Package ‘DIGSS’

August 4, 2021

Title Determination of Intervals Using Georeferenced Survey Simulation

Version 1.0.2

Date 2021-08-03

Imports viridis,ggplot2,stats,grDevices,graphics,utils

Description Simulation tool to estimate the rate of success that surveys possessing user-specific characteristics have in identifying archaeological sites (or any groups of clouds of objects), given specific parameters of survey area, survey methods, and site properties. The survey approach used is largely based on the work of Kintigh (1988) <doi:10.2307/281113>.

Maintainer Mark Hubbe <hubbe.1@osu.edu>

BugReports https://github.com/markhubbe/DIGSS/issues

URL https://github.com/markhubbe/DIGSS

License MIT + file LICENSE

Depends R (>= 3.5)

Encoding UTF-8

RoxygenNote 7.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Mark Hubbe [aut, cre], Cara Hubbell [aut], William Pestle [aut]

Repository CRAN

Date/Publication 2021-08-04 14:30:05 UTC

R topics documented:

areaEstimator ............................................................... 2
cloudGenerator ............................................................. 3
fieldMap ................................................................. 4
areaEstimator

Description
Estimate the area of multiple overlapping ellipses

Usage
areaEstimator(sitemap, fieldarea, precision = 1000)

Arguments
- sitemap: a matrix with sites per row and columns: site_number, site_area, eccentricity, angle, center.x, center.y, ellipse.a, ellipse.b
- fieldarea: vector with dimensions of field surveyed in km: c(x_size, y_size)
- precision: how many dots will be projected of field. Total dots equal precision * precision. Default value = 1000 (1 million dots projected)

Details
This function will estimate the area occupied by sites (ellipses) in a rectangular field, taking into consideration the fact that sites can overlap. It is formatted to be used inside fieldMap(). This function is a cookie-cutter area estimator, given the complexities of calculating the real areas of overlapping ellipses. It projects N x N equally spaced dots in the survey field and calculates the ratio of how many of them fall inside at least one site (ellipse). Using a precision of 1000 x 1000 dots, it approximates area to within 0.1% of real area.

Value
The rate of points that are inside at least one ellipse divided by all points projected in the area.

Examples
# create a matrix with 2 sites randomly located using `fieldMap()`
site.example<-fieldMap(c(1,1),2,250000,plot=TRUE)

# define size of field
field.area<-c(1,1)

# calculate area
areaEstimator(site.example$site.frame,field.area)
cloudGenerator  

Description

Creates a cloud of dots inside ellipsoid sites

Usage

```r
cloudGenerator(
  density, a, b, angle, center.x, center.y,
  type = "uniform", precision = 30, plot = FALSE
)
```

Arguments

density  
dots (artifacts) per \( m^2 \)

a  
ellipse (site) long axis in km

b  
ellipse (site) short axis in km

angle  
ellipse (site) angle of rotation in radians

center.x  
center of ellipse in x axis

center.y  
center of ellipse in y axis

type  
type of density distribution. Choose from:

- uniform: dots are uniformly distributed within ellipse
- linear: density of dots decreases linearly from center
- spherical: density of dots decreases following elliptical function (abrupt drop near margins)
- sinusoidal: density of dots decreases following sinusoidal equation

precision  
how many slices of the distribution will be made (more = much slower run times). Default = 30

plot  
if function should plot results. function does not work outside surveySim()
cloudGenerator() creates a cloud of point inside an ellipsoid site of predefined size and shape, to represent the locations of artifacts in a site. The function can build artifact scatters with different densities profiles. The function uses an "onion-layer" approach to approximate the density of points from the center. In practice, it means that each site is composed of N ellipse slices surrounding the previous slice, with each slice having a different artifact density depending on the density function selected. This approach also makes the surveySim function more efficient, since it will search for artifact hits only on the slices that intersect the survey pits.

Value

A list with two objects:

- coords A list of the bands (N=precision) that represent the site, each populated with the X and Y coordinates for all artifacts generated inside them.
- info A list with the number of pieces created ($total.pieces), area of the site ($total.area), and artifact density in the site ($actual.density)

Examples

```r
# create a small site with low density uniform distribution
uni.site<-cloudGenerator(0.1,0.1,0.05,pi/4,0.5,0.5,type="u")

# plot a site with uniform artifact distribution through surveySim
SiteParameters<-parametersExample
SiteParameters$simulations=1
SiteParameters$site.density=1
SiteParameters$obj.density=0.1
SiteParameters$obj.distribution = "u"
surveySim(SiteParameters,plot.artifacts = TRUE)

# plot a site with sinusoidal artifact distribution through surveySim
SiteParameters$obj.distribution = "si"
surveySim(SiteParameters,plot.artifacts = TRUE)
```

fieldMap

Field Map

Description

Creates randomly placed ellipsoid sites in a rectangular field.

