Package 'dssd'

August 5, 2019

Imports sf, plot3D, methods

Suggests knitr, rmarkdown, tibble, testthat

VignetteBuilder knitr

Type Package

Title Distance Sampling Survey Design

Version 0.1.0

Author Laura Marshall <lhm@st-andrews.ac.uk>

Maintainer Laura Marshall <lhm@st-andrews.ac.uk>

Description Creates survey designs for distance sampling surveys. These designs can be assessed for various effort and coverage statistics. Once the user is satisfied with the design characteristics they can generate a set of transects to use in their distance sampling survey. Many of the designs implemented in this R package were first made available in our 'Distance' for Windows software and are detailed in Chapter 7 of Advanced Distance Sampling, Buckland et. al. (2008, ISBN-13: 978-0199225873). Find out more about estimating animal/plant abundance with distance sampling at <http://distancesampling.org/>.

BugReports https://github.com/DistanceDevelopment/dssd/issues

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Collate 'Class.Constructors.R' 'Coverage.Grid.R' 'Transect.R' 'Region.R' 'generic.functions.R' 'Survey.Design.R' 'Line.Transect.Design.R' 'Line.Transect.R' 'Point.Transect.Design.R' 'Point.Transect.R' 'calc.region.width.R' 'calculate.trackline.pl.R' 'calculate.trackline.zz.R' 'calculate.trackline.zzcom.R' 'check.line.design.R' 'check.point.design.R' 'check.shape.R' 'generate.eqspace.zigzags.R' 'generate.parallel.lines.R' 'generate.random.points.R' 'generate.systematic.points.R' 'get.intersection.points.R' 'run.coverage.R' 'write.transects.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-05 13:00:02 UTC

R topics documented:

`Coverage.Grid-class` *Class "Coverage.Grid"*

Description

Class `"Coverage.Grid"` is an S4 class containing descriptions of a grid used to assess the coverage scores of different designs.

Slots

`grid` sf multipoint object

`spacing` the spacing used to create the coverage grid

Objects from the Class

Objects can be created by calls of the form `make.grid(region = make.region(),no.points = 1000,spacing = numeric(0)`

Methods

`plot signature(x = "Coverage.Grid",y = "missing")`: plots the grid of points.

`generate.transects` *S4 generic method to generate an instance of a design*

Description

Uses the Survey.Design details to generate a set of transects. The transects are returned within an object of class Transect which records some of the design options used to generate it along with the samplers as an sf object of class 'POINT' or 'LINESTRING'/'MULTILINESTRING'. The Transect object also contains the covered areas as a 'POLYGON' or 'MULTIPOLYGON' sf object.

Usage

```
generate.transects(object, quiet = FALSE, ...)

## S4 method for signature 'Line.Transect.Design'
generate.transects(object,
  quiet = FALSE, ...)

## S4 method for signature 'Point.Transect.Design'
generate.transects(object,
  quiet = FALSE, ...)
```

Arguments

`object`	an object which inherits from class Survey.Design
`quiet`	silences some warnings
`...`	optional arguments used for internal calls

Value

an object of class Transect

get.area *Returns the area of the region*

Description

Returns the area of the region

Usage

```
get.area(object)

## S4 method for signature 'Region'
get.area(object)
```

Arguments

object	object of class Region

Value

numeric value specifying the area of the region

get.coverage *S4 generic method to extract coverage scores*

Description

Obtains the coverage scores from the survey design object.

Usage

```
get.coverage(object)

## S4 method for signature 'Survey.Design'
get.coverage(object)
```

Arguments

object	an object which inherits from class Survey.Design

Value

a vector of coverage scores

Line.Transect-class *Class "Line.Transect" extends Class "Transect"*

Description

Virtual Class "Line.Transect" is an S4 class detailing a set of transects from a point transect design.

Slots

line.length the total line length for the transect set

trackline the total on and off effort trackline length from the start of the first transect to the end of the last

cyclictrackline the trackline distance plus the distance required to return from the end of the last transect to the beginning of the first

See Also

make.design

Line.Transect.Design-class *Virtual Class "Line.Transect.Design" extends Class "Survey.Design"*

Description

Virtual Class "Line.Transect.Design" is an S4 class detailing the type of line transect design.

