or analysed using stats::fitted() for purposes of evaluation of model fit etc..

For models with a complementary log-log link function specified with link = "cloglog". the numerator must be coded as 0 or 1. Technically the binomial denominator is then 1 and the value of denominator is used as a rate multiplier.

With the "INLA" method 'iid' random effects are used to model extra-Poisson variation.

Interval estimates for the coefficient of variation of the cluster level outcome are calculated using the method of Vangel (1996).

If a control list is provided then this is passed to the 'stan' call. In addition to the values allowed by 'rstan::stan()', the number of iterations can be specified via value of 'iter'.

Value

list of class CRTanalysis containing the following results of the analysis:

list of class CRTanalysis containing the following results of the analysis:

- description : description of the dataset
- method : statistical method
- pt_ests : point estimates
- int_ests : interval estimates
- model_object : object returned by the fitting routine
- spillover : function values and statistics describing the estimated spillover

Examples

```
example <- readdata('exampleCRT.txt')
# Analysis of test dataset by t-test
exampleT <- CRTanalysis(example, method = "T")
summary(exampleT)
# Standard GEE analysis of test dataset ignoring spillover
exampleGEE <- CRTanalysis(example, method = "GEE")
summary(exampleGEE)
# LME4 analysis with error function spillover function
exampleLME4 <- CRTanalysis(example, method = "LME4", cfunc = "P")
summary(exampleLME4)
```

CRTpower *Power and sample size calculations for a cluster randomized trial*

Description

CRTpower carries out power and sample size calculations for cluster randomized trials.

Usage

```
CRTpower(
  trial = NULL,
  locations = NULL,
  alpha = 0.05,
  desiredPower = 0.8,
  effect = NULL,
  yC = NULL,
  outcome_type = "d",
  sigma2 = NULL,
  denominator = 1,
  N = 1,
  ICC = NULL,
  cv_percent = NULL,
  c = NULL,
  sd_h = 0,
  spillover_interval = 0,
  contaminate_pop_pr = 0,
  distance_distribution = "normal"
)
```

Arguments

trial	dataframe or 'CRTsp' object: optional list of locations
locations	numeric: total number of units available for randomization (required if trial is not specified)
alpha	numeric: confidence level
desiredPower	numeric: desired power
effect	numeric: required effect size
yC	numeric: baseline (control) value of outcome
outcome_type	character: with options - 'y': continuous; 'n': count; 'e': event rate; 'p': proportion; 'd': dichotomous.
sigma2	numeric: variance of the outcome (required for outcome_type = 'y')
denominator	numeric: rate multiplier (for outcome_type = 'n' or outcome_type = 'e')

`N` numeric: mean of the denominator for proportions (for `outcome_type = 'p'`)

`ICC` numeric: Intra-cluster correlation

`cv_percent` numeric: Coefficient of variation of the outcome (expressed as a percentage)

`c` integer: number of clusters in each arm (required if `trial` is not specified)

`sd_h` numeric: standard deviation of number of units per cluster (required if `trial` is not specified)

`spillover_interval`
numeric: 95% spillover interval (km)

`contaminate_pop_pr`
numeric: Proportion of the locations within the 95% spillover interval.

`distance_distribution`
numeric: algorithm for computing distribution of spillover, with options - `'empirical'`: empirical distribution; `'normal'`: normal distribution.

Details

Power and sample size calculations are for an unmatched two-arm trial. For counts or event rate data the formula of Hayes & Bennett (1999), Int. J. Epi., 28(2) pp319–326 is used. This requires as an input the between cluster coefficient of variation (`cv_percent`). For continuous outcomes and proportions the formulae of Hemming et al, 2011 are used. These make use of the intra-cluster correlation in the outcome (`ICC`) as an input. If the coefficient of variation and not the ICC is supplied then the intra-cluster correlation is computed from the coefficient of variation using the formulae from Hayes & Moulton. If incompatible values for `ICC` and `cv_percent` are supplied then the value of the `ICC` is used.

