

Package ‘orange’

September 15, 2026

Type Package

Title Spherical Descriptors of Geographic Distributions

Version 0.1.0

Maintainer Adam T. Kocsis <adam.t.kocsis@gmail.com>

Description Characterization of distribution data on the surface of a sphere. The primary group of these metrics describe the extent of a distribution, geographic ranges. The calculation of geographic descriptors can be executed using point coordinate data, vector polygons, as well as cells on a discretized sphere. Besides using spherical implementations, the package offers the exploration of partial results for visual diagnostics.

License GPL-3

Date 2026-08-23

BugReports <https://github.com/adamkocsis/orange/issues>

Encoding UTF-8

LazyData false

URL <https://adamtkocsis.com/orange/>

Depends R (>= 4.0.0), icoso (>= 0.12.0)

Imports igraph, methods

NeedsCompilation no

Suggests sf, knitr, rmarkdown, vegan

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.2

Author Adam T. Kocsis [cre, aut] (ORCID:
<<https://orcid.org/0000-0002-9028-665X>>),
Elizabeth M. Dowding [ctb] (ORCID:
<<https://orcid.org/0000-0002-2423-8254>>),
FAU GeoZentrum Nordbayern [fnd]

Repository CRAN

Date/Publication 2026-09-15 10:50:14 UTC

Contents

bradypus	2
centroid	3
emptymap	4
gappiness	5
kentsamples	7
latrange	8
maxdist	9
mstlength	11
occupancy	13
orange	15
pinna	16
plot.shull	16
radius	17
sc_center	19
sc_in	19
sc_shape	20
shull	22
Index	25

bradypus	<i>Occurrences of Bradypus variegatus from GBIF</i>
----------	---

Description

This is an example occurrence record data.frame of a single terrestrial species. GBIF.org (30 June 2026) GBIF Occurrence Download <https://doi.org/10.15468/dl.s7zhvp>

Usage

data(bradypus)

Format

- A data.frame with 1466 observations and 6 variables:
- scientificName Taxon name.
 - basisOfRecord The basis of the record.
 - elevation Elevation of recors
 - decimalLongitude Longitude in decimal degrees.
 - decimalLatitude Latitude in decimal degrees.

centroid	<i>Calculate centroid of a point cloud</i>
----------	--

Description

Calculate centroid of a point cloud

Usage

```
centroid(x, ...)

## S4 method for signature 'matrix'
centroid(
  x,
  long = NULL,
  lat = NULL,
  duplicates = FALSE,
  plot = FALSE,
  plot.args = NULL
)

## S4 method for signature 'data.frame'
centroid(
  x,
  tax = NULL,
  long = "long",
  lat = "lat",
  duplicates = FALSE,
  plot = FALSE,
  plot.args = NULL
)
```

Arguments

<code>x</code>	Eiher a 2-column numeric matrix with two columns: longitudes and latitudes, or a <code>data.frame</code> with these columns.
<code>...</code>	Additional arguments passed to class-specific methods.
<code>long</code>	character, column name of the longitudes.
<code>lat</code>	character, column name of the latitudes.
<code>duplicates</code>	logical, should identical coordinates be included in the calculation (default is FALSE).
<code>plot</code>	Logical, should the result be plotted? Will plot over active plot (as in <code>add=TRUE</code>), if here is any.
<code>plot.args</code>	List arguments passed to the plotting function: <code>points</code> .
<code>tax</code>	character, used only in the <code>data.frame</code> method. Column name of groups (e.g. <code>taxa</code>) that allows the iteration of the method for multiple groups.

Value

Either a single numeric or a list with an estimate and other information.

Examples

```
# Records
data(pinna)
nobilis <- pinna[pinna$species=="Pinna nobilis", ]
plot(nobilis[c("decimalLongitude", "decimalLatitude")], pch=16, col="#00BBAA66")

# Number of unique coordinate pairs
cent <- centroid(nobilis, long="decimalLongitude", lat="decimalLatitude")

points(cent[1], cent[2], col="darkred", pch=3, lwd=4, cex=4)
```

emptymap

Empty world map (Plate Carée)

Description

Empty world map (Plate Carée)

Usage

```
emptymap(
  xlim = c(-180, 180),
  ylim = c(-90, 90),
  grat = c(30, 30),
  col = "#DDDDFE",
  grat.col = "#FFFFFF",
  lwd = 2,
  thick = lwd * 2,
  border = grat.col
)
```

Arguments

xlim	Standard xlim argument of par in degrees longitude.
ylim	Standard ylim argument of par in degrees latitude.
grat	Graticule resolution. Use grat=NULL, if you don't want any.
col	Plot background, RGBA possible.
grat.col	Graticule color RGBA possible.
lwd	The baseline thickness of the lines.
thick	The 0-coordinate circles are highlighted with thicker lines, given here.
border	Border color of rectangle on the outside of the plot.

