Package ‘spcosa’

Type Package

Title Spatial Coverage Sampling and Random Sampling from Compact Geographical Strata

Version 0.3-5

Date 2013-05-01

Description Spatial coverage sampling and random sampling from compact geographical strata created by k-means.

Depends R (>= 2.15.1), rJava (>= 0.9-3), methods, ggplot2 (>= 0.9.2), utils

Suggests grid, gstat, rgdal, rgl, RUnit

Imports sp (>= 0.9-97)

SystemRequirements Java (>= 6.0)

License GPL (>= 3)

## R topics documented:

<table>
<thead>
<tr>
<th>Package/Class</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>spcosa-package</td>
<td>2</td>
</tr>
<tr>
<td>CompactStratification-class</td>
<td>3</td>
</tr>
<tr>
<td>CompactStratificationEqualArea-class</td>
<td>5</td>
</tr>
<tr>
<td>CompactStratificationPriorPoints-class</td>
<td>6</td>
</tr>
<tr>
<td>estimate-methods</td>
<td>6</td>
</tr>
<tr>
<td>getArea-methods</td>
<td>8</td>
</tr>
<tr>
<td>getCentroid-methods</td>
<td>8</td>
</tr>
<tr>
<td>getNumberOfStrata-methods</td>
<td>8</td>
</tr>
<tr>
<td>getObjectiveFunctionValue-methods</td>
<td>8</td>
</tr>
<tr>
<td>getRelativeArea-methods</td>
<td>9</td>
</tr>
<tr>
<td>getSampleSize-methods</td>
<td>9</td>
</tr>
<tr>
<td>plot-methods</td>
<td>9</td>
</tr>
<tr>
<td>SamplingPattern-class</td>
<td>10</td>
</tr>
<tr>
<td>SamplingPatternCentroids-class</td>
<td>10</td>
</tr>
<tr>
<td>SamplingPatternPriorPoints-class</td>
<td>11</td>
</tr>
<tr>
<td>SamplingPatternPurposive-class</td>
<td>12</td>
</tr>
<tr>
<td>SamplingPatternRandom-class</td>
<td>12</td>
</tr>
<tr>
<td>SamplingPatternRandomComposite-class</td>
<td>13</td>
</tr>
<tr>
<td>SamplingPatternRandomSamplingUnits-class</td>
<td>14</td>
</tr>
<tr>
<td>SamplingVariance-class</td>
<td>15</td>
</tr>
<tr>
<td>SpatialCumulativeDistributionFunction-class</td>
<td>15</td>
</tr>
<tr>
<td>SpatialMean-class</td>
<td>16</td>
</tr>
<tr>
<td>SpatialVariance-class</td>
<td>17</td>
</tr>
<tr>
<td>spsample-methods</td>
<td>17</td>
</tr>
<tr>
<td>StandardError-class</td>
<td>18</td>
</tr>
<tr>
<td>Statistic-class</td>
<td>19</td>
</tr>
<tr>
<td>Stratification-class</td>
<td>20</td>
</tr>
<tr>
<td>stratify-methods</td>
<td>20</td>
</tr>
</tbody>
</table>

### Index

<table>
<thead>
<tr>
<th>Package/Class</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>spcosa-package</td>
<td>23</td>
</tr>
</tbody>
</table>

### Description

Algorithms for spatial coverage sampling and for random sampling from compact geographical strata based on $k$-means.

### Details

The `spcosa-package` provides algorithms for spatial coverage sampling and for random sampling from compact geographical strata based on $k$-means (see de Gruijter *et al.*, 2006, Walvoort *et al.*, 2010, and `kmeans`). S4-classes and methods are available for spatial coverage sampling, random sampling from compact geographical strata, and...
stratified simple random sampling for composites. In case of spatial coverage sampling, existing sampling points may be taken into account. See the package vignette for more information and examples.

Note

In order to get the spcosa-package running, make sure that a recent version of Java (>= 6.0) is installed. Free Java downloads are available at http://www.java.com.