The calculations do not consider any loss in power due to loss to follow-up and by default there is no adjustment for effects of spillover.

Spillover bias can be allowed for using a diffusion model of mosquito movement. If no location or arm assignment information is available then `contaminate_pop_pr` is used to parameterize the model using a normal approximation for the distribution of distance to discordant locations.

If a trial data frame or `'CRTsp'` object is input then this is used to determine the number of locations. If this input object contains cluster assignments then the numbers and sizes of clusters in the input data are used to estimate the power. If `spillover_interval > 0` and `distance_distribution = 'empirical'` then effects of spillover are incorporated into the power calculations based on the empirical distribution of distances to the nearest discordant location. (If `distance_distribution is not equal to 'empirical'` then the distribution of distances is assumed to be normal.

If geolocations are not input then power and sample size calculations are based on the scalar input parameters.

If buffer zones have been specified in the `'CRTsp'` object then separate calculations are made for the core area and for the full site.

The output is an object of class `'CRTsp'` containing any input trial data frame and values for:

- The required numbers of clusters to achieve the specified power.

- The design effect based on the input ICC.
- Calculations of the power ignoring any bias caused by loss to follow-up etc.
- Calculations of `delta`, the expected spillover bias.

Value

A list of class `'CRTsp'` object comprising the input data, cluster and arm assignments, trial description and results of power calculations

Examples

```
{# Power calculations for a binary outcome without input geolocations
examplePower1 <- CRTpower(locations = 3000, ICC = 0.10, effect = 0.4, alpha = 0.05,
    outcome_type = 'd', desiredPower = 0.8, yC=0.35, c = 20, sd_h = 5)
summary(examplePower1)
# Power calculations for a rate outcome without input geolocations
examplePower2 <- CRTpower(locations = 2000, cv_percent = 40, effect = 0.4, denominator = 2.5,
    alpha = 0.05, outcome_type = 'e', desiredPower = 0.8, yC = 0.35, c = 20, sd_h=5)
summary(examplePower2)
# Example with input geolocations
examplePower3 <- CRTpower(trial = readdata('example_site.csv'), desiredPower = 0.8,
    effect=0.4, yC=0.35, outcome_type = 'd', ICC = 0.05, c = 20)
summary(examplePower3)
# Example with input geolocations, randomisation, and spillover
example4 <- randomizeCRT(specify_clusters(trial = readdata('example_site.csv'), c = 20))
examplePower4 <- CRTpower(trial = example4, desiredPower = 0.8,
    effect=0.4, yC=0.35, outcome_type = 'd', ICC = 0.05, contaminate_pop_pr = 0.3)
summary(examplePower4)
}
```

`CRTsp` *Create or update a* `"CRTsp"` *object*

Description

`CRTsp` coerces data frames containing co-ordinates and location attributes into objects of class `"CRTsp"` or creates a new `"CRTsp"` object by simulating a set of Cartesian co-ordinates for use as the locations in a simulated trial site

Usage

```
CRTsp(
  x = NULL,
  design = NULL,
  geoscale = NULL,
  locations = NULL,
  kappa = NULL,
```

```
  mu = NULL,
  geometry = "point"
)
```

Arguments

x	an object of class "CRTsp" or a data frame containing locations in (x,y) coordinates, cluster assignments (factor cluster), and arm assignments (factor arm). Optionally specification of a buffer zone (logical buffer); any other variables required for subsequent analysis.
design	list: an optional list containing the requirements for the power of the trial
geoscale	numeric: standard deviation of random displacement from each settlement cluster center (for new objects)
locations	integer: number of locations in population (for new objects)
kappa	numeric: intensity of Poisson process of settlement cluster centers (for new objects)
mu	numeric: mean number of points per settlement cluster (for new objects)
geometry	with valid values 'point' (the default, corresponding to point locations), 'triangle', 'square' and 'hexagon' corresponding to grids constructed from pixels of regular polygons.