Value

The function has no return value.

Examples

```
data(kentsamples)
emptymap()
points(kentsamples$central_s, pch=16, col=1)
points(kentsamples$arctic_m, pch=16, col=2)
points(kentsamples$antarctic_l, pch=16, col=3)
points(kentsamples$dateline_l, pch=16, col=4)
```

gappiness

The gappiness a of shape

Description

Proportion of gap cells in an icosahedral grid.

Usage

```
gappiness(x, s, ...)

## S4 method for signature 'character,trigrid'
gappiness(x, s, exclude = NULL, full = FALSE)

## S4 method for signature 'matrix,trigrid'
gappiness(
  x,
  s,
  long = NULL,
  lat = NULL,
  duplicates = FALSE,
  plot = FALSE,
  plot.args = NULL,
  full = FALSE,
  exclude = NULL
)
```

Arguments

x	The list of faces that are part of the shape.
s	Spatial discretization strucutre (currently only trigrid-class icosahedral grid).
...	Arguments passed to class-specific methods.
exclude	The list of faces that is to be excluded from the calculation

<code>full</code>	logical, should only the estimate (FALSE) be returned, or additional data as well?(TRUE).
<code>long</code>	character, column name of the longitudes.
<code>lat</code>	character, column name of the latitudes.
<code>duplicates</code>	logical, should identical coordinates be included in the calculation (default is FALSE)
<code>plot</code>	Logical, should the result be plotted? Will plot over active plot (as in <code>add=TRUE</code>), if here is any.
<code>plot.args</code>	List arguments passed to the plotting function: <code>plot</code> .

Details

Gappiness refers to proportion of area that are covered by the internal gaps that defined by a set of discretized cells or a point set that covers some cells in a discretization structure.

Value

A proportion that gives the ratio of hole-cells compared to all occupied cells.

Examples

```
# 1. Only cells
# create a grid
hex <- hexagrid(2, sf=TRUE)

# an example shape
shape <- paste0("F", c(4, 5, 11, 13, 15, 21, 24, 26, 32, 33, 34, 35, 36))
plot(hex, border="gray80")
plot(hex, shape, col="red", add=TRUE)

# the gappiness
gap <- gappiness(shape, hex, full=TRUE)
gap
# plot the first gap
plot(hex, names(gap$holes[gap$holes==1]), col="#00555577", add=TRUE)

# points
set.seed(7)
rand <- icosal::rpsphere(20, output="polar")
occ <- locate(hex, rand)
plot(hex, border="gray80")
plot(hex, occ, col="red", add=TRUE)
points(rand, col="green", pch=3, cex=2)

# gappiness
gap <- gappiness(occ, hex, full=TRUE)
plot(hex, names(gap$holes), col="orange", add=TRUE)
```

kentsamples

*Samples generated from various Kent distributions***Description**

Geographic test and demonstration sample set

Usage

```
data(kentsamples)
```

Format

A list with 16 matrices (1000 x 2) with columns being equal to longitudes and latitudes.

scientificname Taxon name.

central_s Central position, small spread.

central_m Central position, medium spread.

central_l Central position, large spread.

central_xl Central position, extra large spread.

arctic_s Arctic position, small spread.

arctic_m Arctic position, medium spread.

arctic_l Arctic position, large spread.

arctic_xl Arctic position, extra large spread.

antarctic_s Antarctic position, small spread.

antarctic_m Antarctic position, medium spread.

antarctic_l Antarctic position, large spread.

antarctic_xl Antarctic position, extra large spread.

dateline_s Dateline position, small spread.

dateline_m Dateline position, medium spread.

dateline_l Dateline position, large spread.

dateline_xl Dateline position, extra large spread.

