

Package: stamp (via r-universe)

September 17, 2024

Type Package

Title Spatial Temporal Analysis of Moving Polygons

Version 0.3.1

Description Perform spatial temporal analysis of moving polygons; a longstanding analysis problem in Geographic Information Systems. Facilitates directional analysis, distance analysis, and some other simple functionality for examining spatial-temporal patterns of moving polygons.

URL <https://github.com/jedalong/stamp>

Depends R (>= 4.0)

License GPL-3

Imports sf, spdep, dplyr, graphics, grDevices, rlang, lwgeom, geosphere

RoxygenNote 7.2.3

Encoding UTF-8

Repository <https://jedalong.r-universe.dev>

RemoteUrl <https://github.com/jedalong/stamp>

RemoteRef HEAD

RemoteSha 2be4013f5107119408a6d6ddcebb585c7064a7af

Contents

stampr-package	2
eyeshp	2
fire1	3
fire2	4
glob.change	4
katrina	5
mpb	6
stamp	6
stamp.direction	8

stamp.distance	10
stamp.group.summary	11
stamp.map	12
stamp.multichange	13
stamp.stgroup.summary	14

Index	15
--------------	-----------

stampr-package	<i>stampr: Spatial Temporal Analysis of Moving Polygons</i>
----------------	---

Description

The Package `stampr` provides tools for performing spatial temporal analysis of moving polygons. These tools allow the calculation of directional relationships, distance relations, and other basic functionality, such as global change metrics. More details about each of these functions can be found in its help documentation.

Details

`stampr`'s functions utilize the `sf` objects (as of version 0.3.0) from the package `sf`. Polygon relationships are still understudied in the field of geographic information science, but hopefully `stampr` can provide users with a platform for new developments and applied research looking at interesting geographical phenomena.

Author(s)

Jed Long and Colin Robertson

References

Robertson, C., Nelson, T., Boots, B., and Wulder, M. (2007) STAMP: Spatial-temporal analysis of moving polygons *Journal of Geographical Systems*, 9:207-227. Long, J., Robertson, C., Nelson, T. (2018) `stampr`: Spatial-Temporal Analysis of Moving Polygons in R *Journal of Statistical Software. Code Snippets*, 84(1), 1–19.

eyeshp	<i>Hurricane Katrina eye point dataset</i>
--------	--

Description

A dataset containing points representing the eye of Hurricane Katrina centroid from 21:00 26-AUG-2005 to 21:00 29-AUG-2005. Polygon contours were extracted from the US NOAA H*Wind product, downloadable from: https://www.aoml.noaa.gov/hrd/data_sub/wind.html

Format

A sf object with 33 records of the eye location of Hurricane Katrina, every 3 hrs, from 21:00 25-AUG-2005 to 21:00 29-AUG-2005. The date and time of each polygon is recorded in the column DateTime.

Details

The eyeshp dataset contains points that were derived from the raw NOAA H*Wind data. The data is included here to provide a point-data comparison to the data in the katrina dataset which is polygon data

Examples

```
data(eyeshp)
plot(eyeshp)
```

fire1	<i>Forest Fire dataset</i>
-------	----------------------------

Description

A dataset containing fake forest fire polygons representing the movement of the forest fire from T1 (fire1) to T2 (fire2). The data is provided purely for demonstration purposes.

Format

fire1 — a sf object with polygons representing the location of forest fire.

Source

Simulated data

Examples

```
data(fire1)
plot(fire1)

data(fire2)
plot(fire2)

## Not run:
library(mapview)
mapview(fire1) + mapview(fire2)

## End(Not run)
```

fire2	<i>Forest Fire dataset</i>
-------	----------------------------

Description

see fire1

Format

fire2 — a sf object

glob.change	<i>glob.change</i>
-------------	--------------------

Description

The function `glob.change` computes a set of three global change metrics for comparison between two polygon sets. These metrics are outlined in Robertson et al. (2007; Table 4).

Usage

```
glob.change(T1, T2)
```

Arguments

T1 a sf object of polygons from time 1.
 T2 a sf object of polygons from time 2.