Details

If a data frame or "CRTsp" object is input then the output "CRTsp" object is validated, a description of the geography is computed and power calculations are carried out.

If geoscale, locations, kappa and mu are specified then a new trial dataframe is constructed corresponding to a novel simulated human settlement pattern. This is generated using the Thomas algorithm (rThomas) in spatstat.random allowing the user to defined the density of locations and degree of spatial clustering. The resulting trial data frame comprises a set of Cartesian coordinates centred at the origin.

Value

A list of class "CRTsp" containing the following components:

design	list:	parameters required for power calculations
geom_full	list:	summary statistics describing the site
geom_core	list:	summary statistics describing the core area (when a buffer is specified)
trial	data frame:	rows correspond to geolocated points, as follows:
	x	numeric vector: x-coordinates of locations
	y	numeric vector: y-coordinates of locations
	cluster	factor: assignments to cluster of each location
	arm	factor: assignments to "control" or "intervention" for each location
	nearestDiscord	numeric vector: Euclidean distance to nearest discordant location (km)

buffer	logical: indicator of whether the point is within the buffer
...	other objects included in the input "CRTsp" object or data frame

Examples

```
{# Generate a simulated area with 10,000 locations
example_area = CRTsp(geoscale = 1, locations=10000, kappa=3, mu=40)
summary(example_area)
}
```

CRTwrite *Export of GIS layer from 'CRTsp'*

Description

CRTwrite exports a simple features object in a GIS format

Usage

```
CRTwrite(
  object,
  dsn,
  feature = "clusters",
  buffer_width,
  maskbuffer = 0.2,
  ...
)
```

Arguments

object	object of class 'CRTsp'
dsn	dataset name (relative path) for output objects
feature	feature to be exported, options are:

'cluster'	cluster assignments
'arms'	arm assignments
'buffer'	buffer zone or spillover zone
'mask'	mask for areas that are distant from habitations

buffer_width	width of buffer between discordant locations (km)
maskbuffer	radius of buffer drawn around inhabited areas (km)
...	other arguments passed to 'sf::write_sf'

Details

'sf::write_sf' is used to format the output. The function returns TRUE on success, FALSE on failure, invisibly.

If the input object contains a 'centroid' then this is used to compute lat long coordinates, which are assigned the "WGS84" coordinate reference system. Otherwise the objects have equirectangular co-ordinates with centroid (0,0).

If feature = 'buffer' then buffer width determination is as described under plotCRT().

The output vector objects are constructed by forming a Voronoi tessellation of polygons around each of the locations and combining these polygons. The polygons on the outside of the study area extend outwards to an external rectangle. The 'mask' is used to mask out the areas of these polygons that are at a distance > maskbuffer from the nearest location.

Value

obj, invisibly

Examples

```
tmpdir = tempdir()
dsn <- paste0(tmpdir,'/arms')
CRTwrite(readdata('exampleCRT.txt'), dsn = dsn, feature = 'arms',
driver = 'ESRI Shapefile', maskbuffer = 0.2)
```

---

fitted.CRTanalysis *Extract model fitted values*

Description

fitted.CRTanalysis method for extracting model fitted values

Usage

```
## S3 method for class 'CRTanalysis'
fitted(object, ...)
```

Arguments

object	CRTanalysis object
...	other arguments

Value

the fitted values returned by the statistical model run within the CRTanalysis function

Examples

```
{example <- readdata('exampleCRT.txt')
exampleGEE <- CRTanalysis(example, method = "GEE")
fitted_values <- fitted(exampleGEE)
}
```

`latlong_as_xy` *Convert lat long co-ordinates to x,y*

Description

`latlong_as_xy` converts co-ordinates expressed as decimal degrees into x,y

Usage

```
latlong_as_xy(trial, latvar = "lat", longvar = "long")
```

Arguments

`trial`	A trial dataframe or list of class `"CRTsp"` containing latitudes and longitudes in decimal degrees
`latvar`	name of column containing latitudes in decimal degrees
`longvar`	name of column containing longitudes in decimal degrees

Details

The output object contains the input locations replaced with Cartesian coordinates in units of km, centred on (0,0), corresponding to using the equirectangular projection (valid for small areas). Other data are unchanged.