Details

A set of coordinates generated from Kent distributions with the sample number $n = 1000$. The samples are elements of a list, 16 in total with the combination of location and spread (i.e. the kappa parameter or the Kent distribution). There are four locations: central ($0^{\circ}\text{N } 0^{\circ}\text{E}$), arctic ($85^{\circ}\text{N } 5^{\circ}\text{E}$), antarctic ($85^{\circ}\text{S } 5^{\circ}\text{E}$) and dateline ($170^{\circ}\text{E } 5^{\circ}\text{N}$), and there are four sizes: small (s, $\text{kappa} = 50$), medium (m, $\text{kappa} = 15$), large (l, $\text{kappa} = 5$), and extra-large (xl, $\text{kappa} = 2$).

latrange	<i>Calculate latitudinal ranges</i>
----------	-------------------------------------

Description

Calculate latitudinal ranges

Usage

```
latrange(x, ...)

## S4 method for signature 'matrix'
latrange(
  x,
  long = NULL,
  lat = NULL,
  q = 1,
  duplicates = FALSE,
  plot = FALSE,
  plot.args = NULL,
  full = FALSE
)

## S4 method for signature 'data.frame'
latrange(
  x,
  tax = NULL,
  q = 1,
  long = "long",
  lat = "lat",
  duplicates = FALSE,
  plot = FALSE,
  plot.args = NULL,
  full = FALSE
)
```

Arguments

x	Either a 2D numeric matrix with two columns: longitudes and latitudes, a data.frame with the same information.
...	Arguments passed to class-specific methods.
long	character, column name of the longitudes.
lat	character, column name of the latitudes.
q	numeric, a value between 0 and 1, the quantile.
duplicates	logical, should identical coordinates be included in the calculation (default is FALSE)

plot	Logical, should the result be plotted? Will plot over active plot (as in add=TRUE), if here is any.
plot.args	List arguments passed to the plotting function: points.
full	logical, should only the estimate (FALSE) be returned, or additional data as well?(TRUE).
tax	character, used only in the data.frame method. Column name of groups (e.g. taxa) that allows the iteration of the method for multiple groups.

Value

Either a single numeric or a list with an estimate and other information. If iterated using ‘tax’, then either a named vector or list of lists.

Examples

```
# 1. Records
data(pinna)
# Subset to Pinna nobilis
nobilis <- pinna[pinna$species=="Pinna nobilis", ]
plot(nobilis[c("decimalLongitude", "decimalLatitude")], pch=16, col="#00BBAA66")

# Number of unique coordinate pairs
lr <- latrange(nobilis, long="decimalLongitude", lat="decimalLatitude", full=TRUE)

abline(h=lr$range, col="darkred", lty=3, lwd=4)
```

maxdist

Calculate ranges with the maximum distance method

Description

This family of metrics rely on the maximum distance within a point cloud

Usage

```
maxdist(x, s, ...)

mgcd(x, ...)

## S4 method for signature 'matrix,missing'
maxdist(
  x,
  dm = NULL,
  long = NULL,
  lat = NULL,
  duplicates = FALSE,
  plot = FALSE,
```

```

    plot.args = NULL,
    full = FALSE,
    q = 1
  )

## S4 method for signature 'data.frame,missing'
maxdist(
  x,
  long = "long",
  lat = "lat",
  tax = NULL,
  dm = NULL,
  duplicates = FALSE,
  q = 1,
  plot = FALSE,
  plot.args = NULL,
  full = FALSE
)

```

Arguments

x	Either a 2D numeric matrix with two columns: longitudes and latitudes, a <code>data.frame</code> with the same information.
s	Substitute spatial structure. An icosahedral grid object the inherits from the <code>trigrd</code> class. Providing this argument reduces the point cloud to the centers of the grid cells. (not yet)
...	Additional arguments passed to class-specific methods.
dm	If there is a pre-made distance matrix, it can be plugged in here. If this is provided, the default coordinates will not be used.
long	character, column name of the longitudes.
lat	character, column name of the latitudes.
duplicates	logical, should identical coordinates be included in the calculation (default is FALSE)
plot	Logical, should the result be plotted? Will plot over active plot (as in <code>add=TRUE</code>).
plot.args	List arguments passed to the plotting function: <code>sf::plot</code> .
full	logical, should only the estimate (FALSE) be returned, or additional data as well?(TRUE).
q	numeric, a value between 0 and 1, the quantile.
tax	character, used only in the <code>data.frame</code> method. Column name of groups (e.g. taxa) that allows the iteration of the method for multiple groups.

Details

This metrics includes maximum great circle distance and similar methods.

Value

A list with an estimate and two indices the rows of the input matrix that represent the longest great circle (or one of them).