In case of problems, you may wish to consult the FAQ located at C:\Temp\RtmpgvKycf\Rinst66c6098489\spcosa\FAQ

Author(s)

D.J.J. Walvoort, D.J. Brus, J.J. de Gruijter,
Maintainer: Dennis Walvoort <dennis.walvoort@wur.nl>

References


Walvoort, D. J. J., Brus, D. J. and de Gruijter, J. J. (2010). An R package for spatial coverage sampling and random sampling from compact geographical strata by

\[ k \]


See Also

stratify for stratification, spsample for sampling, and estimate for inference.

CompactStratification-class

Class "CompactStratification"

Description

A class for storing a stratification with compact strata.

Objects from the Class

Objects can be created by calls of the form new("CompactStratification", cells, stratumId, centroids, mssd). However, objects are usually created by calling stratify.
**Slots**

- `cells`: Object of class "SpatialPixels", representing the area to be partitioned.
- `stratumId`: Object of class "integer", indicating to which stratum each cell in `cells` belong.
- `centroids`: Object of class "SpatialPoints", representing the centers of gravity of each stratum.
- `mssd`: Object of class "numeric", representing the mean squared shortest distance.

**Extends**

Class "Stratification", directly.

**Methods**

- `coerce` signature(from = "CompactStratification", to = "data.frame"): coerces to "data.frame".
- `coerce` signature(from = "CompactStratification", to = "SpatialPixels"): coerces to "SpatialPixels".
- `coerce` signature(from = "CompactStratification", to = "SpatialPixelsDataFrame"): coerces to "SpatialPixelsDataFrame".
- `estimate` signature(statistic = "SamplingVariance", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates the sampling variance. See "SamplingVariance" for more details.
- `estimate` signature(statistic = "SpatialMean", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates the spatial mean. See "SpatialMean" for more details.
- `estimate` signature(statistic = "SpatialVariance", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates the spatial variance. See "SpatialVariance" for more details.
- `getArea` signature(object = "CompactStratification"): returns the area of each stratum.
- `getCentroid` signature(object = "CompactStratification"): returns the center of gravity of each stratum.
- `getNumberOfStrata` signature(object = "CompactStratification"): returns the number of strata.
- `getObjectiveFunctionValue` signature(object = "CompactStratification"): extracts the mean squared shortest distance.
- `getRelativeArea` signature(object = "CompactStratification"): returns the relative area of each stratum. The sum of the relative areas equals one.
- `plot` signature(x = "CompactStratification", y = "missing"): plots stratification x.
- `plot` signature(x = "CompactStratification", y = "SamplingPattern"): plots sampling pattern y on top of stratification x.
- `plot` signature(x = "CompactStratification", y = "SamplingPatternPriorPoints"): plots sampling pattern y on top of stratification x.
- `plot` signature(x = "CompactStratification", y = "SamplingPatternRandomComposite"): plots sampling pattern y on top of stratification x.
CompactStratificationEqualArea-class

**Description**

A class for storing a stratification with compact strata of equal size.

**Objects from the Class**

Objects can be created by calls of the form `new("CompactStratificationEqualArea", cells, stratumId, centroids, mssd)`.

However, objects are usually created by calling `stratify`.

**Slots**

- `cells`: Object of class "SpatialPixels", representing the area to be partitioned.
- `stratumId`: Object of class "integer", indicating to which stratum each cell in `cells` belong.
- `centroids`: Object of class "SpatialPoints", representing the centers of gravity of each stratum.
- `mssd`: Object of class "numeric", representing the mean squared shortest distance.

**Extends**


**Methods**

- `estimate` signature(statistic = "SamplingVariance", stratification = "CompactStratificationEqualArea", samplingPattern = "SamplingPatternRandomComposite", data = "data.frame")
  estimates the sampling variance. See "SamplingVariance" for more details.

- `estimate` signature(statistic = "SpatialMean", stratification = "CompactStratificationEqualArea", samplingPattern = "SamplingPatternRandomComposite", data = "data.frame")
  estimates the spatial mean. See "SpatialMean" for more details.

- `spsample` signature(x = "CompactStratificationEqualArea", n = "missing", type = "missing")
  returns the centers of gravity of each stratum.