Value

A list of class `"CRTsp"` containing the following components:

`geom_full`	list:	summary statistics describing the site
`trial`	data frame:	rows correspond to geolocated points, as follows:
	`x`	numeric vector: x-coordinates of locations
	`y`	numeric vector: y-coordinates of locations
	`...`	other objects included in the input `"CRTsp"` object or data frame

Examples

```
examplexy <- latlong_as_xy(readdata("example_latlong.csv"))
```

plotCRT *Graphical displays of the geography of a CRT*

Description

plotCRT returns graphical displays of the geography of a CRT or of the results of statistical analyses of a CRT

Usage

```
plotCRT(
  object,
  map = FALSE,
  distance = "nearestDiscord",
  fill = "arms",
  showLocations = FALSE,
  showClusterBoundaries = TRUE,
  showClusterLabels = FALSE,
  showBuffer = FALSE,
  cpalette = NULL,
  buffer_width = NULL,
  maskbuffer = 0.2,
  labelsize = 4,
  legend.position = NULL
)
```

Arguments

object	object of class 'CRTanalysis' produced by CRTanalysis()
map	logical: indicator of whether a map is required
distance	measure of distance or surround with options:

"nearestDiscord"	distance to nearest discordant location (km)
"disc"	disc
"hdep"	Tukey's half space depth
"sdep"	simplicial depth

fill	fill layer of map with options:

'cluster'	cluster assignment
'arms'	arm assignment
'nearestDiscord'	distance to the nearest discordant location
'disc'	disc measure of surround
'hdep'	Tukey's half space depth

'sdep'	simplicial depth
'prediction'	model prediction of the outcome
'none'	No fill

showLocations	logical: determining whether locations are shown
showClusterBoundaries	logical: determining whether cluster boundaries are shown
showClusterLabels	logical: determining whether the cluster numbers are shown
showBuffer	logical: whether a buffer zone should be overlayed
cpalette	colour palette (to use different colours for clusters this must be at least as long as the number of clusters.
buffer_width	width of buffer zone to be overlayed (km)
maskbuffer	radius of buffer around inhabited areas (km)
labelsize	size of cluster number labels
legend.position	(using ggplot2::themes syntax)

Details

If map = FALSE and the input is a trial data frame or a CRTsp object, containing a randomisation to arms, a stacked bar chart of the outcome grouped by the specified distance is produced. If the specified distance has not yet been calculated an error is returned.

If map = FALSE and the input is a CRTanalysis object a plot of the estimated spillover function is generated. The fitted spillover function is plotted as a continuous green/blue line against the measure of the surround or of the distance to the nearest discordant location. Using the same axes, data summaries are plotted for ten categories of distance from the boundary. Both the average of the outcome and confidence intervals are plotted.If spillover limits have been estimated then these limits are used to delimit a shaded rectangular overlay, providing they fall within the range of distances in the data. If the spillover limits fall outside the range of the data then the range of distances in the data define the boundaries of the shaded area.

- For analyses with logit link function the outcome is plotted as a proportion.
- For analyses with log or cloglog link function the data are plotted on a scale of the Williams mean (mean of exp(log(x + 1))) - 1) rescaled so that the median matches the fitted curve at the midpoint.

If map = TRUE a thematic map corresponding to the value of fill is generated.

- fill = 'clusters' or leads to thematic map showing the locations of the clusters
- fill = 'arms' leads to a thematic map showing the geography of the randomization
- fill = 'distance' leads to a raster plot of the distance to the nearest discordant location.

- fill = 'prediction' leads to a raster plot of predictions from an 'INLA' model.

