Package 'stppSim'

March 7, 2022

Type Package

Title Spatial and Temporal Point Patterns Simulation for Social and Life Science Research

Version 1.0.0

Author Monsuru Adepeju [cre, aut],
Meng Le Zhang [aut]

Maintainer Monsuru Adepeju <monsuur2010@yahoo.com>

Description Generates artificial spatio-temporal point patterns (stpp) by integrating microsimulation and agent-based models. Allows a user to define the movement properties of a set of ` ` walkers" (an agents) distributed across a spatially configured landscape. The interaction of the walkers (be it human offenders in a crime analysis, foraging animals in a Wildlife study, or carriers in disease transmission) with their environment results in the advent of new events, whose spatiotemporal (ST) signatures are measurable at the global levels. These signatures can be used in ST model testing and evaluation across wide variety of fields in social and life sciences.

Language en-US

License GPL-3

URL https://github.com/MAnalytics/stppSim

BugReports https://github.com/Manalytics/stppSim/issues/new/choose

Depends R (>= 4.0.0)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Imports splancs,
dplyr,
tidyr,
magrittr,
sf,
rgdal,
sp,
rgeos,
terra,
raster,
SiMRiv,
data.table,

tibble,
stringr,
lubridate,
spatstat.geom,
sparr,
ggplot2,
doParallel,
foreach,
iterators,
chron

RoxygenNote 7.1.1

Suggests knitr,
rmarkdown,
parallel,
graphics,
grDevices,
utils

VignetteBuilder knitr

R topics documented:

`artif_spo` *Artificial spatial event origins*

Description

Simulate spatial points to serve as event origins across the space. If provided, resistance features are to be avoided. Each final event origin point is assigned a probability value indicating the strength of the origin.

Usage

```
artif_spo(poly, n_origin=50, resistance_feat = NULL,
n_foci=5, foci_separation = 10,
conc_type = "nucleated", p_ratio)
```

Arguments

poly	(An sf or S4 object) A polygon shapefile within which event origins are to be situated.
n_origin	(an integer) Number of locations from which the walkers originate. Default:50.
resistance_feat	(An S4 object) Optional shapefile representing spaces across landscape within which event origins are not allowed. Default: NULL.
n_foci	(an integer) Value indicating the number of focal points amongst event origins. n_foci will usually be smaller than n_origin.
foci_separation	(an integer) A percentage value indicating indicating the nearness of focal points from one another. A 0 separation indicates that focal points are in close proximity of one another, while a 100 indicates focal points being evenly distributed across space.
conc_type	(string) Specifies the spatial pattern of non-focal origin (strengths) in relation to to their nearest focal origins. Value is either "nucleated" or "dispersed".
p_ratio	(an integer) The smaller of the two terms of a Pareto ratio. For example, a value of 20 implies a 20:80 Pareto ratio.

Details

Details of events origins: x,y locations, categories (i.e. focal and non-focal (others) origins), and the probability values. and non-focal point. The focal origins (n_foci) serve as the more dominant origins (e.g. city centres), while the non-focal origins (i.e. non-dominant) origin. The foci_separation indicates the nearness of dominant origins from one another. The conc_type argument allows a user to specify the type of spatial patterns exhibited by the non-focal points around the focal points (See vignette for details). If resistance_feat is provided, the features help to prevent event origins from being situated in the same locations occupied by the features.

Value

Returns event origins with their respective strength (probability) values.

Examples

```
data(camden_boundary)
data(landuse)
spo <- artif_spo(poly = camden_boundary, n_origin = 50,
resistance_feat = landuse, n_foci=5,
foci_separation = 0, conc_type = "dispersed", p_ratio=20)
```

camden_boundary *A boundary shapefile*

Description

A boundary shapefile of Camden Borough, London, UK

Usage

```
camden_boundary
```

Format

A boundary file (ESRI format)

- x: x coordinate
- y: y coordinate

camden_theft *'theft-from-person' crime of Camden Borough, London, UK*

Description

A spatiotemporal point data ('theft-from-person' crime: Source: `police.uk`) of Camden Borough of London, UK of the year 2021. Note: Police.uk data is only accurate to monthly scale (`yyyy-mm`). However, the data here is provided at the daily scales by adding random `days` to produce the dates in the format `yyyy-mm-dd`. So, caution should be taken when interpreting the results from this datasets.