Details

`glob.change` computes three change metrics, detailed below, that can be used to quantify changes between two polygon sets:

NumRatio – ratio between the number of polygons in T2 and T1;

$$\text{NumRatio} = \frac{\#(T1)}{\#(T2)}$$

AreaRatio – ratio between the areas of polygons in T2 and T1;

$$\text{AreaRatio} = \frac{A(T2)}{A(T1)}$$

AvgAreaRatio – ratio between the AreaRatio and NumRatio;

$$\text{AvgAreaRatio} = \frac{\text{AreaRatio}}{\text{NumRatio}} = \frac{\frac{A(T2)}{A(T1)}}{\frac{\#(T1)}{\#(T2)}}$$

Value

A list object with three elements - Results for the NumRatio, AreaRatio, and AvgAreaRatio metrics.

katrina

Hurricane Katrina polygons dataset

Description

A dataset containing polygons representing the movement of Hurricane Katrina from 21:00 26-AUG-2005 to 21:00 29-AUG-2005. Polygon contours were extracted from the US NOAA H*Wind product, downloadable from: https://www.aoml.noaa.gov/hrd/data_sub/wind.html

Format

A sf object with 33 records of the location of Hurricane Katrina, every 3 hrs, from 21:00 25-AUG-2005 to 21:00 29-AUG-2005. The date and time of each polygon is recorded in the column DateTime.

Details

The katrina dataset contains polygons that were derived from the raw NOAA H*Wind data. The 39 mph isotach (contour of equal wind speed) was used to delineate, as a spatial polygon, the extent of Hurricane Katrina at a given time. Polygons were derived at 3 hr intervals; which means there are 33 different time points in the dataset.

Source

https://www.aoml.noaa.gov/hrd/data_sub/wind.html

References

Powell, M.D., Murillo, S., Dodge, P., Uhlhorn, E., Gamache, J., Cardone, V., Cox, A., Otero, S., Carrasco, N., Annane, B., St. Fleur, R. (2010) Reconstruction of Hurricane Katrina's wind fields for storm surge and wave hindcasting. *Ocean Engineering*, 37, 26-36.

Powell, M.D., Houston, S.H. (1998) The HMD real-time hurricane wind analysis system. *Journal of Wind Engineering and Industrial Aerodynamics*, 77/78, 53-64.

Examples

```
data(katrina)
plot(katrina['Id'])
```

mpb

MPB dataset

Description

A dataset containing polygons representing the location of mountain pine beetle hotspot polygons in Morice Forest District, British Columbia, Canada.

Format

mpb — a sf object with 711 mountain pine beetle hotspot polygons that occurred over eight years. The temporal indicator is the TGROUP column. Another variable REGION indicates whether the hotspot was in the northern or southern regions, which experienced mostly independent outbreaks.

Details

These data were derived from helicopter-based GPS surveys during early years of large mountain pine beetle outbreak in Western Canada.

Source

Data obtained from Trisalyn Nelson (ASU)

References

Nelson TA, Boots B, Wulder MA, Carroll AL. Environmental characteristics of mountain pine beetle infestation hot spots. *Journal of Ecosystems and Management*. 2007 Mar 14;8(1).

Examples

```
data(mpb)
plot(mpb['TGROUP'])
```

stamp*Spatial temporal analysis of moving polygons*

Description

This function generates a sf polygons object that can be used for spatial temporal analysis of moving polygons as described in the paper Robertson et al. (2007).

Usage

```
stamp(T1, T2, dc = 0, direction = FALSE, distance = FALSE, ...)
```

Arguments

T1	a sf polygons object of polygons from time 1.
T2	a sf polygons object of polygons from time 2.
dc	spatial distance threshold for determining groupings (see Details) in appropriate units.
direction	logical, whether or not to perform directional analysis. See documentation for stamp.direction for further details.
distance	logical, whether or not to perform distance analysis. See documentation for stamp.distance for further details.
...	additional parameters to be passed to functions if direction, or distance are set to TRUE.

Details

The stamp function can be used to perform spatial temporal analysis of moving polygons (STAMP) as outlined in the paper by Robertson et al., (2007). Polygon movement "groups" are delineated based on polygon connectedness defined by the distance threshold dc. That is, if polygon boundaries (in T1 or T2) are within distance dc of one another they will be designated to the same group. STAMP events are reported at four levels of increasing complexity:

LEV1 – disappearance (DISA), stable (STBL), and generation (GENA);

LEV2 – disappearance (DISA), contraction (CONT), stable (STBL), expansion (EXPN), and generation (GENR);

LEV3 – disappearance (DISA), T1 displacement (DISP1), convergence (CONV), concentration (CONC), contraction (CONT), stable (STBL), expansion (EXP), fragmentation (FRAG), divergence (DIV), T2 displacement (DISP2), and generation (GENR);

LEV4 – LEV4 is different from other levels. It is used to identify those groups where union (UNION), division (DIVISION), and both union and division (BOTH) events occur. These events occur when there are more than one stable event in a group. Groups with one or no stable events receive an NA value for LEV4.