Usage

```r
fieldMap(
  area,
  site.density,
)```
fieldMap

    site.area,
    overlap = 0.5,
    plot = FALSE,
    areaprecision = 1000
)

Arguments

area vector with horizontal and vertical size (hor,ver) of area surveyed in km.
site.density number of sites/km². Can be one constant value or vector with two values (min, max) to create a range of densities between simulations.
site.area either:
  • One values with uniform area for all sites, or
  • Vector with 4 values (min, max, mean, st dev), to create variable areas.
    areas in this case are normally distributed based on mean and stdev, but
    within the range of min and max.
overlap proportion of overlap possible between sites: from (0 = no overlap allowed to 1
  = sites can occupy same space)
plot whether site ellipses should should be plotted.
areaprecision value passed to areaEstimator. Defines precision of area calculation. Default
value (1000), returns area within 0.1% of real area occupied by sites

Details

fieldMap() creates and plots randomly placed ellipses representing archaeological sites. The sites
are created inside a user-defined rectangle, with random positions and random rotations. It allows
also to control the percentage of overlap between sites.

Value

A list with three objects:
site.frame A matrix with the properties of each site generated.
totalArea The sum of areas of all site ellipses.
actualArea The area occupied by the site ellipses taking into account their overlap)

Examples

# example of map with 8 sites or variable areas and partial overlap
field.example<-fieldMap(
    area=c(1,1),
    site.density=8,
    site.area=c(50000,250000,150000,50000),
    overlap=0.5,
    plot=TRUE)
**parametersExample**  

**Parameters Example**

---

**Description**

A test list of parameters for `surveySim()`

**Usage**

`parametersExample`

**Format**

An object of class `surveySim` of length 10.

**Details**

This is just a test list of parameters to be used as an example in `surveySim()`.

The values of this parameters are:

- Width between survey lines (`col.width`) = **50 m**
- Type of survey grid (`grid.type`) = **hexagonal**
- Number of simulations = **10**
- Survey area = **0.5km x 0.5km**
- Density of sites (`site.density`) = **20**
- Area of sites (`site.area`) = **10,000 m²**
- Maximum site overlap = **0.5**
- Density of artifacts (`obj.density`) = **1/m²**
- Artifact distribution (`obj.distribution`) = **spherical**
- Survey radius (`survey.radius`) = **0.5m**
Description

Plots the different results from surveySim()

Usage

plotSurveySumm(summaryList, plot = "sites.found", labels = NULL)

Arguments

summaryList  a list of survey summaries, the output of surveySim()
plot          what variable to plot. Options are:
                   • sites.found - plots frequency of sites found
                   • survey.hits - plots frequency of surveys that hit a site
                   • success.rate.index - plots the success rate of surveys, i.e. the ratio of
                                 successful surveys over total surveys
                   • sites.found.arti - plots frequency of sites found based on artifacts
                                 present in sites
                   • survey.hits.arti - plots frequency of surveys that found at least one
                                 artifact
labels       vector with name of each item in list, to be added to the legend. If NULL, names
             will be taken from list

Details

This function will plot the results of the surveySim() simulations using Kernel Density plots. All
the grids that are to be compared should be grouped into 1 list (list(a,b,c,etc...)) The plot function
allows you to choose different parts of the survey summaries produced by surveySim() you want
to plot.

Examples

# create 3 Simulations with sites of different sizes:
small.sites<-parametersExample
small.sites$site.area=500

medium.sites<-parametersExample
medium.sites$site.area=1000

large.sites<-parametersExample
large.sites$site.area=2000

# run the 3 simulations
small.survey<-surveySim(small.sites)
medium.survey<-surveySim(medium.sites)
large.survey<-surveySim(large.sites)

#create the comparative plot.
#note that the results go into a list. If labels are not given, legend is built on list names
plotSurveySumm(
  list(small.survey,medium.survey,large.survey),
  plot="sites.found",
  labels=c("Small sites","Medium sites","Large sites"))

---

**surveyLoops**

**Survey Loops**

**Description**
Perform multiple survey simulations changing values on one variable and plot the results

**Usage**
surveyLoops(surveyParameters, loopVariable, loopSequence, plotResult)

**Arguments**

- **surveyParameters**
  - list of parameters (object class surveySim)

- **loopVariable**
  - variable to be looped. Can be any of the variables that exist in surveyParameters
    - col.width - vector of numbers with distances between STP rows
    - grid.type - vector of strings with names of grid types
    - simulations - vector of numbers with numbers of simulations
    - area - list with 2 vectors, one for x and one for y of area (vectors MUST be same length)
    - site.density - vector with numbers of site density OR list with 2 vectors, one with minimum site densities, one with maximum (vectors MUST be same length)
    - site.area - vector with numbers of site areas OR list with 4 vectors, one with minimums, one with maximums, one with means, one with st.devs (vectors MUST be same length)
    - overlap - vector of numbers with varying overlaps