- `spsample` signature(x = "CompactStratificationEqualArea", n = "numeric", type = "missing")
  randomly selects `n` sampling points in each stratum.

- `spsample` signature(x = "CompactStratificationEqualArea", n = "numeric", type = "character")
  randomly selects `n` sampling points in each stratum. If `type = "composite"`, stratified simple random sampling of `n` composites.

**Author(s)**

Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter
CompactStratificationPriorPoints-class

Class "CompactStratificationPriorPoints"

Description
A class for storing a stratification with compact strata, given prior sampling locations.

Objects from the Class
Objects can be created by calls of the form new("CompactStratificationPriorPoints", cells, stratumId, centroids, mssd, priorPoints). However, objects are usually created by calling stratify.

Slots
- priorPoints: Object of class "SpatialPoints", containing the coordinates of the existing locations.
- cells: Object of class "SpatialPixels", representing the area to be partitioned.
- stratumId: Object of class "integer", indicating to which stratum each cell in cells belong.
- centroids: Object of class "SpatialPoints", representing the centers of gravity of each stratum.
- mssd: Object of class "numeric", representing the mean squared shortest distance.

Extends

Methods
- spsample signature(x = "CompactStratificationPriorPoints", n = "missing", type = "missing"): returns the centers of gravity of strata without prior points in addition to the prior points.

Author(s)
Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter

estimate-methods

Estimating Statistics

Description
Methods for estimating statistics given a spatial sample.
Methods

statistic = "character", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits"
estimates one of the following statistics, depending on the value of argument statistic: spatial mean, spatial variance, sampling variance, standard error, or scdf. See the examples below for details.

statistic = "character", stratification = "CompactStratificationEqualArea", samplingPattern = "SamplingPatternRandomComposite"
estimates one of the following statistics, depending on the value of argument statistic: spatial mean, sampling variance, or standard error.

statistic = "SamplingVariance", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits"
estimates the sampling variance. See "SamplingVariance" for more details.

statistic = "StandardError", stratification = "CompactStratificationEqualArea", samplingPattern = "SamplingPatternRandomComposite"
estimates the standard error of the spatial mean. See "StandardError" for more details.

statistic = "SpatialCumulativeDistributionFunction", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits"
estimates the spatial cumulative distribution function (SCDF). See "SamplingPatternRandomSamplingUnits" for more details.

statistic = "SpatialMean", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits"
estimates the spatial mean. See "SpatialMean" for more details.

statistic = "SpatialVariance", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits"
estimates the spatial variance. See "SpatialVariance" for more details.

Examples

# Note: the example below requires the 'rgdal'-package.
# You may consider the 'maptools'-package as an alternative
if (require(rgdal)) {
  # read vector representation of the "Mijdrecht" area
  shp <- readOGR(
    dsn = system.file("maps", package = "spcosa"),
    layer = "mijdrecht"
  )

  # stratify into 30 strata
  myStratification <- stratify(shp, nStrata = 30, nTry = 10, verbose = TRUE)

  # random sampling of two sampling units per stratum
  mySamplingPattern <- spsample(myStratification, n = 2)

  # plot sampling pattern
  plot(myStratification, mySamplingPattern)

  # simulate data
  # (in real world cases these data have to be obtained by field work etc.)
  myData <- as(mySamplingPattern, "data.frame")
  myData$observation <- rnorm(n = nrow(myData), mean = 10, sd = 1)

  # design-based inference
  estimate("spatial mean", myStratification, mySamplingPattern, myData["observation"])
  estimate("sampling variance", myStratification, mySamplingPattern, myData["observation"])
  estimate("standard error", myStratification, mySamplingPattern, myData["observation"])
  estimate("spatial variance", myStratification, mySamplingPattern, myData["observation"])
  estimate("scdf", myStratification, mySamplingPattern, myData["observation"])
}
**getArea-methods**  
*Extract the Area of an Object*