Usage

```
camden_theft
```

Format

A matrix containing three variables

- x: x coordinate
- y: y coordinate
- t: t time

chull_poly *Boundary surrounding a set of points*

Description

Generates a boundary (polygon) around a set of points, using Convex Hull technique (Eddy, W. F, 1977).

Usage

```
chull_poly(xycoords,
crsys = NULL)
```

Arguments

xycoords	(matrix) A 2-column coordinate vectors of points: x - the eastings, and y - the northing.
crsys	Optional string specifying the coordinate reference system (crs) of the resulting boundary, e.g., the crs string "+proj=longlat +datum=WGS84" transform the resulting boundary to wgs84 system.

Details

Draw an arbitrary boundary around by joining the outer-most points amongst the spatial data points

References

Eddy, W. F. (1977). A new convex hull algorithm for planar sets. ACM Transactions on Mathematical Software, 3, 398–403.10.1145/355759.355766.

Examples

```
data(xyt_data)
#extract xy coordinates only
xy <- matrix(as.numeric(xyt_data[,1:2]),,2)
bry <- chull_poly(xy, crsys = NULL)
#visualise result
#plot(bry) #to plot
#points(xy[,1], xy[,2], add=TRUE)
```

date_checker *Date (Format) Checker*

Description

Checks if date is in the correct format.

Usage

```
date_checker(x)
```

Arguments

x	A vector of date values

Details

Returns "TRUE" if all entries of a vector are date values and FALSE if any entries of a vector is not a date value. The date vector needs to be in the format: "yyyy-mm-dd".

Value

Returns TRUE or FALSE

Examples

```
date_list_1 <- c("2021-09-12", "2016-xx-02",
"09/08/2012")
date_checker(date_list_1)
#> FALSE (Entries 2 and 3
#are incorrect date inputs)
date_list_2 <- c("2021-09-12", "1998-03-09")
date_checker(date_list_2)
#> TRUE
```

---

extract_coords *Coordinates extraction*

Description

Extracts the bounding (edges) coordinates of a polygon object.

Usage

```
extract_coords(poly)
```

Arguments

poly	(An sf or S4 object) A polygon shapefile.

Details

Given a spatial polygon object, the function extracts the bounding coordinates of the object.

Value

Returns 2-column xy coordinates

Examples

```
data(camden_boundary)
extract_coords(poly=camden_boundary)
```

gtp *Global temporal pattern (GTP)*

Description

Models the global temporal pattern, as combining long-term trend and seasonality.

Usage

```
gtp(start_date, trend = "stable",
slope = NULL, first_pDate = NULL, show.plot =FALSE)
```

Arguments

start_date	The start date of temporal pattern. The date should be in the format "yyyy-mm-dd". The GTP will usually covers a 1-year period.
trend	(string) Specify the trend direction of the GTP. Values are: "decreasing", "stable", and "increasing". Default is: "stable".
slope	(string) Slope GTP trend if "increasing" or "decreasing" trend is specified. Values: "gentle" or "steep". Default value is NULL (i.e., for stable trend).
first_pDate	(in "yyyy-mm-dd" format). Date of the first seasonal peak of the time series. Default value is NULL, in which a seasonal cycle of 180 days is utilized. That is, a first seasonal peak of 90 days.
show.plot	(logical) Shows GTP. Default is FALSE.

Details

Models the GTP which could be utilize for simulating artifical point pattern across space.

Value

Returns a vector of 365 data points representing the global temporal pattern

Examples

```
gtp(start_date = "2020-01-01", trend = "stable",
slope = NULL, first_pDate = "2020-02-28", show.plot = FALSE)
```

landuse *Landuse shapefile*

Description

A land use shapefile of Camden Borough of London, United Kingdom

Usage

```
landuse
```

Format

A boundary file (ESRI format)

- type: Landuse type
- rValues1: Field specifying a uniform resistance for all features (Value = 1) for all features
- rValues2: Field specifying varying resistance values for different feature classes.