If showBuffer = TRUE the map is overlaid with a grey transparent layer showing which areas are within a defined distance of the boundary between the arms. Possibilities are:

- If the trial has not been randomised or if showBuffer = FALSE no buffer is displayed
- If buffer_width takes a positive value then buffers of this width are displayed irrespective of any pre-specified or spillover limits.
- If the input is a 'CRTanalysis' and spillover limits have been estimated then these limits are used to define the displayed buffer providing they fall within the range of distances in the data.
- If buffer_width is not specified and no spillover limits are available, then any pre-specified buffer (e.g. one generated by specify_buffer()) is displayed.

A message is output indicating which of these possibilities applies.

Value

graphics object produced by the ggplot2 package

Examples

```
{example <- readdata('exampleCRT.txt')
#Plot of data by distance
plotCRT(example)
#Map of locations only
plotCRT(example, map = TRUE, fill = 'none', showLocations = TRUE,
           showClusterBoundaries=FALSE, maskbuffer=0.2)
#show cluster boundaries and number clusters
plotCRT(example, map = TRUE, fill ='none', showClusterBoundaries=TRUE,
           showClusterLabels=TRUE, maskbuffer=0.2, labelsize = 2)
#show clusters in colour
plotCRT(example, map = TRUE, fill = 'clusters', showClusterLabels = TRUE,
          labelsize=2, maskbuffer=0.2)
#show arms
plotCRT(example, map = TRUE,
fill = 'arms', maskbuffer=0.2, legend.position=c(0.8,0.8))
#spillover plot
analysis <- CRTanalysis(example)
 plotCRT(analysis, map = FALSE)
}
```

---

predict.CRTanalysis *Model predictions*

Description

predict.CRTanalysis method for extracting model predictions

Usage

```
## S3 method for class 'CRTanalysis'
predict(object, ...)
```

Arguments

object	CRTanalysis object
...	other arguments

Value

the model predictions returned by the statistical model run within the CRTanalysis function

Examples

```
{example <- readdata('exampleCRT.txt')
exampleGEE <- CRTanalysis(example, method = "GEE")
predictions <- predict(exampleGEE)
}#'
```

---

randomizeCRT *Randomize a two-armed cluster trial*

Description

randomizeCRT carries out randomization of clusters and augments the trial data frame with assignments to arms

Usage

```
randomizeCRT(
  trial,
  matchedPair = FALSE,
  baselineNumerator = "base_num",
  baselineDenominator = "base_denom"
)
```

Arguments

trial	an object of class "CRTsp" or a data frame containing locations in (x,y) coordinates, cluster assignments (factor cluster), and arm assignments (factor arm). Optionally: specification of a buffer zone (logical buffer); any other variables required for subsequent analysis.
matchedPair	logical: indicator of whether pair-matching on the baseline data should be used in randomization

`baselineNumerator`
: name of numerator variable for baseline data (required for matched-pair randomization)

`baselineDenominator`
: name of denominator variable for baseline data (required for matched-pair randomization)

Value

A list of class `"CRTsp"` containing the following components:

`design`	list:	parameters required for power calculations
`geom_full`	list:	summary statistics describing the site
`geom_core`	list:	summary statistics describing the core area (when a buffer is specified)
`trial`	data frame:	rows correspond to geolocated points, as follows:
	`x`	numeric vector: x-coordinates of locations
	`y`	numeric vector: y-coordinates of locations
	`cluster`	factor: assignments to cluster of each location
	`pair`	factor: assigned matched pair of each location (for `matchedPair` randomisations)
	`arm`	factor: assignments to `"control"` or `"intervention"` for each location
	`...`	other objects included in the input `"CRTsp"` object or data frame

Examples

```
# Randomize the clusters in an example trial
exampleCRT <- randomizeCRT(trial = readdata('exampleCRT.txt'), matchedPair = TRUE)
```

`readdata` *Read example dataset*

Description

`readdata` reads a file from the package library of example datasets

Usage

```
readdata(filename)
```

Arguments

`filename`	name of text file stored within the package

Details

The input file name should include the extension (either .csv or .txt). The resulting object is a data frame if the extension is .csv.