Slots

line.length Numeric value defining the total line length to be generated (may be multiple values relating to each stratum).

bounding.shape relevant for zigzag designs, either a minimum bounding "rectangle" or a "convex hull".

Methods

generate.transects signature=(object = "Line.Transect.Design",...): generates a set of transects from a shapefile.

See Also

make.design

make.coverage *Creates a Coverage.Grid object*

Description

This creates an instance of the Coverage.Grid class.

Usage

```
make.coverage(region = make.region(), spacing = numeric(0),
  n.grid.points = 1000)
```

Arguments

region	the region name
spacing	spacing to be used to create the coverage grid
n.grid.points	the desired number of grid points (note that the exact number generated may differ slightly depending on the shape of the study region).

Value

object of class Coverage.Grid

Author(s)

Laura Marshall

Examples

```
# This example will take a bit of time to generate
# A coverage grid in a rectangular region of 2000 x 500
region <- make.region()
cover <- make.coverage(region, spacing = 50)
plot(region, cover)
# Create coverage grid by approx number of grid points
cover <- make.coverage(region, n.grid.points = 100)
plot(region, cover)


# Fast running example for CRAN testing purposes
# This spacing is too sparse to assess coverage in a real example
region <- make.region()
cover <- make.coverage(region, spacing = 250)
plot(region, cover)
```

make.design *Creates a Survey.Design object*

Description

Creates a description of a survey design. Designs may use different types of either point or line transect designs across strata but cannot mix point and line transect design types within a single design object.

Usage

```
make.design(region = make.region(), transect.type = "line",
  design = "systematic", samplers = numeric(0),
  line.length = numeric(0), effort.allocation = numeric(0),
  design.angle = 0, spacing = numeric(0), edge.protocol = "minus",
  bounding.shape = "rectangle", truncation = 1, coverage.grid = NULL)
```

Arguments

region	an object of class Region defining the survey region.
transect.type	character variable specifying either "line" or "point"
design	a character variable describing the type of design. Either "random", "systematic", "eszigzag" (equal-spaced zigzag) or "eszigzagcom" (equal spaced zigzag with complementary lines). See details for more information.
samplers	the number of samplers you wish the design to generate (note that the number actually generated may differ slightly due to the shape of the study region for some designs). This may be one value of a value per strata.
line.length	the total line length you desire or a vector of line lengths the same length as the number of strata.
effort.allocation	numeric values used to indicate the proportion of effort to be allocated to each strata from number of samplers or line length. If length is 0 (the default) and only a total line length or total number of samplers is supplied, effort allocated based on stratum area.
design.angle	numeric value detailing the angle of the design. Can provide multiple values relating to strata. The use of the angle varies with design, it can be either the angle of the grid of points, the angle of lines or the design axis for the zigzag design. See details.
spacing	used by systematic designs, numeric value to define spacing between transects. Can be a vector of values with one value per strata.
edge.protocol	character value indicating whether a "plus" sampling or "minus" sampling protocol is used.
bounding.shape	only applicable to zigzag designs. A character value saying whether the zigzag transects should be generated using a minimum bounding "rectangle" or a "convex hull".

truncation	A single numeric value describing the longest distance at which an object may be observed. Truncation distance is constant across strata.
coverage.grid	An object of class Coverage.Grid for use when running the coverage simulation.

Details

For point transect designs the user may either specify "random" or "systematic" for the design argument. If the user specifies "random", they should also provide a value for effort detailing the number of point transects they wish their survey to have. For stratified designs they may specify a vector of numbers detailing the number of transects per strata or alternatively use the effort.allocation argument to allocate a total effort amount proportionally. If effort.allocation is left blank then effort will be allocated according to strata area. If the user specified"systematic" they may either provide their desired number of samplers or a value for spacing which defines the gap between each of the points (again a vector of spacing values can be provided for each strata). Optionally the user may select a design.angle. For both random and systematic point transect designs the user may select either a minus or plus sampling edge protocol.

