# Package 'tlocoh.dev'

September 20, 2015

**Type** Package

**Title** Development package for tlocoh

**Version** 1.31.0

**Date** 2015-09-20

**Author** Andy Lyons

**Maintainer** Andy Lyons <tlocoh@gmail.com>

**Depends** R (>= 3.0.0),
    sp,
    tlocoh,
    shiny

**Imports** pbapply

**Suggests** maptools,
    rgdal,
    move,
    png,
    raster,
    gpclib,
    XML,
    leaflet,
    dismo

**Description** Development package for tlocoh (i.e. functions still being
    developed).

**License** GPL-2

**URL** http://tlocoh.r-forge.r-project.org

## R topics documented:

---

draw.polygons                    *Draw polygons on the plot window*

---

## Description

Returns a SpatialPolygonsDataFrame object containing polygons drawn with the mouse on a plot

## Usage

```
draw.polygons(n, draw.reg = TRUE, col = NULL, alpha = 0.5,
  prompt.labels = TRUE, ID = NULL, proj4string = CRS(as.character(NA)))
```

## Arguments

| | |
|---|---|
| n | Number of polygons to draw. |
| draw.reg | Whether to draw the polygons on the plot. TRUE/FALSE. |
| col | Color values for the polygons to be drawn. Ignored if draw.reg=FALSE. |
| alpha | Transparency of the colors (0..1). Ignored if draw.reg=FALSE. |
| prompt.labels | Whether to prompt for a label after each polygon is drawn (TRUE/FALSE). See Details. |
| ID | A character vector of length n of the ID values for each polygon |
| proj4string | An object of class CRS containing the coordinate system of the drawn polygons |

## Details

This function involves 'digitizing' polygons by clicking with the mouse, so it requires a plot window to be active. The number of polygons to be drawn must be specified by the argument n. If prompt.labels=FALSE, default IDs will be constructed for each polygon.

proj4string can be used if the current plot window is displaying geographic data.

## Value

An object of class SpatialPolygonsDataFrame

| gridlayer | *Returns a square grid covering a bounding box If both mindim and cellsize are passed, cellsize trumps mindim* |
|---|---|

## Description

Returns a square grid covering a bounding box If both mindim and cellsize are passed, cellsize trumps mindim

## Usage

```
gridlayer(bbox, proj4string = CRS(as.character(NA)), mindim = 50,
  cellsize = NULL, plotme = FALSE, status = TRUE)
```

| hexlayer | *Returns a hexagonal grid covering a bounding box If both mindim and cellsize are passed, cellsize trumps mindim* |
|---|---|

## Description

Returns a hexagonal grid covering a bounding box If both mindim and cellsize are passed, cellsize trumps mindim

## Usage

```
hexlayer(bbox, proj4string = CRS(as.character(NA)), mindim = 50,
  cellsize = NULL, plotme = FALSE, status = TRUE)
```

| iso2raster | *Convert isopleths to raster* |
|---|---|

## Description

Converts isopleths in a SpatialPolygonsDataFrame object to a RasterLayer

## Usage

```
iso2raster(polys, raster = NULL, ext = NULL, dimSize = 100,
  cell.size = NULL, sf.cell.size = 2, ll.round = TRUE, status = TRUE,
  debug = FALSE)
```

## Arguments

| | |
|---|---|
| polys | A SpatialPolygonsDataFrame containing isopleths sorted by isopleth level (smallest to largest) |
| raster | A RasterStack object to be used to set the extent and resolution for the output raster. |
| ext | An extent object or NULL. Ignored if `raster` is passed. |
| dimSize | Numeric value used as the number of cells along the largest dimension of the data is numeric. Ignored if `raster` is passed. |
| cell.size | . Ignored if `raster` is passed. |
| sf.cell.size | . Ignored if `raster` is passed. |
| ll.round | Anchor the lower left coordinate of the raster extent to a multiple of the cell size. Ignored if `raster` is passed. |
| status | Show status messages and progress bar |
| debug | Show debugging info |