See Robertson et al. (2007; especially Figure 1) for complete descriptions of all STAMP movement event types.

Note also that there must be a unique ID of each polygon, the function uses the row.names of the polygon objects. Modify the row.names accordingly if you wish to use an alternative ID label.

Value

This function returns a sf polygons object with the following data columns:

ID1	Polygon ID from T1 polygons; NA if it did not exist,
ID2	Polygon ID from T2 polygons; NA if it did not exist,
LEV1	Level 1 STAMP designation,
LEV2	Level 2 STAMP designation,
LEV3	Level 3 STAMP designation,
LEV4	Level 4 STAMP designation,
GROUP	Group ID signifying group membership,

AREA	Polygon area in appropriate areal units,
-	(optional) Additional columns from directional analysis if <code>direction = TRUE</code> ,
-	(optional) Additional columns from distance analysis if <code>distance = TRUE</code> ,

References

Robertson, C., Nelson, T., Boots, B., and Wulder, M. (2007) STAMP: Spatial-temporal analysis of moving polygons. *Journal of Geographical Systems*, 9:207-227.

See Also

stamp.direction stamp.distance stamp.map stamp.group.summary

stamp.direction	<i>Perform polygon directional analysis</i>
-----------------	---

Description

stamp.direction facilitates polygon directional analysis using a variety of methods.

Usage

```
stamp.direction(stmp, dir.mode = "CentroidAngle", ndir = 4, group = FALSE)
```

Arguments

stmp	a sf object generated from the stamp function.
dir.mode	a character item identifying which directional relations method is to be used. See Details for information on each individual method.
ndir	(optional) parameter identifying the number of directions to be computed. See individual method Details for appropriate usage.
group	(optional) a logical value identifying whether direction should be computed on groups or individual event polygons (only used with CentroidAngle method).

Details

The stamp.direction function can be used to facilitate directional analysis on output stamp.obj objects from function stamp. Currently, four directional analysis methods are available:

- "CentroidAngle" – The centroid angle is simply the angle between the centroids of two polygons. The centroid angle method is computed on STAMP objects by first grouping all T1 polygons (by STAMP group) and computing their centroid. Then, the angle from each T1 group centroid, to the centroid of each STAMP event within the group is calculated. Centroid angles are recorded in degrees, with North having a value of 0, East 90, and so on. "CentroidAngle" ignores the ndir parameter.

- "ConeModel" – The cone model method calculates areas of STAMP event polygons within cones radiating from the centroid of the origin polygon. The cone model method first computes the centroid of all T1 polygons in a STAMP grouping. It then computes `ndir` equally spaced cones radiating outward from the T1 centroid. The first cone is always centered on North, but there can be any number of cones. The area of each STAMP event, in each cone (specifying direction), is then calculated. See Peuquet and Zhang (1987) for more detailed information
- "MBRModel" – The minimum bounding rectangle (MBR) method first computes the MBR for all T1 events in a STAMP grouping. Then the lines of four edges of the MBR are extended outwards to infinity creating sections for the eight cardinal directions around the MBR, along with the MBR itself. The area of each stamp event within each of the nine sections is then computed. See Skiadopoulos et al. (2005) for more detailed information. "MBRModel" ignores the `ndir` parameter.
- "ModConeModel" – The modified cone model first computes the centroid of the T1 events. Then `ndir = 4` or `8` cones are created outward from this centroid to the minimum bounding rectangle of the entire grouping. As described by Robertson et al. (2007) this approach is more accommodating to polygon groups that are irregular in size or shape. The modified cone model method first computes the centroid of all T1 polygons in a STAMP grouping. It then computes the bounding box of ALL events in a STAMP grouping. Then, `ndir=4` or `8` cones are computed. In the case of `ndir=4`, cones radiate from the T1 centroid to the four corners of the bounding box. The result of the modified cone model method is that the cones are not equally spaced, but tailored to the individual STAMP groupings shape. See Robertson et al. (2007) for more detailed information. NOTE: This function has been altered slightly as of `stampr v 0.3`.

As of V 0.3 all operations are conducted using `sf` object classes, all directional (azimuth) and area calculations use WGS84.

Value

Appends the input `stamp` object with appropriate columns for the directional analysis chosen, if `dir.mode` is:

"CentroidAngle"

A single column with centroid angle results, in degrees (North = 0 degrees). If `group=TRUE` then values are identical for all event polygons in the group.

"ConeModel"

`ndir` new columns with the area (m²) of the STAMP event in each direction, named appropriately (e.g., as `DIR45`, where 45 refers to the mid-point of that directional cone).