For line transect designs the user may either specify "random" (randomly placed full width lines), "systematic" (systematically placed full width lines), "eszigzag" (equally spaced zigzag lines) or "eszigzagcom" (two sets of complementary equally spaced zigzag lines). If the user specifies "random", they should provide the either the number of samplers they wish the design to generate or the line length they wish to achieve, either by strata or as a total. If the user specifies "systematic" they should specify either the number of samplers, the desired line length or the spacing between lines. The design angle for these parallel line designs refers to the angle of the lines where 0 is a vertical line and moving round in a clockwise direction. If the user specifies a zigzag design they should specify the systematic spacing value, number of samplers or line length to be used and should choose between generating the design in a minimum bounding rectangle or a convex hull. The default is minimum bounding rectangle which gives more even coverage but the convex hull is generally more efficient. The designs may be generated using plus or minus sampling protocols. Similar to the point transect designs different values may be specified for each strata for all of the above options. The design angle for the zigzag designs refers to the angle of a line which would run through the middle of each zigzag transect if the zigzags were to be generated within a rectangle. The design angle for zigzags should usually run along the longest dimension of the study region.

See the Multi Strata Vignette for more complex examples of defining designs across multiple strata.

Value

object of a class which inherits from class Survey.Design

Author(s)

Laura Marshall

Examples

```
#Point transect example
shapefile.name <- system.file("extdata", "TrackExample.shp", package = "dssd")
region <- make.region(region.name = "study area",
                      shape = shapefile.name)
```

```
# Generate coverage grid
cover <- make.coverage(region,
                       n.grid.points = 500)

# Define design
design <- make.design(region = region,
                      transect.type = "point",
                      design = "random",
                      samplers = 25,
                      design.angle = 45,
                      edge.protocol = "minus",
                      truncation = 3,
                      coverage.grid = cover)

# Generate a single survey instance
survey <- generate.transects(design)
plot(region, survey, covered.area = TRUE)

# Warning! this will take some time to run
design <- run.coverage(design, reps = 500)
# Plot the coverage
plot(design)
# Display the design statistics
design

#Multi-strata line transect example
shapefile.name <- system.file("extdata", "AreaRProjStrata.shp", package = "dssd")
region <- make.region(region.name = "study area",
                      strata.name = c("North", "NW", "West Upper",
                                      "West Lower", "SW", "South"),
                      shape = shapefile.name)
plot(region)
# Make a coverage grid
cover <- make.coverage(region,
                       n.grid.points = 500)
# Define the design
design <- make.design(region = region,
                      transect.type = "line",
                      design = c("systematic", "systematic",
                                 "eszigzag", "systematic",
                                 "systematic", "eszigzagcom"),
                      line.length = 5000*1000, #5000km x 1000m (projection in m)
                      design.angle = c(160, 135, 170, 135, 50, 60),
                      edge.protocol = "minus",
                      truncation = 3000,
                      coverage.grid = cover)

# Create a single set of transects to check
survey <- generate.transects(design)
plot(region, survey, covered.area = TRUE)

# Warning! this will quite a long time to run as it is a complex example.
design <- run.coverage(design, reps = 500)
```

```
# Plot the coverage
plot(design)
# Display the design statistics
design


# Fast running example for CRAN testing purposes
# This spacing is too sparse to assess coverage in a real example and
# the number of repetitions is too low to assess design statistics
cover <- make.coverage(region,
                       n.grid.points = 50)
design <- make.design(region = region,
                      transect.type = "point",
                      design = "random",
                      samplers = 25,
                      design.angle = 45,
                      edge.protocol = "minus",
                      truncation = 3,
                      coverage.grid = cover)
survey <- generate.transects(design)
plot(region, survey, covered.area = TRUE)
design <- run.coverage(design, reps = 3)
plot(design)
design
```

`make.region` *Creates a Region object*

Description

This creates an instance of the Region class which defines the study area for the survey.

Usage

```
make.region(region.name = "region", strata.name = character(0),
  units = character(0), shape = NULL)
```

Arguments

`region.name`	the region name
`strata.name`	the stratum names (character vector, same length as the number of areas in the shapefile / sf object). If not supplied "A", "B", "C", ... will be assigned.
`units`	measurement units; either `"m"` for metres or `"km"` for kilometres. If the shapefile has a projection file associated with it the unit will be taken from there.
`shape`	shapefile path to .shp file or an sf object of class sf, sfc or sfg.