Examples

```
# 1. Records
data(pinna)
# Subset to Pinna nobilis
nobilis <- pinna[pinna$species=="Pinna nobilis", ]
plot(nobilis[c("decimalLongitude", "decimalLatitude")], pch=16, col="#00BBAA66")

# 2. calculate and visualize
mgcd <- maxdist(nobilis, long="decimalLongitude", lat="decimalLatitude", plot=TRUE, full=TRUE)
```

mstlength

Calculate ranges with the minimum spanning tree length method

Description

This family of metrics rely on constructing a minimum spanning tree from the distances between the points, and use its length to describe spreading.

Usage

```
mstlength(x, s, ...)

## S4 method for signature 'matrix,missing'
mstlength(
  x,
  dm = NULL,
  long = NULL,
  lat = NULL,
  duplicates = FALSE,
  plot = FALSE,
  plot.args = NULL,
  full = FALSE,
  q = 1
)

## S4 method for signature 'data.frame,missing'
mstlength(
  x,
  long = "long",
  lat = "lat",
  tax = NULL,
```

```

    dm = NULL,
    duplicates = FALSE,
    q = 1,
    plot = FALSE,
    plot.args = NULL,
    full = FALSE
  )

```

Arguments

<code>x</code>	Either a 2D numeric matrix with two columns: longitudes and latitudes, a <code>data.frame</code> with the same information.
<code>s</code>	Structure to replace the points, either missing (coordinate pairs) or a trigrigrid (icosahedral grid from the package <code>icosa</code>).
<code>...</code>	Additional arguments passed to class-specific methods.
<code>dm</code>	If there is a pre-made distance matrix, it can be plugged in here. If this is provided, the default coordinates will not be used.
<code>long</code>	character, column name of the longitudes.
<code>lat</code>	character, column name of the latitudes.
<code>duplicates</code>	logical, should identical coordinates be included in the calculation (default is FALSE)
<code>plot</code>	Logical, should the result be plotted? Will plot over active plot (as in <code>add=TRUE</code>).
<code>plot.args</code>	List arguments passed to the plotting function: <code>sf::plot</code> .
<code>full</code>	logical, should only the estimate (FALSE) be returned, or additional data as well?(TRUE).
<code>q</code>	numeric, a value between 0 and 1, the quantile.
<code>tax</code>	character, used only in the <code>data.frame</code> method. Column name of groups (e.g. taxa) that allows the iteration of the method for multiple groups.

Details

This metrics includes maximum great circle distance and similar methods.

Value

A list with an estimate and two indices the rows of the input matrix that represent the length of the tree (or one of them).

Examples

```

# 1. Records
data(pinna)
# Subset to Pinna nobilis
nobilis <- pinna[pinna$species=="Pinna nobilis", ]
plot(nobilis[c("decimalLongitude", "decimalLatitude")], pch=16, col="#00BBAA66")

# 2. calculate and visualize

```

```
mst <- mstlength(nobilis, long="decimalLongitude", lat="decimalLatitude", plot=TRUE, full=TRUE)
mst$estimate
```

occupancy

Calculate ranges with the occupancy method

Description

High-level, abstract function to calculate how many or what proportion of components in a pre-specified spatial structure is occupied by a distribution dataset

Usage

```
occupancy(x, s, ...)

## S4 method for signature 'matrix,missing'
occupancy(x, long = NULL, lat = NULL, full = FALSE, prop = NULL)

## S4 method for signature 'data.frame,missing'
occupancy(
  x,
  tax = NULL,
  long = "long",
  lat = "lat",
  full = FALSE,
  listarray = TRUE,
  prop = NULL
)

## S4 method for signature 'data.frame,character'
occupancy(x, s, tax = NULL, full = FALSE, prop = NULL, listarray = TRUE)

## S4 method for signature 'matrix,trigrid'
occupancy(
  x,
  s,
  long = NULL,
  lat = NULL,
  q = 1,
  plot = FALSE,
  plot.args = NULL,
  full = FALSE,
  prop = NULL
)

## S4 method for signature 'data.frame,trigrid'
occupancy(
```

```

x,
s,
long = "long",
lat = "lat",
tax = NULL,
q = 1,
plot = FALSE,
plot.args = NULL,
full = FALSE,
prop = NULL
)

