

# Package: CRTspat (via r-universe)

August 25, 2024

**Title** Workflow for Cluster Randomised Trials with Spillover

**Version** 1.3.0

**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

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**Additional\_repositories** <https://inla.r-inla-download.org/R/stable/>

**Imports** ggplot2, stats, utils, geepack, MASS, tidyr, magrittr, dplyr, OOR, lme4, sf, Matrix, spatstat.geom, spatstat.random, jagsUI, TSP

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

**Config/testthat/edition** 2

**Depends** R (>= 3.5.0)

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

**RemoteUrl** <https://github.com/thomasasmith/crtspat>

**RemoteRef** HEAD

**RemoteSha** 0f57da7625fcf0229624adbe050cc8c70fa6c50e

## Contents

aggregateCRT . . . . .	2
CRTanalysis . . . . .	3

CRTsp . . . . .	6
latlong_as_xy . . . . .	7
plotCRT . . . . .	8
randomizeCRT . . . . .	11
readdata . . . . .	12
simulateCRT . . . . .	13
specify_buffer . . . . .	16
specify_clusters . . . . .	17
summary.CRTanalysis . . . . .	18

## Index 19

---

aggregateCRT	<i>Aggregate data across records with duplicated locations</i>
--------------	--

---

### 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"))
}
```

**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,
  requireMesh = FALSE,
  inla_mesh = 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) and outcome data (see details).
method	statistical method with options: <ul style="list-style-type: none"> <li>"EMP" simple averages of the data</li> <li>"T" comparison of cluster means by t-test</li> <li>"GEE" Generalised Estimating Equations</li> <li>"LME4" Generalized Linear Mixed-Effects Models</li> <li>"INLA" Integrated Nested Laplace Approximation (INLA)</li> <li>"MCMC" Markov chain Monte Carlo using "JAGS"</li> <li>"WCA" Within cluster analysis</li> </ul>

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
"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	

<code>clusterEffects</code>	logical: indicator of whether the model includes cluster random effects
<code>spatialEffects</code>	logical: indicator of whether the model includes spatial random effects (available only for <code>method = "INLA"</code> )
<code>requireMesh</code>	logical: indicator of whether spatial predictions are required (available only for <code>method = "INLA"</code> )
<code>inla_mesh</code>	string: name of pre-existing INLA input object created by <code>compute_mesh()</code>

## Details

CRTanalysis is a wrapper for the statistical analysis packages: `geepack`, `INLA`, `jagsUI`, 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. The value of denominator is used as a rate multiplier.

With the "INLA" and "MCMC" methods '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 \(1996\)](#).

## Value

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

- `description` : description of the dataset
- `method` : statistical method
- `pt_est` : point estimates
- `int_est` : 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)
```

---

 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)
}
```

---

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: <table> <tbody> <tr> <td>"nearestDiscord"</td> <td>distance to nearest discordant location (km)</td> </tr> <tr> <td>"disc"</td> <td>disc</td> </tr> <tr> <td>"hdep"</td> <td>Tukey's half space depth</td> </tr> <tr> <td>"sdep"</td> <td>simplicial depth</td> </tr> </tbody> </table>	"nearestDiscord"	distance to nearest discordant location (km)	"disc"	disc	"hdep"	Tukey's half space depth	"sdep"	simplicial depth								
"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: <table> <tbody> <tr> <td>'cluster'</td> <td>cluster assignment</td> </tr> <tr> <td>'arms'</td> <td>arm assignment</td> </tr> <tr> <td>'nearestDiscord'</td> <td>distance to the nearest discordant location</td> </tr> <tr> <td>'disc'</td> <td>disc measure of surround</td> </tr> <tr> <td>'hdep'</td> <td>Tukey's half space depth</td> </tr> <tr> <td>'sdep'</td> <td>simplicial depth</td> </tr> <tr> <td>'prediction'</td> <td>model prediction of the outcome</td> </tr> <tr> <td>'none'</td> <td>No fill</td> </tr> </tbody> </table>	'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
'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 overlaid																
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 overlaid (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 blue line against the measure 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.

- 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 ( $\text{mean of } \exp(\log(x + 1)) - 1$ ) 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 by an 'LME4' or 'INLA' model then these limits are used to define the displayed buffer.
- 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)
}

```

---

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	<i>Read example dataset</i>
----------	-----------------------------

---

**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')
```

---

 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 `00R::StoS00` 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:

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

## 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	<i>Specification of buffer zone in a cluster randomized trial</i>
----------------	---

---

### 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)
}
```

# Index

`aggregateCRT`, [2](#)

`CRTanalysis`, [3](#)

`CRTsp`, [6](#)

`latlong_as_xy`, [7](#)

`plotCRT`, [8](#)

`randomizeCRT`, [11](#)

`readdata`, [12](#)

`simulateCRT`, [13](#)

`specify_buffer`, [16](#)

`specify_clusters`, [17](#)

`summary.CRTanalysis`, [18](#)