**Description**  
Methods for extracting the area of objects.

**Methods**  
object = "CompactStratification" returns the area of each stratum.

**See Also**  
getRelativeArea

**getCentroid-methods**  
*Extract Centroids*

**Description**  
Methods for extracting centroids

**Methods**  
object = "CompactStratification" returns the centers of gravity of each stratum.

**getNumberOfStrata-methods**  
*Extract the Number of Strata in an Object*

**Description**  
Methods for extracting the number of strata of objects.

**Methods**  
object = "CompactStratification" returns the number of strata in a compact stratification.

**getObjectiveFunctionValue-methods**  
*Extract the Objective Function Value of an Object*

**Description**  
Methods for extracting the objective function value

**Methods**  
object = "CompactStratification" extracts the mean squared shortest distance.
getRelativeArea-methods

Extract the Relative Area of an Object

Description

Methods for extracting relative areas of objects. The total area equals unity.

Methods

object = "CompactStratification" returns the relative area of each stratum. The sum of the relative areas equals 1.

See Also

getArea

ggetTextSampleSize-methods

Extract the sample size of an object

Description

Methods for extracting the sample size.

Methods

object = "SamplingPattern" returns the sample size.
object = "SamplingPatternRandomComposite" returns the number of composites

plot-methods

Visualizing Compact Stratifications and Sampling Patterns

Description

The plot method can be used to visualize compact stratifications and sampling patterns. Since it has been built on top of the ggplot2 package, functions provided by this package can be used to modify the plots.

Methods

x = "CompactStratification", y = "missing" plots stratification x.
x = "CompactStratification", y = "SamplingPattern" plots sampling pattern y on top of stratification x.
x = "CompactStratification", y = "SamplingPatternPriorPoints" plots sampling pattern y on top of stratification x.
x = "CompactStratification", y = "SamplingPatternRandomComposite" plots sampling pattern y on top of stratification x.
x = "SamplingPattern", y = "missing" plots sampling pattern x.
x = "SamplingPatternPriorPoints", y = "missing" plots sampling pattern x.
x = "SamplingPatternRandomComposite", y = "missing" plots sampling pattern x.
SamplingPattern-class  Class "SamplingPattern"

Description
A class for storing a sampling pattern.

Objects from the Class
Objects can be created by calls of the form new("SamplingPattern", ...). However, objects are usually created by calling `spsample`.

Slots
sample: Object of class "SpatialPoints", containing the sampling locations.

Methods
`coerce` signature(from = "SamplingPattern", to = "data.frame"): coerces to "data.frame".
`coerce` signature(from = "SamplingPattern", to = "SpatialPoints"): coerces to "SpatialPoints".
`getSampleSize` signature(object = "SamplingPattern"): returns the sample size.
`plot` signature(x = "CompactStratification", y = "SamplingPattern"): plots sampling pattern y on top of stratification x.
`plot` signature(x = "SamplingPattern", y = "missing"): plots sampling pattern x.
`show` signature(object = "SamplingPattern"): prints object on the output device.

Author(s)
Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter

SamplingPatternCentroids-class  Class "SamplingPatternCentroids"

Description
A class for storing a sampling pattern, where the sampling locations are the centers of gravity of each stratum.

Objects from the Class
Objects can be created by calls of the form new("SamplingPatternCentroids", ...). However, objects are usually created by calling `spsample`.
SamplingPatternPriorPoints-class

Slots

sample: Object of class "SpatialPoints", containing the sampling locations

Extends


Methods

No methods defined with class "SamplingPatternCentroids" in the signature.

Author(s)

Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter
**SamplingPatternPurposive-class**  
*Class "SamplingPatternPurposive"*

**Description**
An ancestor class for storing purposive sampling patterns.

**Objects from the Class**
Objects can be created by calls of the form `new("SamplingPatternPurposive", ...)`.

**Slots**
- `sample`: Object of class "SpatialPoints", containing the sampling locations

**Extends**
Class "SamplingPattern", directly.

**Methods**
No methods defined with class "SamplingPatternPurposive" in the signature.

**Author(s)**
Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter

---

**SamplingPatternRandom-class**  
*Class "SamplingPatternRandom"*

**Description**
An ancestor class for storing random sampling patterns.

**Objects from the Class**
Objects can be created by calls of the form `new("SamplingPatternRandom", ...)`.

**Slots**
- `sample`: Object of class "SpatialPoints", containing the sampling locations

**Extends**
Class "SamplingPattern", directly.

**Methods**
No methods defined with class "SamplingPatternRandom" in the signature.
SamplingPatternRandomComposite-class

Description
A class for storing composites obtained by random sampling.