Value

R object corresponding to the text file

Examples

```
exampleCRT <- readdata('exampleCRT.txt')
```

`residuals.CRTanalysis` *Extract model residuals*

Description

`residuals.CRTanalysis` method for extracting model residuals

Usage

```
## S3 method for class 'CRTanalysis'
residuals(object, ...)
```

Arguments

`object`	CRTanalysis object
`...`	other arguments

Value

the residuals from the statistical model run within the `CRTanalysis` function

Examples

```
{example <- readdata('exampleCRT.txt')
exampleGEE <- CRTanalysis(example, method = "GEE")
residuals <- residuals(exampleGEE)
}
```

simulateCRT *Simulation of cluster randomized trial with spillover*

Description

simulateCRT generates simulated data for a cluster randomized trial (CRT) with geographic spillover between arms.

Usage

```
simulateCRT(
  trial = NULL,
  effect = 0,
  outcome0 = NULL,
  generateBaseline = TRUE,
  matchedPair = TRUE,
  scale = "proportion",
  baselineNumerator = "base_num",
  baselineDenominator = "base_denom",
  denominator = NULL,
  ICC_inp = NULL,
  kernels = 200,
  sigma_m = NULL,
  spillover_interval = NULL,
  tol = 0.005
)
```

Arguments

trial	an object of class "CRTsp" or a data frame containing locations in (x,y) coordinates, cluster assignments (factor cluster), and arm assignments (factor arm). Each location may also be assigned a propensity (see details).
effect	numeric. The simulated effect size (defaults to 0)
outcome0	numeric. The anticipated value of the outcome in the absence of intervention
generateBaseline	logical. If TRUE then baseline data and the propensity will be simulated
matchedPair	logical. If TRUE then the function tries to carry out randomization using pair-matching on the baseline data (see details)
scale	measurement scale of the outcome. Options are: 'proportion' (the default); 'count'; 'continuous'.
baselineNumerator	optional name of numerator variable for pre-existing baseline data
baselineDenominator	optional name of denominator variable for pre-existing baseline data
denominator	optional name of denominator variable for the outcome

ICC_inp	numeric. Target intra cluster correlation, provided as input when baseline data are to be simulated
kernels	number of kernels used to generate a de novo propensity
sigma_m	numeric. standard deviation of the normal kernel measuring spatial smoothing leading to spillover
spillover_interval	numeric. input spillover interval
tol	numeric. tolerance of output ICC

Details

Synthetic data are generated by sampling around the values of variable propensity, which is a numerical vector (taking positive values) of length equal to the number of locations. There are three ways in which propensity can arise:

1. propensity can be provided as part of the input trial object.
2. Baseline numerators and denominators (values of baselineNumerator and baselineDenominator may be provided. propensity is then generated as the numerator:denominator ratio for each location in the input object
3. Otherwise propensity is generated using a 2D Normal kernel density. The OOR::StoSOO is used to achieve an intra-cluster correlation coefficient (ICC) that approximates the value of 'ICC_inp' by searching for an appropriate value of the kernel bandwidth.

num[i], the synthetic outcome for location i is simulated with expectation:

$$E(num[i]) = outcome0[i] * propensity[i] * denom[i] * (1 - effect * I[i]) / mean(outcome0[] * propensity[])$$

The sampling distribution of num[i] depends on the value of scale as follows:

- scale='continuous': Values of num are sampled from a Normal distributions with means E(num[i]) and variance determined by the fitting to ICC_inp.
- scale='count': Simulated events are allocated to locations via multivariate hypergeometric distributions parameterised with E(num[i]).
- scale='proportion': Simulated events are allocated to locations via multinomial distributions parameterised with E(num[i]).

denominator may specify a vector of numeric (non-zero) values in the input "CRTsp" or data.frame which is returned as variable denom. It acts as a scale-factor for continuous outcomes, rate-multiplier for counts, or denominator for proportions. For discrete data all values of denom must be > 0.5 and are rounded to the nearest integer in calculations of num.