```

Arguments

<code>x</code>	Eiher a 2-column numeric matrix with two columns: longitudes and latitudes, or a <code>data.frame</code> with these columns.
<code>s</code>	Structure to be occupied, either <code>NULL</code> (coordinate pairs), character (column name indicating locality) or a <code>trigrd</code> (icosahedral grid from the package <code>icosa</code>).
<code>...</code>	Additional arguments passed to class-specific methods.
<code>long</code>	character, column name of the longitudes.
<code>lat</code>	character, column name of the latitudes.
<code>full</code>	Logical switch indicating whether only the estimate should be shown (<code>FALSE</code>), or other info (i.e. the list of occupied components in <code>s</code>) as well.
<code>prop</code>	Should counts be returned (<code>prop=NULL</code>), or proportions? If <code>prop="global"</code> , then global proportions are returned, if <code>prop="relative"</code> , relative proportions are calculated.
<code>tax</code>	character, used only in the <code>data.frame</code> method. Column name of groups (e.g. taxa) that allows the iteration of the method for multiple groups.
<code>listarray</code>	If the full traceable output is required, should this be organized with list-array (native output of <code>tapply</code>).
<code>q</code>	Minimum occupancy with <code>q</code> proportion of occurrences (not yet implemented!).
<code>plot</code>	Logical, should the result be plotted? Will plot over active plot (as in <code>add=TRUE</code>).
<code>plot.args</code>	List arguments passed to the plotting function: <code>lines</code> .

Details

The function by default returns counts (i.e. natural numbers) of how many discrete units are occupied. However, there are many cases, when proportions are much more useful, which can be set with the `prop` argument. Proportions are either global (`prop="global"`) or relative (`prop="relative"`). Global proportions express how much of the overall spatial structure (`s`) is occupied; for example, what is the proportion of cells in a grid that are occupied. This method is only applicable, when `s` is defined as a spatial structure that is independent from `x`. In contrast, relative proportional occupancies express proportions in the sampled set, i.e. what proportion of the overall sampled grid cells or localities are occupied by a taxon. This is particularly useful when `tax!=NULL`.

Value

For single subsets (tax=NULL) either a single numeric or an orange list with an estimate and other information. Iterations for multiple taxa result in a named numeric vector or a list.

Examples

```
# I. Single taxon: Pinna nobilis
# 1. Records
data(pinna)
# Subset to Pinna nobilis
nobilis <- pinna[pinna$species=="Pinna nobilis", ]

# Number of unique coordinate pairs
cpairs <- occupancy(nobilis, long="decimalLongitude", lat="decimalLatitude")

# 2. Occupancy in icosahedral grid
hex <- hexagrid(deg=5, sf=TRUE)
plot(nobilis[c("decimalLongitude", "decimalLatitude")], pch=16, col="#00BBAA66",
      xlim=c(-20,45), ylim=c(20,60))
plot(hex, border="gray70", add=TRUE)

# calculate occupancy
occ <- occupancy(nobilis, s=hex, plot=TRUE,
                 long="decimalLongitude", lat="decimalLatitude", full=TRUE)

# manual coloring from full output
plot(hex, occ$occupied, add=TRUE, col="#00BB0088")
```

orange

Orange

Description

Spherical Ranges and Parametrization of Geographic Shapes

Details

A collation of tools and metrics to characterize distribution data on the surface of a sphere or an ellipsoid. The primary group of these metrics is those that describe the extent of a distribution (geographic ranges). The calculation of geographic ranges can be executed using point coordinates data, vector polygons, as well as cells on a discretized sphere. Besides ensuring the use a geometrically correct implementations, the package offers the exploration of partial results for visual diagnostics. This is still the pre-alpha version. Notes about found bugs and suggestions are more than welcome!

Author(s)

Adam T. Kocsis (adam.t.kocsis@gmail.com)

See Also

Useful links:

- <https://adamtkocsis.com/orange/>
- Report bugs at <https://github.com/adamkocsis/orange/issues>

pinna

Occurrences of the genus Pinna from OBIS

Description

This is an example occurrence record data.frame with multiple taxa. Occurrence records downloaded from OBIS on 2026-07-01.

Usage

```
data(pinna)
```

Format

A data.frame with 1466 observations and 6 variables:

scientificName Taxon name.

species Species name.

basisOfRecord The basis of the record.

year Year.

depth Depth in meters.

decimalLongitude Longitude in decimal degrees.

decimalLatitude Latitude in decimal degrees.

plot.shull

Plotting of spherical hulls

Description

Plotting of spherical hulls

Usage

```
## S3 method for class 'shull'
plot(x, add = FALSE, ...)
```

Arguments

x	Spherical hull object.
add	Logical parameter specifying whether the previous plot should be overwritten.
...	Additional plotting parameters as defined in par.

Value

The function has no return value.

radius	<i>Calculate ranges with the radius-group of methods</i>
--------	--

Description

This family of metrics rely on estimating the distance of an extent from a fixed point.