`make_grids` *Make square grids*

Description

Generates a system of square grids across a specified spatial polygon (boundary).

Usage

```
make_grids(poly, size = 250,
show_output = FALSE)
```

Arguments

`poly`	(as `spatialPolygons`, `spatialPolygonDataFrames`, or simple features). A polygon object over which square grids are to be overlaid.
`size`	Size of square grids to be created. For example, the input `size` for a 250 by 250 square grids is `250`.
`show_output`	(logical) Display the output. Default: `FALSE`

Details

Exports a grid system in a shapefile format (in the same crs as the input `poly`)

Value

Returns a spatial square grid system in a shapefile format

Examples

```
data(camden_boundary)
make_grids(poly=camden_boundary, size = 250,
show_output = FALSE)
```

poly *Boundary Coordinates*

Description

Boundary coordinates of Camden Borough of London

Usage

```
poly
```

Format

A dataframe containing one variable:

- x: x coordinate
- y: y coordinate

poly_tester *Geometry and crs of a polygon*

Description

Test whether an input polygon has the correct geometry, namely `S4` or `sf`. Further, to test that there is a valid projection attached to the polygon.

Usage

```
poly_tester(poly)
```

Arguments

poly	(as `spatialPolygons`, `spatialPolygonDataFrames`, or simple features). A spatial polygon

Details

Returns an error message if the polygon is not in correct geometry or CRS.

Value

Returns error messages, or mute

Examples

```
poly_tester(poly=camden_boundary)
```

psim_artif *Stpp from synthetic origins*

Description

Generate spatiotemporal point pattern from synthesized origins.

Usage

```
psim_artif(n_events=1000, start_date = "yyyy-mm-dd",
poly, n_origin, resistance_feat, field,
n_foci, foci_separation, conc_type = "dispersed",
p_ratio, s_threshold = 50, step_length = 20,
trend = "stable", first_pDate=NULL,
slope = NULL, ..., show.plot=FALSE, show.data=FALSE)
```

Arguments

n_events	(integer) Number of points (events) to simulate. Default: 1000. A vector of integer values can be supplied, in the format c(a1,a2,....), where a1, a2, ... represent different values.
start_date	The start date of temporal pattern. The date should be in the format "yyyy-mm-dd". The GTP will usually covers a 1-year period.
poly	(An sf or S4 object) A polygon shapefile within which event origins are to be situated.
n_origin	(an integer) Number of locations from which the walkers originate. Default:50.
resistance_feat	(An S4 object) Optional shapefile representing spaces across landscape within which event
field	A number in the range of [0-1] (i.e. resistance values) to assign to all features covered by resistance_feat; or the name of a numeric field to extract such resistance values for different feature classes. The resistance value 0 and 1 indicate the lowest and the highest restrictions, respectively, to an event occuring within the space occupied by a feature. origins are not allowed. Default: NULL.
n_foci	(an integer) Value indicating the number of focal points amongst event origins. n_foci will usually be smaller than n_origin.
foci_separation	(an integer) A percentage value indicating indicating the nearness of focal points from one another. A 0 separation indicates that focal points are in close proximity of one another, while a 100 indicates focal points being evenly distributed across space.
conc_type	(string) Specifies the spatial pattern of non-focal origin (strengths) in relation to to their nearest focal origins. Value is either "nucleated" or "dispersed".
p_ratio	(an integer) The smaller of the two terms of a Pareto ratio. For example, a value of 20 implies a 20:80 Pareto ratio.
s_threshold	(numeric) Spatial threshold value. This is the spatial range within which a walker perceives it's environment at any instant. Default: 250 (in the same linear unit as the poly - polygon shapefile).

step_length	(numeric) A maximum step taken at a time by a walker from one point to the next.
trend	(string) Specify the trend direction of the GTP. Values are: "decreasing", "stable", and "increasing". Default is: "stable".
first_pDate	(in "yyyy-mm-dd" format). Date of the first seasonal peak of the time series. Default value is NULL, in which a seasonal cycle of 180 days is utilized. That is, a first seasonal peak of 90 days.
slope	(string) Slope GTP trend if "increasing" or "decreasing" trend is specified. Values: "gentle" or "steep". Default value is NULL (i.e., for stable trend).
...	additional arguments to pass from gtp, walker and artif_spo functions.
show.plot	(logical) Shows GTP. Default is FALSE.
show.data	(TRUE or FALSE) To show the output data. Default is FALSE.