Objects from the Class
Objects can be created by calls of the form new("SamplingPatternRandomComposite", ...). However, objects are usually created by calling spsample.

Slots
composite: Object of class "integer", indicating to which composite sample a sampling unit belongs.
sample: Object of class "SpatialPoints", containing the sampling locations.

Extends

Methods

coerce signature(from = "SamplingPatternRandomComposite", to = "data.frame"): coerces to "data.frame".

coerce signature(from = "SamplingPatternRandomComposite", to = "SpatialPointsDataFrame"): coerces to "SpatialPointsDataFrame".


estimate signature(statistic = "SpatialMean", stratification = "CompactStratificationEqualArea", samplingPattern = "SamplingPatternRandomComposite", data = "data.frame"): estimates the spatial mean. See "SpatialMean" for more details.

gSampleSize signature(object = "SamplingPatternRandomComposite"): returns the sample size per stratum.

plot signature(x = "CompactStratification", y = "SamplingPatternRandomComposite"): plots sampling pattern y on top of stratification x.

plot signature(x = "SamplingPatternRandomComposite", y = "missing"): plots sampling pattern x.

Author(s)
Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter
SamplingPatternRandomSamplingUnits-class

Class "SamplingPatternRandomSamplingUnits"

Description

A class for storing sampling units obtained by random sampling.

Objects from the Class

Objects can be created by calls of the form `new("SamplingPatternRandomSamplingUnits", ...)`. However, objects are usually created by calling `spsample`.

Slots

- `sample`: Object of class "SpatialPoints", containing the sampling locations.

Extends


Methods

- `estimate` signature(statistic = "SamplingVariance", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates the sampling variance. See "SamplingVariance" for more details.
- `estimate` signature(statistic = "SpatialMean", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates the spatial mean. See "SpatialMean" for more details.
- `estimate` signature(statistic = "SpatialVariance", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates the spatial variance. See "SpatialVariance" for more details.
- `estimate` signature(statistic = "character", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates statistic, i.e., "spatial mean", "spatial variance", "sampling variance", "standard error", SCDF.

Author(s)

Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter
SamplingVariance-class

Class "SamplingVariance"

Description


Objects from the Class

Objects can be created by calls of the form new("SamplingVariance", ...).

Slots

description: Object of class "character" A description of the statistic.

Extends

Class "Statistic", directly.

Methods

estimate signature(statistic = "SamplingVariance", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame")

estimates the sampling variance, given a stratification, a sampling pattern and data.

estimate signature(statistic = "SamplingVariance", stratification = "CompactStratificationEqualArea", samplingPattern = "SamplingPatternRandomComposite", data = "data.frame")

estimates the sampling variance, given a stratification, a sampling pattern and data.

Author(s)

Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter

References


SpatialCumulativeDistributionFunction-class

Class "SpatialCumulativeDistributionFunction"

Description

The spatial cumulative distribution function (SCDF) is estimated by applying Equation 7.13 in de Gruijter et al., (2006) to indicator transformations of the data. See also page 83 of de Gruijter et al., (2006).

Objects from the Class

Objects can be created by calls of the form new("SpatialCumulativeDistributionFunction", ...).
Slots

description: Object of class "character" A description of the statistic.

Extends

Class "Statistic", directly.

Methods

estimate signature(statistic = "SpatialCumulativeDistributionFunction", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame")
estimates the spatial cumulative distribution function (SCDF), given a stratification, a sampling pattern and data.

Author(s)

Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter

References

SpatialVariance-class

Description

The spatial variance is estimated by means of Equation 7.16 in de Gruijter et al., (2006).

Objects from the Class

Objects can be created by calls of the form `new("SpatialVariance", ...)`. 

Slots

description: Object of class "character" A description of the statistic.

Extends

Class "Statistic", directly.