Usage

```
radius(x, s, ...)

## S4 method for signature 'matrix,missing'
radius(
  x,
  p = NULL,
  long = NULL,
  lat = NULL,
  duplicates = FALSE,
  plot = FALSE,
  plot.args = NULL,
  full = FALSE,
  q = 1
)

## S4 method for signature 'data.frame,missing'
radius(
  x,
  p = NULL,
  long = "long",
  lat = "lat",
  tax = NULL,
  duplicates = FALSE,
  q = 1,
  plot = FALSE,
  plot.args = NULL,
  full = FALSE
)
```

Arguments

<code>x</code>	Either a 2D numeric matrix with two columns: longitudes and latitudes, a <code>data.frame</code> with the same information - or a character vector of cell identifiers.
<code>s</code>	Structure to substitute the points, either missing (using coordinate pairs) or a <code>trigrd</code> (icosahedral grid from the package <code>icosa</code>).
<code>...</code>	Additional arguments passed to class-specific methods.
<code>p</code>	A single point of reference (longitude/latitude). If missing, the point of reference will be the centroid given by <code>centroid</code> .
<code>long</code>	character, column name of the longitudes.
<code>lat</code>	character, column name of the latitudes.
<code>duplicates</code>	logical, should identical coordinates be included in the calculation (default is <code>FALSE</code>)
<code>plot</code>	Logical, should the result be plotted? Will plot over active plot (as in <code>add=TRUE</code>).
<code>plot.args</code>	List arguments passed to the plotting function: <code>sf::plot</code> .
<code>full</code>	logical, should only the estimate (<code>FALSE</code>) be returned, or additional data as well? (<code>TRUE</code>).
<code>q</code>	numeric, a value between 0 and 1, the quantile.
<code>tax</code>	character, used only in the <code>data.frame</code> method. Column name of groups (e.g. <code>taxa</code>) that allows the iteration of the method for multiple groups.

Details

This group of methods rely on calculating single distances.

Value

A list with an estimate and two indices the rows of the input matrix that represent the length of the tree (or one of them).

Examples

```
# 1. Records
data(pinna)
# Subset to Pinna nobilis
nobilis <- pinna[pinna$species=="Pinna nobilis", ]
plot(nobilis[c("decimalLongitude", "decimalLatitude")], pch=16, col="#00BBAA66")

# 2. calculate and visualize
rad <- radius(nobilis, long="decimalLongitude", lat="decimalLatitude", plot=TRUE, full=TRUE)
```

sc_center	<i>Identify the center of a small circle based on three points of a sphere</i>
-----------	--

Description

Identify the center of a small circle based on three points of a sphere

Usage

```
sc_center(x, origin = c(0, 0, 0), output = "polar")
```

Arguments

x	A matrix of 3 points, either polar longitude and latitude coordinates, or XYZ Cartesian coordinates.
origin	The origin of the sphere.
output	If set to "polar" then the function will return the longitude and latitude of the small circle center.

Value

A list with two elements, the center of the small circle and the surface radius of the small circle.

Examples

```
# generate 3 points on a sphere
set.seed(2)
ps <- icosar::rpsphere(3, output="polar")
small <- sc_center(x=ps)
circle<- sc_shape(x=small$center, r=small$r, output="polar")

plot(NULL, NULL, xlim=c(-180, 180), ylim=c(-90, 90))
points(ps, col="red", pch=3, cex=4)
points(small$center, col="blue", pch=16)
points(circle, col="green", pch=16)
```

sc_in	<i>Identify whether a given set of points is in, on or out of a small circle</i>
-------	--

Description

This method calculates the plane of the small circle and uses the normal to evaluate whether points are in or outside the small circle.

Usage

```
sc_in(x, center, r, origin = c(0, 0, 0))
```

Arguments

x	A set of points to be determined a matrix of longitudes and latitudes.
center	The center of a small circle
r	The surface radius of a small circle.
origin	The origin of the sphere.

Value

A numeric vector indicating whether the points are in (1), on (0) or outside of the small circle.