## Details

This presumes the SPDF contains isopleths ordered from lowest level to highest. The cell values of the resulting raster will sum up to the largest isopleth leve. In order for the resulting raster to sum to 1, the 100

---

| lhs.gmap | *Download background images for plots* |
|---|---|

---

## Description

Download background images for plots

## Usage

```
lhs.gmap(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, gmap = c("roadmap", "satellite", "hybrid", "terrain")[3],
  status = TRUE)
```

## Arguments

| | |
|---|---|
| lhs | A [LoCoH-hullset](#) object |
| id | The name(s) of individuals to extract |
| k | The k value of hullsets to extract |
| r | The r value of hullsets to extract |
| a | The a value of hullsets to extract |
| s | The s value of hullsets to extract |
| hs.names | The name(s) of saved hullsets to extract |
| gmap | The type of image to download: `"roadmap"`, `"satellite"`, `"hybrid"`, or `"terrain"` |
| status | Show status messages, T/F |

**Details**

This function can be used to download images for plotting hullsets. By downloading them once and storing them as a variable, plotting is faster. One image is obtained for each id (individual).

**Value**

A list containing one element for each id in `lhs`. The list is of class `locoh.gmap`. Each element is another named list containing i) a raster (in the same projection as `lhs`, ii) a vector of color values, and iii) the type of image (e.g., hybrid, terrain, etc.).

**Examples**

```
## Not run:
require(raster)
require(dismo)
toni.bg <- lhs.gmap(toni.lhs, type="terrain")
plot(toni.lhs, iso=T, gmap=toni.bg)

## End(Not run)
```

---

lhs.iso.overlap            *Compute isopleth overlap*

---

**Description**

Computes the area of isopleth overlap

**Usage**

```
lhs.iso.overlap(lhs, id = NULL, k = NULL, r = NULL, a = NULL,
  s = NULL, hs.names = NULL, iso.level = NULL, hsnames_simplify = TRUE,
  status = TRUE)
```

**Arguments**

| | |
|---|---|
| lhs | A LoCoH-hullset object |
| id | The id(s) of the hullsets to compare |
| k | The k value of hullsets to compare |
| r | The r value of hullsets to compare |
| a | The a value of hullsets to compare |
| s | The s value of hullsets to compare |
| hs.names | The name(s) of saved hullsets to compare |
| iso.level | A numeric vector of the isopleth level(s) of interest |
| hsnames_simplify | |
| | If True, will simplify the hullset names to just the IDs for the row and column names of the overlap matrices returned (see Value) |
| status | Show messages. T/F |

## Details

This function computes the area intersection of isopleths among different hullsets. This might be done, for example, if the hullsets belong to different individuals, and you want to see which individuals share space. All pairs of hullsets in `lhs` will be compared, and all isopleth levels will be compared.

## Value

A list object with three named elements. The *spdf* contains a SpatialPolygonsDataFrame, with a data table that saves the areas of intersection expressed in map units and proportions of the isopleth area for each hullset. *overlap_area* contains a square matrix whose values are the area of intersection in map units. *overlap_area* contains a square matrix whose values are the areas of intersection expressed as proportions of the each the two isopleths.

## See Also

[isopleths](isopleths)

---

| lhs.iso.rast | *Convert isopleths to rasters* |
|---|---|

---

## Description

Adds rasterized version of isopleth(s) to a LoCoH-hullset object

## Usage

```
lhs.iso.rast(lhs, id = NULL, k = NULL, r = NULL, a = NULL, s = NULL,
  hs.names = NULL, sort.metric = NULL, iso.method = "pt.quantiles",
  raster = NULL, dimSize = 100, cell.size = NULL, sf.cell.size = 2,
  ll.round = TRUE, status = TRUE)
```