Methods

- `estimate` signature(statistic = "SpatialVariance", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame") estimates the spatial variance, given a stratification, a sampling pattern and data.

Author(s)

Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter

References

Methods

- `x = "CompactStratification", n = "missing", type = "missing"` samples the centroids of each stratum.
- `x = "CompactStratification", n = "numeric", type = "missing"` stratified simple random sampling with `n` samples per stratum.
- `x = "CompactStratificationEqualArea", n = "numeric", type = "character"` if `type = "composite"`, stratified simple random sampling of `n` composites.
- `x = "CompactStratificationPriorPoints", n = "missing", type = "missing"` spatial infill sampling

See Also

- `stratify` for stratification, `spsample` for other types of spatial sampling, and `estimate` for inference.

Examples

```r
# Note: the example below requires the 'rgdal'-package.
# You may consider the 'maptools'-package as an alternative
if (require(rgdal)) {

  # read a vector representation of the 'Farmsum' field
  shpFarmsum <- readOGR(
    dsn = system.file("maps", package = "spcosa"),
    layer = "farmsum"
  )

  # stratify 'Farmsum' into 50 strata
  # NB: increase argument 'nTry' to get better results
  set.seed(314)
  myStratification <- stratify(shpFarmsum, nStrata = 50, nTry = 1)

  # sample two sampling units per stratum
  mySamplingPattern <- spsample(myStratification, n = 2)

  # plot the resulting sampling pattern on
  # top of the stratification
  plot(myStratification, mySamplingPattern)
}
```

---

**StandardError-class**  

**Class** "StandardError"

**Description**


**Objects from the Class**

Objects can be created by calls of the form `new("StandardError", ...)`. 
**Statistic-class**

**Slots**
- **description**: Object of class "character" A description of the statistic.

**Extends**

**Methods**
- **estimate** signature(statistic = "StandardError", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame")
  estimates the standard error, given a stratification, a sampling pattern and data.

**Author(s)**
- Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter

**References**

---

**Statistic-class**

**Class "Statistic"**

**Description**
- A superclass (ancestor class) for statistics to estimate.

**Objects from the Class**
- A virtual Class: No objects may be created from it.

**Slots**
- **description**: A description of the statistic

**Methods**
- **show** signature(object = "Statistic"): prints the statistic

**Author(s)**
- Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter
Stratification-class  Class "Stratification"

Description
Virtual class to store a spatial stratification.

Objects from the Class
A virtual Class: No objects may be created from it.

Methods
- show  signature(object = "Stratification"): a method for printing objects of class Stratification

Author(s)
Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter

Examples
```
showClass("Stratification")
```

stratify-methods  Stratification

Description
Methods for partitioning a spatial object into compact strata by means of 

\[ k \]

-k-means. The objective function to minimize is the mean squared shortest distance (MSSD). Optionally, the strata may be forced to be of equal size. This facilitates field work in case of stratified simple random sampling for composites. Another option is spatial infill sampling, a variant of spatial coverage sampling where existing sampling points are taken into account. Use nTry > 1, to reduce the risk of ending up in an unfavorable local optimum. Better results will generally be obtained by increasing the ratio nGridCells/nStrata and by increasing nTry.

Usage
```
## S4 method for signature 'SpatialPixels'
stratify(object, nStrata, priorPoints = NULL, maxIterations = 1000, nTry = 1,
        equalArea = FALSE, verbose = getOption("verbose"))

## S4 method for signature 'SpatialGrid'
stratify(object, nStrata, priorPoints = NULL, maxIterations = 1000, nTry = 1,
        equalArea = FALSE, verbose = getOption("verbose"))

## S4 method for signature 'SpatialPolygons'
stratify(object, nStrata, priorPoints = NULL, maxIterations = 1000, nTry = 1,
        nGridCells = 2500, cellSize, equalArea = FALSE, verbose = getOption("verbose"))
```
Arguments