By default, denom is generated as a vector of ones, leading to simulation of dichotomous outcomes if scale='proportion'.

If baseline numerators and denominators are provided then the output vectors `base_denom` and `base_num` are set to the input values. If baseline numerators and denominators are not provided then the synthetic baseline data are generated by sampling around `propensity` in the same way as the outcome data, but with the effect size set to zero.

If `matchedPair` is `TRUE` then pair-matching on the baseline data will be used in randomization providing there are an even number of clusters. If there are an odd number of clusters then matched pairs are not generated and an unmatched randomization is output.

Either `sigma_m` or `spillover_interval` must be provided. If both are provided then the value of `sigma_m` is overwritten by the standard deviation implicit in the value of `spillover_interval`. Spillover is simulated as arising from a diffusion-like process. For further details see Multerer (2021)

Value

A list of class `"CRTsp"` containing the following components:

`geom_full`	list:	summary statistics describing the site cluster assignments, and randomization
`design`	list:	values of input parameters to the design
`trial`	data frame:	rows correspond to geolocated points, as follows:
	`x`	numeric vector: x-coordinates of locations
	`y`	numeric vector: y-coordinates of locations
	`cluster`	factor: assignments to cluster of each location
	`arm`	factor: assignments to `control` or `intervention` for each location
	`nearestDiscord`	numeric vector: signed Euclidean distance to nearest discordant location (km)
	`propensity`	numeric vector: propensity for each location
	`base_denom`	numeric vector: denominator for baseline
	`base_num`	numeric vector: numerator for baseline
	`denom`	numeric vector: denominator for the outcome
	`num`	numeric vector: numerator for the outcome
	`...`	other objects included in the input `"CRTsp"` object or `data.frame`

Examples

```
{smalltrial <- readdata('smalltrial.csv')
 simulation <- simulateCRT(smalltrial,
  effect = 0.25,
  ICC_inp = 0.05,
  outcome0 = 0.5,
  matchedPair = FALSE,
  scale = 'proportion',
  sigma_m = 0.6,
  tol = 0.05)
 summary(simulation)
 }
```

specify_buffer *Specification of buffer zone in a cluster randomized trial*

Description

specify_buffer specifies a buffer zone in a cluster randomized trial (CRT) by flagging those locations that are within a defined distance of those in the opposite arm.

Usage

```
specify_buffer(trial, buffer_width = 0)
```

Arguments

trial	an object of class "CRTsp" or a data frame containing locations in (x,y) coordinates, cluster assignments (factor cluster), and arm assignments (factor arm).
buffer_width	minimum distance between locations in opposing arms for them to qualify to be included in the core area (km)

Value

A list of class "CRTsp" containing the following components:

geom_full	list:	summary statistics describing the site, cluster assignments, and randomization.
geom_core	list:	summary statistics describing the core area
trial	data frame:	rows correspond to geolocated points, as follows:
	x	numeric vector: x-coordinates of locations
	y	numeric vector: y-coordinates of locations
	cluster	factor: assignments to cluster of each location
	arm	factor: assignments to "control" or "intervention" for each location
	nearestDiscord	numeric vector: signed Euclidean distance to nearest discordant location (km)
	buffer	logical: indicator of whether the point is within the buffer
	...	other objects included in the input "CRTsp" object or data frame

Examples

```
#Specify a buffer of 200m
exampletrial <- specify_buffer(trial = readdata('exampleCRT.txt'), buffer_width = 0.2)
```

specify_clusters *Assign locations to clusters in a CRT*

Description

specify_clusters algorithmically assigns locations to clusters by grouping them geographically

Usage

```
specify_clusters(
  trial = trial,
  c = NULL,
  h = NULL,
  algorithm = "NN",
  reuseTSP = FALSE,
  auxiliary = NULL
)
```