## Arguments

| | |
|---|---|
| lhs | A [LoCoH-hullset](LoCoH-hullset) object |
| id | The id(s) of the hullsets to create isopleths for |
| k | The k value of hullsets to create isopleths for |
| r | The r value of hullsets to create isopleths for |
| a | The a value of hullsets to create isopleths for |
| s | The s value of hullsets to create isopleths for |
| hs.names | The name(s) of saved hullsets to create isopleths for |
| sort.metric | The name(s) of hull metric(s) used to form isopleths that rasters should be created for |
| iso.method | The method(s) used to define isopleths that will be converted to raster |
| raster | A RasterLayer object that will be used to set the extent and cell size of the rasterized isopleth |
| dimSize | The number of cells along the largest dimension of the track. The according raster will be calculated internally. Default is 100. Ignored if `raster` is passed. |

| | |
|---|---|
| cell.size | The size of each square cell in map units. Ignored if `raster` is passed. |
| sf.cell.size | The number of significant figures to use if the cell size has be computed based on `dimSize`. Default=2. Ignored if `raster` is passed. |
| ll.round | Whether to round the lower left coordinate to the lowest multiple of `cell.size`. Ignored if `raster` is passed. (T/F) |
| status | Show status messages. T/F |

### Details

This will take exising isopleths and create raster versions of them.

### Value

A LoCoH-hullset object

### See Also

[lhs.iso.add](lhs.iso.add)

### Examples

```
## Not run:
lhs <- lhs.iso.add(lhs)
lhs <- lhs.iso.rast(lhs)

## End(Not run)
```

---

lhs.revisit.add            *Add revisitation hull metrics to a LoCoH-hullset object*

---

### Description

Computes revisitation rate based on a minimum and maximum time away period

### Usage

```
lhs.revisit.add(lhs, ta.min = NULL, ta.max = NULL, ta.cuts = NULL,
  status = TRUE)
```

### Arguments

| | |
|---|---|
| lhs | A LoCoH-hullset object |
| ta.min | Minimum value(s) for time away in seconds (numeric vector) |
| ta.max | Maximum value(s) for time away in seconds (numeric vector) |
| ta.cuts | A numeric vector of time values in seconds that define the time-away intervals |
| status | Show status messages. T/F |

## Details

ta.min and ta.max define the minimum and maximum period of time (in seconds) which must pass for another occurence in the hull to be considered a 'revisit'. They should be the same lengths. If ta.max is NULL, no upper bound will be set

ta.cuts is an alternative way of specifying the time-away intervals. The time values in ta.cuts will serve as the values between time away intervals. For example if ta.cuts = c(1000,4000,7000,10000), three time-away intervals will be examined: 1000 to 4000 seconds, 4000 to 7000 seconds, and 7000 to 10000 seconds. One way to get the values for ta.cuts is to plot the distribution of revisit times using lhs.plot.revisit, and then use the get.vals function to select time values between clusters of revisit times.

## Value

A LoCoH-hullset object

## See Also

lhs.plot.revisit, get.vals

---

| lhs.revisit.del | *Delete all revisitation hull metrics in a LoCoH-hullset object* |

---

## Description

Delete all revisitation hull metrics in a LoCoH-hullset object

## Usage

```
lhs.revisit.del(lhs, status = TRUE)
```

## Arguments

| lhs | A LoCoH-hullset object |
| status | Display summary, T/F |

## Value

A LoCoH-hullset object

## See Also

lhs.revisit.add

## lhs.selection       *Create a hullset selection object*

### Description

Create a hullset selection object

### Usage

```
lhs.selection(lhs, k = NULL, a = NULL, r = NULL)
```

### Arguments

| | |
|---|---|
| lhs | A LoCoH-hullset object |
| k | Optional initial value for *k* |
| a | Optional initial value for *a* |
| r | Optional initial value for *r* |

### Details

This will create a new hullset selection object of class locoh.selection. You can then feed this object into codelhs.shiny.select to visually select a hullset for each individual. This is typically done when you have create a hullset containing multiple hullsets for each individual, and you want to pick one per individual for the rest of the analysis.