"MBRModel"

9 new columns with the area (m²) of the STAMP event in each direction, named appropriately as, for example, "`MBR_SW`", "`MBR_S`", ... etc.

"ModConeModel"

`ndir` new columns with the area (m²) of the STAMP event in each direction, named appropriately as, for example, "`MC4_N`", "`MC8_SE`", ... etc.

Note: STAMP events that are singular (i.e., only 1 polygon in the group) will have NA's from directional analysis.

References

Robertson, C., Nelson, T., Boots, B., and Wulder, M. (2007) STAMP: Spatial-temporal analysis of moving polygons. *Journal of Geographical Systems*, 9:207-227.

Peuquet, D., Zhang, C.X. (1987) An algorithm to determine the directional relationship between arbitrarily-shaped polygons in the plane. *Pattern Recognition*, 20:65-74.

Skiadopoulos, S. Giannoukos, C., Sarkas, N., Vassiliadis, P., Sellis, T., and Koubarakis, M. (2005) Computing and managing directional relations. *IEEE Transactions on Knowledge and Data Engineering*, 17:1610-1623.

See Also

stamp stamp.distance

stamp.distance	<i>stamp.distance</i>
----------------	-----------------------

Description

The function `stamp.distance` can be used to compute various measures of distance between polygon events and groups. In turn, distance measurements can be used to estimate the velocity of polygon movement.

Usage

```
stamp.distance(stmp, dist.mode = "Centroid", group = FALSE)
```

Arguments

<code>stmp</code>	a <code>sf</code> object generated from the <code>stamp</code> function.
<code>dist.mode</code>	Character determining the method by which polygon distances are computed. If "Centroid" then the centroid distance is calculated, if "Hausdorff" then the discrete Hausdorff distance is calculated; see <code>Details</code> .
<code>group</code>	logical indicating whether distances should be computed from the T1 polygon to each individual stamp event (<code>group = FALSE</code> – the default), or whether T2 polygons should combined (through a spatial union) in order to compute the measure of distance for each stamp group (<code>group = TRUE</code>)

Details

`stamp.distance` computes distance between polygon sets based on either centroid or Hausdorff distance calculations. Centroid distance is simply the distance from the centroid of all T1 polygons (combined) to each stamp event (`group = FALSE`), or to the union of all T2 polygons within a group (`group = TRUE`), in the second case, all events within a group are given an identical distance value.

The Hausdorff distance calculation uses the Hausdorff distance, as programmed in the function `st_distance`. A value of `par = 0.1` is used to increase the precision of this measurement – see `help(st_distance)`. The returned distance is then the Hausdorff distance of all T1 polygons (combined) to each stamp event (`group = FALSE`), or to the union of all T2 polygons within a group (`group = TRUE`), in the second case, all events within a group are given an identical distance value. All distance calculations are computed in meters using the geographical projection WGS84.

Value

Appropriately named columns (e.g., `CENDIST` or `HAUSDIST`) in the stamp `sf` object. Distances are in meters.

References

Hausdorff Distance: https://en.wikipedia.org/wiki/Hausdorff_distance

See Also

`stamp stamp.direction`

`stamp.group.summary` *Compile stamp summary statistics by group*

Description

The function `stamp.group.summary` compiles summary statistics for each STAMP grouping. Specifically, it computes the area of each STAMP event type (e.g., generation, expansion, etc.) within each grouping. It also computes the number of events belonging to each event type.

Usage

```
stamp.group.summary(stmp, area = TRUE, count = TRUE)
```

Arguments

<code>stmp</code>	a <code>sf</code> object generated from the stamp function.
<code>area</code>	logical, whether or not to compute the STAMP event areas.
<code>count</code>	logical, whether or not to compute the count of STAMP events within each group.

Details

`stamp.group.summary` computes area and count summary statistics of STAMP output. Note that if both `area` and `count` are set to `FALSE`, `stamp.group.summary` returns a `data.frame` with just the group IDs as the only column.

Value

A data.frame where rows are stamp groups and columns correspond to the STAMP event types (ID, areas, and counts).

stamp.map	<i>Mapping (plotting) functionality for stamp output</i>
-----------	--

Description

This function maps STAMP output for visual assessment of STAMP events and groupings. Choice of which aspect of the stamp output to be visualized is controlled by passing the column name to the stamp.map function.