Arguments

trial	A CRT object or data frame containing (x,y) coordinates of households
c	integer: number of clusters in each arm
h	integer: number of locations per cluster
algorithm	algorithm for cluster boundaries, with options:

NN	Nearest neighbour: assigns equal numbers of locations to each cluster
kmeans	kmeans clustering: aims to partition locations so that each belongs to the cluster with the nearest centroid.
TSP	travelling salesman problem heuristic: Assigns locations sequentially along a travelling salesman path.

reuseTSP	logical: indicator of whether a pre-existing path should be used by the TSP algorithm
auxiliary	"CRTsp" object containing external cluster and or arm assignments.

Details

Either c or h must be specified. If both are specified the input value of c is ignored.

The reuseTSP parameter is used to allow the path to be reused for creating alternative allocations with different cluster sizes.

If an auxiliary auxiliary "CRTsp" object is specified then the other options are ignored and the cluster assignments (and arm assignments if available) are taken from the auxiliary object. The trial data frame is augmented with a column "nearestPixel" containing the distance to boundary of the nearest grid pixel in the auxiliary. If the auxiliary is a grid with design$geometry set to 'triangle', 'square' or 'hexagon' then the distance is computed to the edge of the nearest grid

pixel in the discordant arm (using a circular approximation for the perimeter) rather than to the point location itself. If the point is within the pixel then the distance is given a negative sign.

Value

A list of class "CRTsp" containing the following components:

geom_full	list:	summary statistics describing the site, and cluster assignments.
trial	data frame:	rows correspond to geolocated points, as follows:
	x	numeric vector: x-coordinates of locations
	y	numeric vector: y-coordinates of locations
	cluster	factor: assignments to cluster of each location
	...	other objects included in the input "CRTsp" object or data frame

Examples

```
#Assign clusters of average size h = 40 to a test set of co-ordinates, using the kmeans algorithm
exampletrial <- specify_clusters(trial = readdata('exampleCRT.txt'),
                            h = 40, algorithm = 'kmeans', reuseTSP = FALSE)
```

---

summary.CRTanalysis *Summary of the results of a statistical analysis of a CRT*

Description

summary.CRTanalysis generates a summary of a CRTanalysis including the main results

Usage

```
## S3 method for class 'CRTanalysis'
summary(object, ...)
```

Arguments

object	an object of class "CRTanalysis"
...	other arguments used by summary

Value

No return value, writes text to the console.

Examples

```
{example <- readdata('exampleCRT.txt')
exampleT <- CRTanalysis(example, method = "T")
summary(exampleT)
}
```

summary.CRTsp *Summary description of a* "CRTsp" *object*

Description

summary.CRTsp provides a description of a "CRTsp" object

Usage

```
## S3 method for class 'CRTsp'
summary(object, maskbuffer = 0.2, ...)
```

Arguments

object	an object of class "CRTsp" or a data frame containing locations in (x,y) coordinates, cluster assignments (factor cluster), arm assignments (factor arm) and buffer zones (logical buffer), together with any other variables required for subsequent analysis.
maskbuffer	radius of area around a location to include in calculation of areas
...	other arguments used by summary

Value

No return value, write text to the console.

Examples

```
summary(CRTsp(readdata('exampleCRT.txt')))
```

Index

aggregateCRT, 2
anonymize_site, 3

coef.CRTanalysis, 4
compute_distance, 4
compute_mesh, 6
CRTanalysis, 7
CRTpower, 11
CRTsp, 13
CRTwrite, 15

fitted.CRTanalysis, 16

latlong_as_xy, 17

plotCRT, 18
predict.CRTanalysis, 20

randomizeCRT, 21
readdata, 22
residuals.CRTanalysis, 23

simulateCRT, 24
specify_buffer, 27
specify_clusters, 28
summary.CRTanalysis, 29
summary.CRTsp, 30