### See Also

codelhs.shiny.select

## lhs.shiny.select       *Interactive selection of hullsets from a locoh-hullset*

### Description

Visually select one hullset per individual from a LoCoH-hullset

### Usage

```
lhs.shiny.select(lhs, selection = NULL, gmap = "none")
```

### Arguments

| | |
|---|---|
| lhs | A LoCoH-hullset object |
| selection | An object of class *locoh.selection* containing a named list (one per individual) of selected hullset parameter values |
| gmap | The name of a background image that will be downloaded from Google: "none", "roadmap", "satellite", "hybrid", or "terrain". May also be a object of type locoh.gmap, see Notes. |

**Value**

An object of class *locoh.selection* containing a list of selected hullset parameter value for each id (see Details)

with plots of isopleth area and isopleth edge:area ratio, and select the value of 'k', 'a', or 'r' that does the best job balancing over- and under-estimation.

This function can be useful when you have a locoh-hullset object with multiple hullsets for each individual over a range of *k* or *a* values, and you want to pick one hullset per individual for the rest of the analysis. Note the range of parameter values should be uniform.

This function requires using RStudio and the Shiny package. To return the selection, be sure to click the 'Save and Return' button in the shiny app. The object returned can be passed to the function again as the 'selection' argument.

To display an image from Google in the background, set gmap to `"roadmap"`, `"satellite"`, `"hybrid"`, or `"terrain"`. This requires an internet connection. You may also set gmap to an object of type `locoh.gmap`, so the image(s) don't have to be downloaded each time. See `lhs.gmap` (in the tlocoh.dev package).

**See Also**

codelhs.selection, codelhs.gmap

---

lhs.so.add                              *Add hull metrics for association analysis*

---

**Description**

Add hull metrics for association analysis

**Usage**

```
lhs.so.add(lhs, id = "all", hs2.id = "all", tbuff = 0, ivg = NULL,
  test = 0, skip.dups = TRUE, save.hso = TRUE, status = TRUE,
  piFUN = c("pIntersect", "pIntersectSat")[1])
```

**Arguments**

| | |
|---|---|
| lhs | A LoCoH-hullset object |
| id | A character vector of the hullset ids to compute metrics for. Can also be `'all'`. |
| hs2.id | A character vector of the hullset ids to use as the comparison hullsets. Can also be `'all'`. |
| tbuff | A temporal overlap threshhold (in seconds). See details. |
| ivg | The intervisit gap period used to collapse intersecting hulls into discrete visits, see details |
| test | A two-element numeric vector containing the number of hulls in hullset 1 and hullset 2 respectively to identify intersections |
| skip.dups | Skip duplicate hulls (faster) |
| save.hso | Whether to save the hull intersection list, T/F |
| status | Show status messages. T/F |
| piFUN | The function to use to identify which pairs of hulls intersect: 'pIntersect' or 'pIntersectSat' |

**Details**

This function computes hull metrics for the spatially overlapping hulls from two ids. Typically this would be used when you have hullset from two individuals (i.e., two animals) and you want to see the spatial and temporal patterns of shared space use.

You can impose a temporal overlap requirement as well by passing a value for tbuff. Two hulls will be considered spatially overlapping only if their parent points also were recorded within tbuff seconds of each other. This essentially produces metrics for spatially and temporally overlapping hulls.

Hullset metrics are computed for each pair of ids. Thus if a hullset has hulls for three unique ids, each hull will have spatial overlap metrics computed for each of the other two hullsets. You can narrow which id(s) to compute metrics for, and which hullset(s) to use as the comparison, with the id and hs2.id arguments.

Up to three spatial overlap metrics are computed. so.count is simply the number of hulls in hullset 2 that overlap. so.dtmin is the minimum amount of time (expressed in seconds) that passes between overlapping hulls. This reflects temporal partitioning of shared space - low values of so.dtmin suggest the two individuals don't mind being in the same area at the same time. so.nsv (number of separate visits) is similiar to so.count, but collapses overlapping hulls into discrete visits based on an intervisit gap period ivg. so.nsv is only computed if a value for ivg is passed.

pIntersect and pIntersectSat are two functions that identify which pairs of hulls actually intersect. Neither are terribly fast, but pIntersect appears to work faster than pIntersectSat.