Usage

```
stamp.map(stmp, by = "LEV1", ...)
```

Arguments

stmp	output from the stamp function, i.e., a (sf object).
by	tells the function which attribute to visualize, one of "LEV1", "LEV2", "LEV3", "LEV4", or "GROUP"
...	additional parameters to be passed to the sf plot function

Details

The stamp.map function can be used to visualize any of the stamp event designation levels (e.g., "LEV1", "LEV2", "LEV3", "LEV4", or the STAMP groupings (based off of parameter dc in the stamp function).

Value

stamp.map returns a map of the stamp output using the plot.sf functionality. It implements a pre-defined coloring scheme.

See Also

```
stamp data("fire1") data("fire2") ch <- stamp(fire1, fire2, dc=1, direction=FALSE, distance=FALSE)
stamp.map(ch, "LEV1") stamp.map(ch, "LEV2") stamp.map(ch, "LEV3") stamp.map(ch, "LEV4")
```

stamp.multichange *run stamp function for multiple years of polygons at once*

Description

The function `stamp.multichange` is a wrapper function that makes multiple calls to the `stamp` function to ease spatial-temporal analysis of multiple years of polygon data

Usage

```
stamp.multichange(polys, changeByRow = TRUE, changeField = "", ...)
```

Arguments

<code>polys</code>	a sf polygon object with 2+ years of data to run through the <code>stamp</code> function.
<code>changeByRow</code>	logical, whether or not each time period is a separate unique row of data (e.g., as per the <code>katrina</code> data)
<code>changeField</code>	string, name of the field which contains time period if <code>changeByRow</code> is FALSE
<code>...</code>	list of parameter values to provide to the <code>stamp</code> function

Details

`stamp.multichange` is a simple wrapper function for the `stamp` function. The two options for data structure are those in the `katrina` data, where each time period is a row, and rows are time-ordered, and the structure of the `mpb` data, where time period is specified by a column. Time periods should be ordered from 1 through T.

Value

A sf object which includes all outputs from the calls to the `stamp` function. If there are T time periods, there will be T-1 time periods in the resulting sf object.

See Also

`stamp.stgroup.summary`

Examples

```
## Not run:
## NOT RUN ##
data("katrina")
ch <- stamp.multichange(katrina, changeByRow = TRUE, dc = 0, distance = TRUE, direction = FALSE)
STGroup <- stamp.stgroup.summary(ch)
head(STGroup)

## End(Not run)
```

stamp.stgroup.summary *Compile stamp summary statistics by space-time group*

Description

The function `stamp.stgroup.summary` compiles summary statistics for each STAMP space-time grouping. Specifically, it computes the area of each STAMP event type (e.g., generation, expansion, etc.) within each grouping. It also computes the number of events belonging to each event type.

Usage

```
stamp.stgroup.summary(stmp, area = TRUE, count = TRUE)
```

Arguments

<code>stmp</code>	a <code>sf</code> polygon object generated from the <code>stamp.multichange</code> function.
<code>area</code>	logical, whether or not to compute the STAMP event areas.
<code>count</code>	logical, whether or not to compute the count of STAMP events within each group.

Details

`stamp.stgroup.summary` computes area and count summary statistics of STAMP output derived from multi-time analysis using `stamp.multichange`. `stamp.multichange` is just a wrapper function for applying `stamp` to multiple time periods in the same dataset. Note that if both `area` and `count` are set to `FALSE`, `stamp.stgroup.summary` returns a `data.frame` with just the `stgroup` IDs as the only column.

Value

A `data.frame` where rows are stamp groups and columns correspond to the STAMP event types (ID, areas, and counts).

See Also

`stamp.multichange`

Examples

```
## Not run:
##NOT RUN##
library(sf)
data("katrina")
ch <- stamp.multichange(katrina, changeByRow = TRUE, dc = 0, distance = TRUE, direction = FALSE)
STGroup <- stamp.stgroup.summary(ch)
head(STGroup)

## End(Not run)
```

Index

* datasets

- eyeshp, 2
- fire1, 3
- fire2, 4
- katrina, 5
- mpb, 6

* metrics

- glob.change, 4

* plot

- stamp.map, 12

* stamp

- stamp, 6
- stamp.direction, 8
- stamp.distance, 10
- stamp.group.summary, 11
- stamp.map, 12
- stamp.multichange, 13
- stamp.stgroup.summary, 14

eyeshp, 2

fire1, 3

fire2, 4

glob.change, 4

katrina, 5

mpb, 6

stamp, 6

stamp.direction, 8

stamp.distance, 10

stamp.group.summary, 11

stamp.map, 12

stamp.multichange, 13

stamp.stgroup.summary, 14

stampr-package, 2