Value

object of class Region

Author(s)

Laura Marshall

Examples

```
# A basic study rectangular study region
region <- make.region()
plot(region)

#Load the region from a projected shapefile
shapefile.name <- system.file("extdata", "TrackExample.shp", package = "dssd")
region <- make.region(region.name = "study area",
                      shape = shapefile.name)
plot(region)

#Load a multi strata unprojected shapefile
shapefile.name <- system.file("extdata", "AreaRStrata.shp", package = "dssd")
# Need to load shapefile first as it is not projected
sf.shape <- sf::read_sf(shapefile.name)
# Check current coordinate reference system
sf::st_crs(sf.shape)
# Define a European Albers Equal Area projection
proj4string <- "+proj=aea +lat_1=43 +lat_2=62 +lat_0=30 +lon_0=-9 +x_0=0 +
                y_0=0 +ellps=intl +units=km"
# Project the study area on to a flat plane
projected.shape <- sf::st_transform(sf.shape, crs = proj4string)
# Create region with default strata names
region <- make.region(region.name = "study area",
                      shape = projected.shape)
# By plotting the region we can verify the order of the strata
plot(region)
```

---

plot,Coverage.Grid,ANY-method
Plot

Description

Plots an S4 object of class 'Coverage.Grid'

Usage

```
## S4 method for signature 'Coverage.Grid,ANY'
plot(x, y, ...)
```

Arguments

x	object of class Coverage.Grid
y	not used
...	other general plot parameters

`plot,Line.Transect,ANY-method`
Plot

Description

Plots an S4 object of class 'Transect'

Plots an S4 object of class 'Transect'

Usage

```
## S4 method for signature 'Line.Transect,ANY'
plot(x, y, ...)

## S4 method for signature 'Point.Transect,ANY'
plot(x, y, ...)
```

Arguments

x	object of class Transect
y	not used
...	other general plot parameters

`plot,Region,ANY-method`
Plot

Description

Plots an S4 object of class 'Region'

Usage

```
## S4 method for signature 'Region,ANY'
plot(x, y, main = "", cols = "default",
  legend.params = list(inset = c(-0.2, 0), cex = 0.75, wrap = 15), ...)

## S4 method for signature 'Region,Transect'
plot(x, y, main = "",
  region.col = "default", ...)

## S4 method for signature 'Region,Coverage.Grid'
plot(x, y, main = "",
  region.col = "default", ...)
```

Arguments

x	object of class Region or inheriting from Survey
y	optionally a Survey object to plot with the Region
main	the main title for the plot
cols	colours for the strata
legend.params	a list of parameters which affect the location and appearance of the legend. 'inset' affects the location of the legend, 'cex' affects the text size and 'wrap' is the number of character in a line before the text is wrapped on to the next line.
...	other general plot parameters
region.col	fill colours for strata

---

plot,Survey.Design,ANY-method
Plot

Description

Plots an S4 object of class 'Survey.Design'

Usage

```
## S4 method for signature 'Survey.Design,ANY'
plot(x, y, ...)
```

Arguments

x	object of class Survey.Design
y	not used
...	other general plot parameters

Point.Transect-class *Class "Point.Transect" extends Class "Survey"*

Description

Virtual Class "Point.Transect" is an S4 class detailing a set of transects from a point transect design.

See Also

make.design

Point.Transect.Design-class *Virtual Class "Point.Transect.Design" extends Class "Survey.Design"*

Description

Virtual Class "Point.Transect.Design" is an S4 class detailing the type of point transect design.

Methods

generate.transects signature=(object = "Point.Transect.Design",...): generates a set of transects from a shapefile.

See Also

make.design

Region-class *Class "Region"*

Description

Class "Region" is an S4 class containing descriptions of the study area. Uses an object of class

Slots

region.name Object of class "character"; giving the name of the region.

strata.name Object of class "character"; character vector giving the names of the strata.

units Object of class "character"; character describing the coordinate units ("km" or "m")

area Object of class "numeric"; the area of the survey region

region Object of class "sf" defining the survey region

Objects from the Class

Objects can be created by calls of the form `make.region(region.name = "region.name",shapefile = region.shapefile)`

Methods

`get.area signature(obj = "Region")`: retrieves the area element

`plot signature(x = "Region",y = "missing")`: plots the survey region defined by the object.