Details

Generate spatiotemporal point pattern based on the actions of specified 'walkers' moving across a landscape. Both the walkers and the landscape are configured arbitrarily (in accordance with the users (expert) knowledge of the domain in question.

Value

Returns a list of artificial spatiotemporal point patterns.

Examples

```
## Not run:
data(camden_boundary)
data(landuse)
artif_stpp <- psim_artif(n_events=200, start_date = "2021-01-01",
poly=camden_boundary, n_origin=50, resistance_feat = landuse,
field = "rValue1",
n_foci=5, foci_separation = 10, conc_type = "dispersed",
p_ratio = 20, s_threshold = 50, step_length = 20,
trend = "stable", first_pDate=NULL,
slope = NULL,show.plot=FALSE, show.data=FALSE)

## End(Not run)
```

---

psim_real *Stpp from real (sample) origins*

Description

Generate spatiotemporal point pattern from origins sampled based on real dataset.

Usage

```
psim_real(n_events, ppt, start_date = NULL, poly = NULL,
s_threshold = NULL, step_length = 20, n_origin=50,
resistance_feat, field=NA,
p_ratio=20, crsys = NULL)
```

Arguments

n_events	(integer) Number of points (events) to simulate. Default: 1000. A vector of integer values can be supplied, in the format c(a1,a2,....), where a1, a2, ... represent different values.
ppt	A 3-column matrix or list containing x - eastings, y - northing, and t - time of occurrence (in the format: 'yyyy-mm-dd').
start_date	Specifies the start date of the sample data provided (format: yyyy-mm-dd). If NULL, the earliest date of the t field of ppt is utilized. The end date is automatically set as the 365th day from the start date.
poly	(An sf or S4 object) Spatial (administrative) boundary covering the area under study. The default is NULL, in which an arbitrary boundary is drawn to cover the spatial extent of the data. The projection system of poly is assume for ppt, therefore, a user needs to ensure that both poly and ppt(-xy cordinates) are in the same reference system for accurate result.
s_threshold	(numeric) Spatial range from the origin within which a walker re-generate events. Default: NULL, in which the value is automatically estimated from the sample data (i.e., ppt).
step_length	(numeric) A maximum step taken at a time by a walker from one point to the next.
n_origin	(an integer) Number of locations from which the walkers originate. Default:50.The value has largest impacts on the computational time.
resistance_feat	(An S4 object) Optional shapefile representing spaces across landscape within which events are prohibited.
field	A number in the range of [0-1] (i.e. resistance values) to assign to all features covered by resistance_feat; or the name of a numeric field to extract such resistance values for different feature classes. The resistance value 0 and 1 indicate the lowest and the highest restrictions, respectively, to an event occuring within the space occupied by a feature. origins are not allowed. Default: NULL.
p_ratio	(an integer) The smaller of the two terms of a Pareto ratio. For example, a value of 20 implies a 20:80 Pareto ratio.
crsys	(string) The EPSG projection code that defines the xy coordinates (of ppt). This will be utilized if poly argument is NULL. See "http://spatialreference.org/" for the list of EPSG codes for different regions of the world. As an example, the EPSG code for the British National Grid projection system is: "EPSG:27700".

Details

Generate spatiotemporal point pattern based on the actions of specified 'walkers' moving across a landscape.The walkers and the landscape are configured based on spatiotemporal information learnt from real sample datasets.

References

Davies, T.M. and Hazelton, M.L. (2010), Adaptive kernel estimation of spatial relative risk, Statistics in Medicine, 29(23) 2423-2437. Terrell, G.R. (1990), The maximal smoothing principle in density estimation, Journal of the American Statistical Association, 85, 470-477.