Examples

```
# set.seed(1)
# generate a random small circle
set.seed(1)
pol <- icos::rpsphere(1000, output="polar")
# use the first as a small circle's center
center <- pol[1, , drop=FALSE]
radius <- 5000 # define radius in km

# draw a small circle
circle <- sc_shape(x=center, r=radius, breaks=200)

# visualize
plot(NULL, NULL, xlim=c(-180, 180), ylim=c(-90, 90))
points(pol, col="red", pch=3, cex=2)
arcs(circle, col="blue", lwd=2)

# find points in small circle (about 10x faster than distance evaluation)
inside <- sc_in(x=pol, center=center, r=radius)
points(pol[inside==1, ], col="green", pch=3, cex=2)
```

sc_shape

Generate coordinates of small circles

Description

Creating points along small circles

Usage

```
sc_shape(
  x = "random",
  r = NULL,
  r.ex = NULL,
  r.rad = NULL,
  r.deg = NULL,
```

```

    n = NULL,
    breaks = 100,
    radius = 6371.007,
    origin = c(0, 0, 0),
    output = "polar",
    sf.type = "polygon",
    sf.wrap.dateline = TRUE,
    drop = TRUE
  )

```

Arguments

x	Numeric matrix or character string. Coordinates of the centers of the small circles
r	The surface radius of the small circle expressed as great circle distance of the small circle's points from the center of the small circles.
r.ex	The extrinsic radius of the small circles, i.e. the radius of the circle in the plane of the small circle.
r.rad	The angle of the small circle in radians.
r.deg	The angle of the small circle in degrees.
n	Integer or NULL. only used if "x= random"
breaks	Integer. The number of points to create from the small circle.
radius	The radius of the sphere, defaults to the authalic radius of Earth - relevant only for Cartesian output.
origin	The center of the sphere (don't touch this, unless you really think you know what you are doing!).
output	Output structure for the function. The value output="polar" will return the polar (longitude-latitude) coordinates of the small circles in an array. The setting output="cartesian" will return the 3D cartesian coordinates of the small circles in an array. The option output="sf" will return an sfc geometry collection.
sf.type	The type of sf object to be returned.
sf.wrap.dateline	Argument indicating whether returned sf object should be wrapped around the dateline.
drop	If there is a single small circle to be generated, should its array wrapper be dropped?

Details

The function generates a single reference (standard) small circle around the (0,0) long-lat coordinates using a given radius. This reference circle is then rotated to match a given center. The function is iterated for multiple radii and centers (separately!).

Value

Either a numeric array or and sfc geometry collection.

Examples

```
set.seed(1)
# generate a random small circle
central <- sc_shape(r=6000, breaks=40)
plot(NULL, xlim=c(-180, 180), ylim=c(-90,90), xlab="longitude", ylab="latitude")
points(central, pch=17, col="gray")
# repeat, same radius multiple centers
smaller <- sc_shape(x=central, r=1000, breaks=100)
for(i in 1:nrow(central)) points(smaller[,i], col=i, pch=16)

# Random circles with different radii
diff200 <- sc_shape(r=runif(200, 100, 10000))
plot(NULL, xlim=c(-180, 180), ylim=c(-90,90), xlab="", ylab="", axes=FALSE)
for(i in 1:(dim(diff200)[3])){
  points(diff200[,i], col=i,pch=16)
}

# Same center, different radius
cent <- central[10, ,drop=FALSE]
outward <- sc_shape(cent, r=seq(100, 15000, length.out=20))
plot(NULL, xlim=c(-180, 180), ylim=c(-90,90), xlab="", ylab="", axes=FALSE)
for(i in 1:(dim(outward)[3])){
  points(outward[,i], col=i,pch=16)
}
```

shull

Spherical hull geometries and hull area calculations

Description

Calculation of hull geometries in spherical space

Usage

```
shull(x, s, ...)

## S4 method for signature 'matrix,missing'
shull(
  x,
  method,
  long = NULL,
  lat = NULL,
  duplicates = FALSE,
  plot = FALSE,
  plot.args = NULL,
  sphererad = authRadius
)
```

```

## S4 method for signature 'data.frame,missing'
shull(x, long = "long", lat = "lat", ...)

## S4 method for signature 'matrix,trigrid'
shull(
  x,
  s,
  method,
  long = NULL,
  lat = NULL,
  duplicates = FALSE,
  plot = FALSE,
  plot.args = NULL,
  sphererad = authRadius,
  drop = TRUE
)

shullarea(x, ...)

## Default S3 method:
shullarea(
  x,
  method,
  metric = "area",
  full = FALSE,
  sphererad = authRadius,
  ...
)

## S3 method for class 'shull'
shullarea(x, metric = "area", s = NULL, full = FALSE, ...)