**Value**

A [LoCoH-hullset](#) object

---

    `lhs.to.add`               *Compute hull metrics for temporally overlapping hulls*

---

**Description**

Compute hull metrics for pairs of temporally overlapping hulls of separate individuals

**Usage**

```
lhs.to.add(lhs, id = "all", hs2.id = "all", maxdt = "auto",
  save.hto = TRUE, status = TRUE)
```

**Arguments**

| | |
|---|---|
| lhs | A [LoCoH-hullset](#) object |
| id | A character vector of the hullset ids to compute metrics for. Can also be 'all'. |
| hs2.id | A character vector of the hullset ids to use as the comparison hullsets. Can also be 'all'. |
| maxdt | The maximum difference in time (in seconds) for two hulls to be considered 'overlapping' in time. Can also be 'auto', in which case half of the smallest of the two median sampling intervals will be used. |
| save.hto | Whether to save the list of hull indices that temporally overlap in the hullset, T/F |
| status | Show status messages, T/F |

**Details**

This will compute hull metrics for pairs of hulls from two individuals. This only works for a LoCoH-hullset object that contains hulls from two or more individuals (id's). For each hull for individual A, for example, it will identify the hulls in individual B that temporally overlap, and compute the mean centroid distance (hull metric name = to.mcd). By default *to.mcd* is computed for all pairs of individuals, but you can specify specific pairs by passing values for id and hs2.id.

This metric is used for association analysis. You can use this metric for example to look for spatial and temporal patterns in how close individuals get to each other. Other association metrics exist for spatially overlapping hulls (see lhs.so.add).

**Value**

A LoCoH-hullset object

**See Also**

lhs.so.add, lhs.merge

---

lhs.to.stats                *Computes statistics on the centroid distances for time-overlapped hulls*

---

**Description**

Computes statistics on the hull-to-hull centroid distance for time-overlapped hulls, with an option to plot and overlay the distribution of centroid distances for random pairs of hulls

**Usage**

```
lhs.to.stats(lhs, id1 = "all", id2 = "all", n = "all", iso.lower = NULL,
  iso.upper = NULL, iso.oz = TRUE, iso.sort.metric = "auto",
  to.comp.hist = TRUE, breaks = 20, to.mcd.outline.only = FALSE,
  lwd.outline = 3, hist.type = c("density", "counts")[1],
  col.to.mcd = "blue", col.h2h.cd = "red", title = NULL,
  title.show = TRUE, title.id.only = FALSE, title.sub.iso.enc = TRUE,
  mar = c(3, 3, if (title.show) 1.5 + (if (title.sub.iso.enc) 1.3 else 0) else
  0.5, 0.5), mgp = c(1.8, 0.5, 0), figs.per.page = 1, panel.num = NULL,
  panel.num.inside.plot = !title.show, png.dir = NULL,
  png.dir.make = TRUE, png.width = 800, png.height = png.width,
  png.overwrite = TRUE, png.pointsize = 12 + (png.width - 480)/80,
  status = TRUE, ...)
```

**Arguments**

| | |
|---|---|
| lhs | A LoCoH-hullset object |
| id1 | Hullset 1 id value(s). Can also be 'all'. |
| id2 | Hullset 2 id value(s). Can also be 'all'. |
| n | The number of randomly selected paired hulls to use as the NULL model of no association. Can also be "all" |
| iso.lower | The lower level isopleth for the isopleth filter |