    - obj.density - vector with varying artifact density OR list with 2 vectors, one with minimum artifact densities, one with maximum (vectors MUST be same length)
    - obj.distribution - vector of names of artifact distribution
    - survey.radius vector with varying survey radii
loopSequence object with varying values, as defined above
plotResult which results from the summary table will be plotted:
- surveysPerSim - plots the total number of surveys created in each simulation
- sitesFound - plots the frequency of sites found
- sitesFoundOnArtifacts - plots the frequency of sites found based on surveys finding at least one artifact
- artifactsPerSurvey - plots the average number of artifacts found per survey in each simulation
- successRateIndex - plots the success rate index (ratio of survey pits that found a site over total number of survey pits made)

Details
surveyLoops() will run a series of simulations along one variable with values provided by user. Through this function, the user can simulate and evaluate the changes in efficiency and efficacy of specific variables, while holding every other value constant. The function runs multiple instances of surveySim() using values of surveyParameters and replacing one of them with a sequence of values offered by the user.

Value
A list with five objects:
surveysPerSim A matrix with the number of survey pits done in each simulation.
sitesFound A matrix with the summary statistics about frequency of sites found in each simulation.
sitesFoundOnArtifacts A matrix with the summary statistics about frequency of sites detected based on artifacts found in each simulation.
artifactsPerSurvey A matrix with the summary statistics about artifacts found per survey pit in each simulation.
successRateIndex A matrix with the summary statistics about success rate (number of successful survey pits/total survey pits made)

Examples
#Loop the impact of increasing distances between survey rows
width.loop<-surveyLoops(parametersExample,"col.width",c(50,75,100,125,150),"sitesFound")

#Loop the impact of different artifact distributions on surveys
distr.loop<-surveyLoops(
    parametersExample,
    "obj.distribution",
    c("uniform","linear","spherical","sinusoidal"),
    "sitesFoundOnArtifacts")
surveySim

Survey Simulations

Description

Master function that runs survey simulations

Usage

surveySim(
  survey.parameters,
  artifact.analysis = TRUE,
  plot = TRUE,
  plot.artifacts = FALSE,
  areaprecision = 1000
)

Arguments

survey.parameters

List of class surveySim. See package's vignette or the help of parametersExample
for details. The following items must be included in list of class surveySim

- col.width the space between columns in the grid IN METERS
- grid.type options are: "square", "rectangle", "staggered", "hexagonal", "arbitrary.staggered", following Kintigh 1988
- simulations number of random maps to be created and contrasted with the grids
- area vector with horizontal and vertical size of area surveyed in km OBS: Sites will all be ellipses with radii not too different and random angles
- site.density measured as number of sites/km^2. Can be either one value or a vector with 2 values (min and max) to create a range of densities
- site.area can be one of two options: 1. one value indicating the area of all sites, in m^2; or 2. a vector with 4 values: min, max, mean (or median), and standard deviation in m^2
- overlap maximum overlap of site area, ranging from 0 = no overlap allowed to 1 = complete overlap possible
- obj.density artifacts per m^2. Can be a single value (uniform for all sites) or a range of values defined as min and max
- obj.distribution type of cloud distribution for artifacts inside sites. Choose from: 'uniform', 'linear', 'spherical', 'sinusoidal'
- survey.radius the radius of the survey pit (assumed to be a circle)

artifact.analysis

Includes the analysis of artifacts in sites. Can be very time-consuming for mid-high artifact densities. Default = TRUE

plot

If TRUE the last iteration of the simulations will be plotted.
plot.artifacts  If TRUE the artifacts in each site will be plotted. Default = FALSE because it is resource intensive for mid-high densities.

areaprecision Area covered by sites is approximated by a cookie-cutter approach. Default precision = 1000 gives approximate area within 1% error. Higher values reduce error but are more resource intensive.

Details

surveySim() is the main function that will conduct the simulations of survey grids based on the Survey Parameters created by user.

Value

A list with four objects:

- **Summary**: A matrix with summary statistics about number of surveys, frequency of site founds, artifacts presence, and success rate of simulations.
- **BySite**: A matrix with results of the analyses by site from each of the survey areas created in the simulation.
- **ByArtifact**: A matrix with results of the analyses by artifact presence in survey pits from each of the survey areas created in the simulation.
- **Parameters**: A list with the parameters used to run the simulation.

Examples

```r
# Runs simulations based on parametersExample
surveyExample <- surveySim(parametersExample)

# Edit parametersExample to have 50 simulations and then run simulations
tmp_parameters <- parametersExample
tmp_parameters$simulations <- 50
survey50 <- surveySim(tmp_parameters)
```

Index

* datasets
  - parametersExample, 6

areaEstimator, 2
cloudGenerator, 3
fieldMap, 4
parametersExample, 6
plotSurveySumm, 7
surveyLoops, 8
surveySim, 10