See Also

`make.region`

`run.coverage` *run.coverage*

Description

This function can be used to assess the coverage of a design and also assess design statistics, such as how the number of samplers, the line length, trackline length or percentage coverage varies between surveys generated from the same design. It generates the specified number of surveys from the design and looks to see which of the coverage grid points, a systematic grid of points across the survey region, are included in each survey. When calculating coverage scores if more than one sampler falls on a grid point then that grid point gets allocated the appropriate count. These counts are then averaged over the number of surveys which have been generated. At the same time it records the relevant statistics for the design. While 100 repetitions may be sufficient to get an idea of design statistics 1000 or even more repetitions may be needed to gain a good representation of the coverage scores across the study region.

Usage

```
run.coverage(design, reps = 10, save.transects = "")
```

Arguments

`design`	an object which inherits from the Survey.Design class.
`reps`	the number of times you wish the coverage simulation to be carried out.
`save.transects`	a directory where the shapefiles for the transects can be saved. The shapefile names will be S1, S2, ... existing files in the directory will not be overwritten.

show,Line.Transect-method
Show

Description

Displays details of an S4 object of class 'Transect'

Displays details of an S4 object of class 'Transect'

Usage

```
## S4 method for signature 'Line.Transect'
show(object)

## S4 method for signature 'Point.Transect'
show(object)
```

Arguments

object	an object of class Transect

show,Survey.Design-method
show

Description

Summarises and displays an S4 object of class 'Survey.Design'

Usage

```
## S4 method for signature 'Survey.Design'
show(object)
```

Arguments

object	an object which inherits from the Survey.Design class
...	other general plot parameters

Survey.Design-class *Virtual Class "Survey.Design"*

Description

Virtual Class "Survey.Design" is an S4 class detailing the survey design.

Slots

region An object of class 'Region' defining the study area.

design Character value describing the name of the design.

samplers Numeric values defining the number of samplers in each stratum.

effort.allocation numeric values used to indicate the proportion of effort to be allocated to each strata from number of samplers or line length. If length 0, effort allocated based on stratum area.

spacing used by systematic designs, numeric value to define spacing between transects.

design.angle numeric value detailing the angle of the design. Can provide multiple values relating to strata. The use of the angle varies with design, it can be either the angle of the grid of points, the angle of lines or the design axis for the zigzag design.

edge.protocol Character value defining whether a "minus" or "plus" sampling strategy should be used.

truncation Object of class "numeric"; The maximum distance at which observations can be made. This is used to determine the covered area during the coverage calculations.

coverage.grid The coverage grid used to assess the uniformity of coverage during simulations.

coverage.scores The average number of times each point in the coverage grid is included in a survey.

coverage.reps The number of times the coverage simulation was repeated.

design.statistics A list of values obtained when investigating coverage. This includes the minimum, maximum, mean and median

See Also

make.design

`Transect-class` *S4 Class "Transect"*

Description

Virtual Class "Transect"

Details

Virtual Class `"Transect"` is an S4 class detailing a single survey, a single set of transects.

Slots

`strata.names` a character vector of the strata names

`design` Describes the design algorithm used to create the survey.

`samplers` Contains the survey transects

`strata.area` The areas of the strata in the design

`cov.area` The total areas sampled within each strata. Areas sampled twice are counted twice.

`cov.area.polys` The polygons representing the covered area of the survey.

`samp.count` Numeric value(s) giving the number of realised transects.

`effort.allocation` a vector of probabilities determining how effort is allocated between strata. Effort allocated based on area if left empty.

`spacing` determines the spacing of systematic samplers

`design.angle` numeric value detailing the angle of the design. Can provide multiple values relating to strata. The use of the angle varies with design, it can be either the angle of the grid of points, the angle of lines or the design axis for the zigzag design.

`edge.protocol` character value indicating whether a "plus" sampling or "minus" sampling protocol is used.

See Also

`make.design`

`write.transects` *Writes transects to file*

Description

This function will write the geographic information inside an object inheriting from class Transect to file. Currently the only options is to write to shapefile. It is basically a simple wrapper around the sf::st_write function. The sf::wt_write function can be used directly on the samplers slot of the Transect object if more options are required.

Usage

```
write.transects(object, dsn)
```

Arguments

`object`	and object inheriting from class Transect
`dsn`	the data source name, for this simple function this is a pathway and filename with a '.shp' extension.

Value

invisibly the Transect object

Author(s)

Laura Marshall

Examples

```
# Make the default design in the default study area
design <- make.design()
transects <- generate.transects(design)
write.transects(transects, dsn = paste0(tempdir(), "/", "transects.shp"))
```

Index