| | |
|---|---|
| iso.upper | The upper level isopleth for the isopleth filter |
| iso.oz | Apply isopleth filter to hs1 using the isopleths from hs2. T/F. |
| iso.sort.metric | |
| | The name of a hull metric that was used to sort hulls in the construction of the isopleths to be used as filters. If auto (default) it will pick the default sort metric used for density isopleths (i.e., area for the k-method, and number of enclosed points for the a and r methods) |
| to.comp.hist | Draw a histogram of the centroid distances of temporally overlapping hulls |
| breaks | The number of breaks in the histgram (or another valid value for breaks, see [hist](#)). |
| to.mcd.outline.only | |
| | Show the outline only of the histogram of the mean centroid distance for temporally overlapping hulls. T/F. |
| lwd.outline | The line width of the histogram outline (ignored if to.mcd.outline.only=F). |
| hist.type | The type of histogram to plot: 'density' or 'counts'. |
| col.to.mcd | The color of the outline of the histogram of the distribution of the centroid distances of temporally overlapping hulls. |
| col.h2h.cd | The color of the outline of the histogram of the distribution of the centroid distances for randomly paired hulls. |
| title | The title to be displayed. Character. If NULL a title will be constructed. |
| title.show | Whether to show the title. T/F. |
| title.id.only | Whether to construct the title from the id values only. T/F. Ignored if title is passed or title.show=FALSE |
| title.sub.iso.enc | |
| | Whether to include the isopleth filter information as the second line of the title, T/F. |
| mar | The plot margins. A four item numeric vector |
| mgp | The distance away from the edge of the plot for the 1) label, 2) tick marks, and 3) axis line. A three-item numeric vector |
| figs.per.page | The number of plots per page. |
| panel.num | A number or letter to display in the upper left hand corner of the plot when the plot will be used as part of a multi-frame graphic (as in publications). Character |
| panel.num.inside.plot | |
| | Whether to display panel.num inside the plot area itself, as opposed to the title area. T/F |
| png.dir | The directory for a PNG file (filename will be constructed automatically). Ignored if png.fn is passed |
| png.dir.make | Whether to create png.dir if it doesn't exist. T/F |
| png.width | The width of the PNG image |
| png.height | The height of the PNG image |
| png.overwrite | Whether to overwrite an existing PNG file if it exists. T/F |
| png.pointsize | The pointsize (in pixels) for the PNG image, equivalent to the height or width of a character in pixels (increase to make labels appear larger) |
| status | Display status messages. T/F |
| ... | Additional parameters that will be passed to the [plot](#) function |

**Details**

This returns a list object containing the centroid-to-centroid distances of a random selection of hulls from two individuals, which serves as a NULL model of no interaction. It can also plot the histogram of mean-centroid-distance of time-overlapped hulls with the outline of the centroid distance of random-pairs of hulls overlain on top, to visually see how close the distributions match.

It will also compute the Welch Two Sample t-test to see if the distribution for time-overlapped and randomly paired hulls have statistically significant means, and the Two-sample Kolmogorov-Smirnov test which tells you how likely the two distributions are the same.

Note that before you can use this function, the mean-centroid-distance for time-overlapped hulls must be computed using `lhs.to.add`

You can apply an isopleth filter by passing values for iso.lower and iso.upper. These should be the isopleth level (normally between 0 and 1) that the hull parent point must fall in to be included in the analysis. If, for example, you wanted to see whether association in the core area is significantly different from nuetral interaction, you would pass `iso.upper=0.5` and leave `iso.lower` NULL. Note that the isopleths with matching isopleth levels must already be present.

**Value**

A list object

**See Also**

`lhs.to.add`, `lhs.iso.add`

---

| `locoh.lhs` | *Class for a hullsets* |

---

**Description**

Class for a collection of hullsets and associated objects including hull metrics and isopleths

**Details**

An object of class `locoh.lhs` is a list (S3) containing one or more sets of hulls (local MCPs) for a set of points. One hull is saved for each point if the method used to create the hulls returned at least two nearest neighbors for each point, otherwise the number of hulls might be less than the number of points. In addition to hulls (saved as a SpatialPolygonsDataFrame), LoCoH-hullset object may also contain:

- the original points and their attribute table
- an index of the points a) used to construct the hull and b) enclosed by each hull
- hull metrics
- isopleths (progressive unions of hulls) as both vector (SpatialPolygonsDataFrame) and raster (utilzation distribution) formats.
- saved parameters for scatterplots of hull metrics.

In code examples, objects of class `locoh.lhs` are often noted by a 'lhs' suffix, and functions that are designed to work on LoCoH-hullset objects typically start with 'lhs'.

For a complete description of the data structure, see the vignette *T-LoCoH Data Classes*.