```

Arguments

<code>x</code>	Either a 2-column numeric matrix with two columns: longitudes and latitudes, or a <code>data.frame</code> with these columns.
<code>s</code>	An external spatial structure to resolve the hull (e.g. a hexagrid object.)
<code>...</code>	Additional arguments passed to class-specific methods.
<code>method</code>	character Hull construction method. Currently available: "centroidcircle"
<code>long</code>	character, column name of the longitudes.
<code>lat</code>	character, column name of the latitudes.
<code>duplicates</code>	logical, should identical coordinates be included in the calculation (default is FALSE).
<code>plot</code>	Logical, should the result be plotted? Will plot over active plot (as in <code>add=TRUE</code>), if there is any.
<code>plot.args</code>	List arguments passed to the plotting function: <code>lines</code> .

sphererad	numeric The radius of the sphere used in the calculations, defaults to the australian radius of Earth in km (6370.997).
drop	logical In case s is provided, should the returned object be a hull-class object, or only the identifiers of s (drop=FALSE)?
metric	character What metric should be used for area of the hull? area returns the values in square of the unit of sphererad (defaults to square km), prop returns the area as a proportion of the sphere, count returns the area as the number of components it occupies in s, when applicable.
full	Logical switch indicating whether only the estimate should be shown (FALSE), or the hull itself should be returned as well?.

Details

Constructing and enclosing geometry around a distribution on the surface of a sphere represent unique challenges. The "centroidcircle" method is using the centroid of the distribution as the center of small circle that defines a spherical cap. This is likely not the smallest possible cap to cover the distribution and takes the heterogeneous density of the distribution into account.

Value

A hull-class object, a vector of identifiers (if applicable and drop=FALSE) or the area of the hull as a numeric.

Examples

```
# example data
data(kentsamples)
p <- kentsamples$dateline_m

# plotting
plot(NULL, NULL, xlim=c(-180, 180), ylim=c(-90,90), xlab="", ylab="")
points(p, pch=3, col="red")

# basic spherical hull definition (native)
cc <- shull(p, method="centroidcircle")
plot(cc, add=TRUE)

# spherical hull area calculation
shullarea(cc)

# basic hull definition (icosahedral grid)
hex <- icosah::hexagrid(spacing=8, sf=TRUE)
# dropping hull container
cc_hex <- shull(p, s=hex, method="centroidcircle")
plot(hex, cc_hex, col="#0000DD33", border="#FFFFFF77", add=TRUE)

# with shull container
cc_hex2 <- shull(p, s=hex, method="centroidcircle", drop=FALSE)
shullarea(cc_hex2, s=hex, metric="prop")
```

Index

* datasets

- bradypus, [2](#)
- kentsamples, [7](#)
- pinna, [16](#)
- bradypus, [2](#)
- centroid, [3](#)
- centroid, data.frame-method (centroid), [3](#)
- centroid, matrix-method (centroid), [3](#)
- emptymap, [4](#)
- gappiness, [5](#)
- gappiness, character, trigrid-method (gappiness), [5](#)
- gappiness, matrix, trigrid-method (gappiness), [5](#)
- kentsamples, [7](#)
- latrange, [8](#)
- latrange, data.frame-method (latrange), [8](#)
- latrange, matrix-method (latrange), [8](#)
- maxdist, [9](#)
- maxdist, data.frame, missing-method (maxdist), [9](#)
- maxdist, matrix, missing-method (maxdist), [9](#)
- mgcd (maxdist), [9](#)
- mstlength, [11](#)
- mstlength, data.frame, missing-method (mstlength), [11](#)
- mstlength, matrix, missing-method (mstlength), [11](#)
- occupancy, [13](#)
- occupancy, data.frame, character-method (occupancy), [13](#)
- occupancy, data.frame, missing-method (occupancy), [13](#)
- occupancy, data.frame, trigrid-method (occupancy), [13](#)
- occupancy, matrix, missing-method (occupancy), [13](#)
- occupancy, matrix, trigrid-method (occupancy), [13](#)
- orange, [15](#)
- orange-package (orange), [15](#)
- pinna, [16](#)
- plot.shull, [16](#)
- radius, [17](#)
- radius, data.frame, missing-method (radius), [17](#)
- radius, matrix, missing-method (radius), [17](#)
- sc_center, [19](#)
- sc_in, [19](#)
- sc_shape, [20](#)
- shull, [22](#)
- shull, data.frame, missing-method (shull), [22](#)
- shull, matrix, missing-method (shull), [22](#)
- shull, matrix, trigrid-method (shull), [22](#)
- shullarea (shull), [22](#)