Examples

```
## Not run:
data(camden_theft)
#specify the proportion of full data to use
sample_size <- 0.2
set.seed(1000)
dat_sample <- camden_theft[sample(1:nrow(camden_theft),
round((sample_size * nrow(camden_theft)), digits=0),
replace=FALSE),]
#plot(dat_sample$x, dat_sample$y) #preview
result <- psim_real(n_events=2000, ppt=dat_sample,
start_date = NULL, poly = NULL, s_threshold = NULL,
step_length = 20, n_origin=50, resistance_feat, field=NA,
p_ratio=20, crsys = "EPSG:27700")

## End(Not run)
```

---

p_prob *Pareto (probability) distribution*

Description

Generates an `n` probability values in accordance with a specified Pareto ratio.

Usage

```
p_prob(n,  p_ratio = 20)
```

Arguments

`n`	(an integer) Number of data points.
`p_ratio`	(an integer) The smaller of the terms of a Pareto ratio. For instance, for a `20:80` ratio, `p_ratio` is `20`. Default value is set as `20`. Acceptable `p_ratio` values are: (`5,10,20,30,40`).

Details

Produces a list probability values based on the specified Pareto ratio. Each ratio term is assigned the proportion of area under an exponential curve that is equal to the value of the other ratio term. For example, for a `20:80` ratio, 20% of data points covers 80% of areas under the curve, and vice versa.

Value

Returns a vector of probability values

Examples

```
p_prob(n = 15,  p_ratio = 20)
```

`space_restriction` *Space restriction map (raster)*

Description

Build a space restriction map from one or more shapefiles. A space restriction map (raster) shows the restrictions (to event occurences) across space. Function builds on raster- and SimRIv-packages.

Usage

```
space_restriction(shp, baseMap, res, binary = is.na(field),
field = NA, background = 1)
```

Arguments

`shp`	polygon shapefile object.
`baseMap`	if provided, a raster onto which to stack the next rasterized shapefile.
`res`	the desired pixel resolution of the raster to be created, when baseMap is not provided.
`binary`	if TRUE, the shapefile will be rasterized so that any feature is assigned a value of `0`, and the background is assigned `1`.
`field`	A number in the range of `[0-1]` (i.e. resistance values) to assign to all features covered by `shp`; or the name of a numeric field to extract such resistance values for different feature classes. The resistance value `0` and `1` indicate the lowest and the highest restrictions, respectively, to an event occuring within the space occupied by a feature.
`background`	the value in the range 0 and 1 to assign to all pixels that are not covered by any shapefile feature.

Details

Help to create a complete space restriction map with cell values ranging from 0(lowest) and 1(highest). All other area not covered by any features is assigned the value of `background`. When stacking additional features to existing `baseMap`, only the areas covered by features are updated, while the remaining areas retain the original values of `baseMap`.

Value

Returns a raster map

References

1. Paul Murrell (2019). rasterize: Rasterize Graphical Output. R package version 0.1. https://CRAN.R-project.org/package=rasterize
2. Quaglietta L, Porto M (2019). *SiMRiv: Individual-Based, Spatially-Explicit Simulation and Analysis of Multi-State Movements in River Networks and Heterogeneous Landscapes*. R package version 1.0.4, <URL: https://CRAN.R-project.org/package=SiMRiv>.

Examples

```
data(camden_boundary)
restrct_space <- space_restriction(shp = camden_boundary,
res = 20, binary = TRUE)
#plot the result
#plot(restrct_space)
#Setting 'restrct_space' raster as basemap, the landuse
#map can now be stacked onto the basemap as follows:
data(cam_landuse)
restrct_Landuse <- space_restriction(shp = landuse,
baseMap = restrct_space,
res = 20, field = "rValues2", background = 1)
#plot(restrct_Landuse)
```

`stp_learner` *Learning spatiotemporal properties*

Description

Learns both spatial and temporal properties of a real sample dataset.