- **object**: an object of class "SpatialPixels", "SpatialGrid" or "SpatialPolygons"
- **nStrata**: number of strata (nStrata >= 1).
- **priorPoints**: object of class "SpatialPoints", containing the prior (i.e., existing) points
- **maxIterations**: maximum number of iterations.
- **nTry**: the stratify method will try nTry initial configurations and will keep the best solution in order to reduce the risk of ending up with an unfavorable solution.
- **nGridCells**: in case object is an instance of class "SpatialPolygons", the approximate number of grid cells to be used for discretizing the vector map in object.
- **cellSize**: in case object is an instance of class "SpatialPolygons", the cell size to be used for discretizing the vector map in object. Note that cellsize takes precedence over argument nGridCells.
- **equalArea**: If FALSE the algorithm results in compact strata. If TRUE, the algorithm results in compact strata of equal size.
- **verbose**: if TRUE, progress information and intermediate results will be printed to the output device.

Methods

- **object = "SpatialPixels"**: Stratify a raster representation of the study area.
- **object = "SpatialPolygons"**: Stratify a vector representation of the study area.

Note

The stratify method may raise an error when the projection attributes ("CRS") have been set. A solution is to remove these attributes by calling the following function from the sp-package: proj4string(myMap) <- NA_character_, where myMap is the map to be stratified.

References


See Also

spsample for sampling, and estimate for inference.
Example

# Note: the example below requires the 'rgdal'-package
# You may consider the 'maptools'-package as an alternative
if (require(rgdal)) {

    # read a vector representation of the 'Farmsum' field
    shpFarmsum <- readOGR(
        dsn = system.file("maps", package = "spcosa"),
        layer = "farmsum"
    )

    # stratify 'Farmsum' into 50 strata
    # NB: increase argument 'nTry' to get better results
    set.seed(314)
    myStratification <- stratify(shpFarmsum, nStrata = 50, nTry = 1)

    # plot the resulting stratification
    plot(myStratification)
}
Index

*Topic classes
CompactStratification-class, 3
CompactStratificationEqualArea-class, 5
CompactStratificationPriorPoints-class, 6
SamplingPattern-class, 10
SamplingPatternCentroids-class, 10
SamplingPatternPriorPoints-class, 11
SamplingPatternPurposive-class, 12
SamplingPatternRandom-class, 12
SamplingPatternRandomComposite-class, 13
SamplingPatternRandomSamplingUnits-class, 14
SamplingVariance-class, 15
SpatialCumulativeDistributionFunction-class, 15
SpatialMean-class, 16
SpatialVariance-class, 17
StandardError-class, 18
Statistic-class, 19
Stratification-class, 20

*Topic methods
estimate-methods, 6
getArea-methods, 8
getCentroid-methods, 8
getNumberOfStrata-methods, 8
getObjectiveFunctionValue-methods, 8
getRelativeArea-methods, 9
getSampleSize-methods, 9
plot-methods, 9
spsample-methods, 17
stratify-methods, 20

*Topic package
spcosa-package, 2

coerce,CompactStratification,data.frame-method (CompactStratification-class), 3
coerce,CompactStratification,SpatialPixelsDataFrame-method (CompactStratification-class), 3
coerce,SamplingPattern,data.frame-method (SamplingPattern-class), 10
coerce,SamplingPattern,SpatialPoints-method (SamplingPattern-class), 10
coerce,SamplingPatternRandomComposite,data.frame-method (SamplingPatternRandomComposite-class), 13
coerce,SamplingPatternRandomComposite,SpatialPointsDataFrame-method (SamplingPatternRandomComposite-class), 13

estimate,3, 18, 21
estimate,character,CompactStratification,SamplingPatternRandomSamplingUnits,data.frame-method (estimate-methods), 6
estimate,character,CompactStratificationEqualArea,SamplingPatternRandomComposite,data.frame-method (estimate-methods), 6
estimate,SamplingVariance,CompactStratification,SamplingPatternRandomSamplingUnits,data.frame-method (estimate-methods), 6
estimate,SamplingVariance,CompactStratificationEqualArea,SamplingPatternRandomComposite,data.frame-method (estimate-methods), 6
estimate,SpatialCumulativeDistributionFunction,CompactStratification,SamplingPatternRandomSamplingUnits,data.frame-method (estimate-methods), 6
estimate,SpatialMean,CompactStratification,SamplingPatternRandomSamplingUnits,data.frame-method (estimate-methods), 6
estimate,SpatialMean,CompactStratificationEqualArea,SamplingPatternRandomComposite,data.frame-method (estimate-methods), 6