**See Also**

lxy.lhs, summary.locoh.lhs, plot.locoh.lhs, lhs.merge, lhs.iso.add, lhs.save, lhs.exp.shp

---

locoh.lxy *Class for location data*

---

**Description**

Data class for locations with associated dates, nearest neighbor info, parameters for a random walk null model, and assorted meta-data

**Details**

An object of class locoh.lxy is a list containing a series of point locations and ancillary variables that go with those locations (e.g., time stamps, point IDs, animal IDs). These items are bundled together and 'ready to go' as inputs into T-LoCoH functions. The benefits of bundling all of the objects related to a set of point locations in a single list object includes:

- cleaning and error checking only has to be done once when the lxy object is created (e.g., with xyt.lxy)

- having all of the ancillary variables together simplifies the task of passing parameters to other functions as well as saving/retrieving your work to disk

- the nearest neighbor lookup-table (which can take a long time to compute) can be reused

- locations for multiple individuals can be saved in one object

In code examples, objects of class locoh.lxy are often indicated by a 'lxy' suffix, and functions that are designed to work on LoCoH-xy objects typically start with 'lxy'.

For a complete description of the data structure, see the vignette *T-LoCoH Data Classes*.

**See Also**

xyt.lxy, lxy.proj.add, lxy.repair, lxy.subset, lxy.merge, lxy.exp.csv, lxy.exp.kml, lxy.exp.shp, move.lxy

---

lxy.explore *Explore a LoCoH-xy object in a interactive map*

---

**Description**

Explore a LoCoH-xy object in a zoomable clickable interactive map

**Usage**

```
lxy.explore(lxy, id = NULL, bg = c("esri_world_imagery", "none")[1],
  connect.dots = FALSE, line.popup = FALSE, status = TRUE)
```

## Arguments

| | |
|---|---|
| `lxy` | A [LoCoH-xy](#) object |
| `id` | The name(s) of individuals to export |
| `bg` | The background layer (i.e., tile) to display, or `'none'` |
| `connect.dots` | Whether to draw a line between segments, T/F |
| `line.popup` | Enable popup balloons on the line segments, T/F |
| `status` | Display status messages, T/F |

## Details

This function displays a Locoh-xy object in an interactive window with a satellite image in the background. You may zoom in and out, pan, and click on locations.

To use this function, you must be using RStudio with the leaflet package installed. To display a satellite image in the background, you must also be connected to the internet.

## See Also

[lxy.exp.kml](#)

## Examples

```
## Not run:
if (!require('devtools')) install.packages('devtools')
devtools::install_github('rstudio/leaflet')
mycon <- url("http://tlocoh.r-forge.r-project.org/toni.n5775.2005-08-22.2006-04-23.lxy.RData")
load(mycon)
close(mycon)
lxy.explore(toni.lxy, connect.dots=TRUE, line.popup=TRUE)

## End(Not run)
```

---

| | |
|---|---|
| `lxy.tumap` | *Computes the time use statistics for a LoCoH-xy object using a gridded surface* |

---

## Description

Computes the time use statistics for a LoCoH-xy object using a gridded surface

## Usage

```
lxy.tumap(lxy, id = NULL, ivg = NULL, gridtype = c("hex", "square")[1],
  cellsize = NULL, mindim = 20, status = TRUE)
```

## Arguments

| | |
|---|---|
| `lxy` | A LoCoH-xy object |
| `id` | The name(s) of individuals to analyze |

| plot.locoh.tumap | *Plot time use maps* |
|---|---|

### Description

Plot time use maps

### Usage

```
plot.locoh.tumap(tumap, breaks = 10, nsv = TRUE, mnlv = TRUE,
  ivg = NULL)
```

| print.locoh.selection | *Prints a locoh.selection object* |
|---|---|

### Description

Prints a locoh.selection object

### Usage

```
## S3 method for class 'locoh.selection'
print(x, ...)
```

### Arguments

| x | An object of class locoh.selection |
|---|---|
| ... | Other arguments |

# Index