Usage

```
stp_learner(ppt, start_date = NULL, poly = NULL,
n_origin=50, p_ratio, gridSize = 150,
crsys = NULL, show.plot = FALSE)
```

Arguments

`ppt`	A 3-column matrix or list containing `x` - eastings, `y` - northing, and `t` - time of occurrence (in the format: 'yyyy-mm-dd').
`start_date`	Specifies the start date of the sample data provided (format: `yyyy-mm-dd`). If `NULL`, the earliest date of the `t` field of `ppt` is utilized. The end date is automatically set as the 365th day from the start date.
`poly`	(An sf or S4 object) Spatial (administrative) boundary covering the area under study. The default is `NULL`, in which an arbitrary boundary is drawn to cover the spatial extent of the data. The projection system of `poly` is assume for `ppt`, therefore, a user needs to ensure that both `poly` and `ppt`(-xy cordinates) are in the same reference system for accurate result.
`n_origin`	(an integer) Number of locations from which the walkers originate. Default:`50`.
`p_ratio`	(an integer) The smaller of the two terms of a Pareto ratio. For example, a value of `20` implies a `20:80` Pareto ratio.
`gridSize`	(an integer) The size of square grid for discretizing the entire space. Default is: `150`.
`crsys`	(string) The EPSG projection code that defines the xy coordinates (of `ppt`). This will be utilized if `poly` argument is `NULL`. See "http://spatialreference.org/" for the list of EPSG codes for different regions of the world. As an example, the EPSG code for the British National Grid projection system is: `"EPSG:27700"`.
`show.plot`	(TRUE or FALSE) Whether to show some displays.

Details

Returns an object of the class `real_spo`, storing details of the learnt spatiotemporal properties of the sample data.

Examples

```
data(camden_theft)
#specify the proportion of full data to use
sample_size <- 0.2
set.seed(1000)
dat_sample <- camden_theft[sample(1:nrow(camden_theft),
round((sample_size * nrow(camden_theft)), digits=0),
replace=FALSE),]
#plot(dat_sample$x, dat_sample$y) #preview
stp_learner(dat_sample,
start_date = NULL, poly = NULL, n_origin=50,
p_ratio=20, gridSize = 150, crsys = "EPSG:27700",
show.plot = FALSE)
```

---

walker *A landscape walker*

Description

A dynamic object capable of walking across any landscape (constraint or unconstraint), in line with an in-built transition matrix, and specified spatial and temporal properties. The transition matrix defines a two-state status, namely; the `exploration` and the `action` states. See vignette for details.

Usage

```
walker(n = 5, s_threshold = 250, step_length = 20,
poly = NULL, resistance_feat=NULL, field = NA, coords=c(0,0),
pt_itx = TRUE, show.plot = FALSE)
```

Arguments

`n`	(integer) Number of events to be generated by a walker.
`s_threshold`	(numeric) Spatial threshold of a walker. This is the spatial range within which events are re-generated (or repeated) by a walker. Default: `250` (in the unit as the polygon shapefile (below))
`step_length`	(numeric) A maximum step taken at a time by a walker from one point to the next.
`poly`	(An sf or S4 object) A polygon shapefile within which event origins are to be generated.
`resistance_feat`	(An S4 object) Optional shapefile representing spaces across landscape within which event origins are not allowed. Default: `NULL`.
`field`	A number in the range of `[0-1]` (i.e. resistance values) to assign to all features covered by `shp`; or the name of a numeric field to extract such resistance values for different feature classes. The resistance value `0` and `1` indicate the lowest and the highest restrictions, respectively, to an event occuring within the space occupied by a feature.

coords	a vector of the form c(x, y) giving the initial coordinates of a walker (i.e., coordinates of origins). Default value is `c(0,0)` for an arbitrary square space.
pt_itx	(logical) Check whether any of the specified initial origin coordinates falls outside the boundary. Default: `TRUE`.
show.plot	(TRUE or False) To show the time series plot. Default is `FALSE`.

Details

Walks freely in all directions in accordance with a transition matrix, but avoids obstacles (i.e., the `resistance_feat`, if provided) along the way.

Value

Returns a trace of walker's path, and the corresponding events.

References

#https://google.co.uk

Examples

```
data(camden_boundary)
path <- walker(n = 5, s_threshold = 250, step_length = 20,
poly=camden_boundary, resistance_feat=NULL, field = NA,
coords = c(0,0), pt_itx = TRUE, show.plot = FALSE)
#plot(path)
```

---

xyt_data *Spatiotemporal point data*

Description

A spatiotemporal point data of a part of San Francisco City, California, US

Usage

```
xyt_data
```

Format

A matrix containing three variables

- x: x coordinate
- y: y coordinate
- t: t time

Index