23
INDEX

estimate-methods, 6
getArea, 9
getArea (getArea-methods), 8
getArea (CompactStratification-method (getArea-methods)), 8
getArea-methods, 8
getCentroid (getCentroid-methods), 8
getCentroid (CompactStratification-method (getCentroid-methods)), 8
getCentroid-methods, 8
getNumberOfStrata (getNumberOfStrata-methods), 8
getNumberOfStrata (CompactStratification-method (getNumberOfStrata-methods)), 8
getNumberOfStrata-methods, 8
getObjectiveFunctionValue (getObjectiveFunctionValue-methods), 8
getObjectiveFunctionValue (CompactStratification-method (getObjectiveFunctionValue-methods)), 8
getObjectiveFunctionValue-methods, 8
getRelativeArea, 8
getRelativeArea (getRelativeArea-methods), 9
getRelativeArea (CompactStratification-method (getRelativeArea-methods)), 9
getRelativeArea-methods, 9
getSampleSize (getSampleSize-methods), 9
getSampleSize (SamplingPattern-method (getSampleSize-methods)), 9
getSampleSize (SamplingPatternRandomComposite-method (getSampleSize-methods)), 9
getSampleSize-methods, 9

kmeans, 2

plot, CompactStratification, missing-method (plot-methods), 9
plot, CompactStratification, SamplingPattern-method (plot-methods), 9
plot, CompactStratification, SamplingPatternPriorPoints-method (plot-methods), 9
plot, CompactStratification, SamplingPatternRandomComposite-method (plot-methods), 9
plot, SamplingPattern, missing-method (plot-methods), 9
plot, SamplingPatternPriorPoints, missing-method (plot-methods), 9
plot, SamplingPatternRandomComposite, missing-method (plot-methods), 9
plot-methods, 9

SamplingPattern, 11–14
SamplingPattern-class, 10
SamplingPatternCentroids-class, 10
SamplingPatternClassificationEqualArea-class, 11
SamplingPatternPurpose-class, 11
SamplingPatternRandom-class, 12
SamplingPatternRandomComposite-class, 13
SamplingPatternRandomSamplingUnits, 7
SamplingPatternRandomSamplingUnits-class, 14
SamplingVariance, 4, 5, 7, 13, 14, 19
SamplingVariance-class, 15
show, SamplingPattern-method (SamplingPattern-class), 10
show, Statistic-method (Statistic-class), 19
show, Stratification-method (Stratification-class), 20
SpatialCumulativeDistributionFunction, 4
SpatialCumulativeDistributionFunction-class, 15
SpatialGrid, 21
SpatialMean, 4, 5, 7, 13, 14
SpatialMean-class, 16
SpatialPixels, 4–6, 21
SpatialPixelsDataFrame, 4
SpatialPoints, 4–6, 10–14, 21
SpatialPointsDataFrame, 13
SpatialPolygons, 21
SpatialVariance, 4, 7, 14
SpatialVariance-class, 17
spcosa (spcosa-package), 2
spcosa-package, 2
spsample, 3, 10, 11, 13, 14, 18, 21
spsample (spsample-methods), 17
spsample (CompactStratification, missing, missing-method (spsample-methods)), 17
spsample (CompactStratification, numeric, missing-method (spsample-methods)), 17
spsample (CompactStratificationEqualArea, numeric, character-method (spsample-methods)), 17
spsample (CompactStratificationPriorPoints, missing, missing-method (spsample-methods)), 17
spsample-methods, 17
StandardError, 4, 7, 14
StandardError-class, 18
Statistic, 15–17, 19
Statistic-class, 19
Stratification, 4–6
Stratification-class, 20
stratify, 3, 5, 6, 18
stratify (stratify-methods), 20
stratify, SpatialGrid-method (stratify-methods), 20
stratify, SpatialPixels-method (stratify-methods), 20
stratify, SpatialPolygons-method (stratify-methods), 20
stratify-methods, 20