Package ‘spcosa’

April 11, 2023

Type Package

Title Spatial Coverage Sampling and Random Sampling from Compact Geographical Strata

Version 0.4-2

Date 2023-04-11

Description Spatial coverage sampling and random sampling from compact geographical strata created by k-means. See Walvoort et al. (2010) <doi:10.1016/j.cageo.2010.04.005> for details.

Depends R (>= 4.1.0), rJava (>= 1.0-0), methods, utils

Imports sp (>= 1.4-6), ggplot2 (>= 3.3.0)

Suggests grid, RUnit (>= 0.4.32), sf (>= 1.0-12), knitr (>= 1.36), markdown (>= 2.11)

SystemRequirements Java (>= 6)

License GPL (>= 3)

URL https://git.wur.nl/Walvo001/spcosa

VignetteBuilder knitr

Collate class_Stratification.R class_CompactStratification.R
class_CompactStratificationEqualArea.R
class_CompactStratificationPriorPoints.R
class_SamplingPattern.R class_SamplingPatternPurposive.R
class_SamplingPatternCentroids.R
class_SamplingPatternPriorPoints.R
class_SamplingPatternRandom.R
class_SamplingPatternRandomSamplingUnits.R
class_SamplingPatternRandomComposite.R class_Statistic.R
class_SpatialMean.R class_SamplingVariance.R
class_StandardError.R class_SpatialVariance.R
class_SpatialCumulativeDistributionFunction.R
generic_estimate.R generic_getArea.R generic_getRelativeArea.R
generic_getAttributes.R generic_getCellSize.R
generic_getCentroid.R generic_getSampleSize.R
generic_getNumberOfCells.R generic_getNumberOfStrata.R
R topics documented:

generic_getObjectiveFunctionValue.R  generic_stratify.R
generic_plot.R  methods_estimate.R
method_getArea-CompactStratification.R
method_getRelativeArea-CompactStratification.R
method_getCellSize-CompactStratification.R
method_getCellSize-SpatialPixels.R
method_getCentroid-CompactStratification.R
method_getNumberOfCells-CompactStratification.R
method_getNumberOfStrata-CompactStratification.R
method_getObjectiveFunctionValue-CompactStratification.R
method_getSampleSize-SamplingPattern.R
method_getSampleSize-SamplingPatternRandomComposite.R
method_initialize-CompactStratification.R
method_initialize-CompactStratificationPriorPoints.R
method_stratify-SpatialGrid.R  method_stratify-SpatialPixels.R
method_stratify-SpatialPolygons.R  methods_plot.R
methods_spsample.R  methods_setAs.R  method_show-Stratification.R
onAttach.R

NeedsCompilation: no

Author: Dennis Walvoort [aut, cre, cph]
        (<https://orcid.org/0000-0002-3967-5430>),
        Dick Brus [aut, cph] (<https://orcid.org/0000-0003-2194-4783>),
        Jaap de Gruijter [aut, cph]

Maintainer: Dennis Walvoort <dennis.Walvoort@wur.nl>

Repository: CRAN

Date/Publication: 2023-04-11 10:30:02 UTC

R topics documented:

spcosa-package ...................................................... 3
CompactStratification-class ......................................... 4
CompactStratificationEqualArea-class .............................. 6
CompactStratificationPriorPoints-class ............................. 7
estimate-methods ..................................................... 8
getArea-methods ..................................................... 9
getCentroid-methods ................................................ 9
getNumberOfStrata-methods .......................................... 10
getObjectiveFunctionValue-methods ................................. 10
getRelativeArea-methods ........................................... 10
getSampleSize-methods ............................................. 11
plot-methods ......................................................... 11
SamplingPattern-class .............................................. 12
SamplingPatternCentroids-class .................................... 12
SamplingPatternPriorPoints-class .................................. 13
SamplingPatternPurposive-class .................................... 14
spcosa-package

Spatial Coverage Sampling and Random Sampling from Compact Geographical Strata

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 is installed. Free Java downloads are available at https://www.java.com.

Author(s)

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


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.

estimate signature(statistic = "StandardError", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates the standard error of the spatial mean. See "StandardError" for more details.

getArea signature(object = "CompactStratification"): returns the area of each stratum.

gCentroid signature(object = "CompactStratification"): returns the center of gravity of each stratum.

getNumberOfStrata signature(object = "CompactStratification"): returns the number of strata.

gObjectiveFunctionValue 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.

spsample signature(x = "CompactStratification", n = "missing", type = "missing"): returns the centers of gravity of each stratum.

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

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

CompactStratificationEqualArea-class

Class "CompactStratificationEqualArea"

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 = "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
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
Description

Methods for estimating statistics given a spatial sample.

Methods

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

- `statistic = "character", stratification = "CompactStratificationEqualArea", samplingPattern = "SamplingPatternRandomComposite", data = "data.frame"`
  estimates one of the following statistics, depending on the value of argument `statistic`:
  - spatial mean
  - sampling variance
  - standard error
  See "SamplingVariance" for more details.

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

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

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

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

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

Examples

```r
# Note: the example below requires the 'sf'-package.
if (require(sf)) {
  # read vector representation of the "Mijdrecht" area
  shp <- as(st_read(
    dsn = system.file("maps", package = "spcosa"),
    layer = "mijdrecht"), "Spatial")

  # 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
getSampleSize-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.

See Also

ggplot2-package
**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`.

---
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

---

SamplingPatternPriorPoints-class

Class "SamplingPatternPriorPoints"

Description

A class for storing a sampling pattern consisting of existing points and new points. The new points are the centers of gravity of their stratum.

Objects from the Class

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

Slots

.isPriorPoint: Object of class "logical", which is TRUE if the location is a prior point, and FALSE if it is not.

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

Extends


Methods

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

plot signature(x = "SamplingPatternPriorPoints", y = "missing"): plots sampling pattern x.
Author(s)
Dennis J. J. Walvoort <dennis.walvoort@wur.nl>, D.J. Brus, J.J. de Gruijter

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

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
SamplingPatternRandomComposite-class

Extends

Class "SamplingPattern", directly.

Methods

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

Author(s)

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

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.

t: 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.
**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.
SamplingVariance-class

estimate signature(statistic = "SpatialVariance", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates the spatial variance. See "SpatialVariance" for more details.

estimate signature(statistic = "StandardError", stratification = "CompactStratification", samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates the standard error of the spatial mean. See "StandardError" 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

---

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
SpatialCumulativeDistributionFunction-class

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

Description

The spatial mean is estimated by means of Equation 7.13 in de Gruijter et al., (2006).

Objects from the Class

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

Slots

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

Extends

Class "Statistic", directly.

Methods

**estimate** signature(statistic = "SpatialMean", stratification = "CompactStratification",
samplingPattern = "SamplingPatternRandomSamplingUnits", data = "data.frame"): estimates the spatial mean, given a stratification, a sampling pattern and data.

**estimate** signature(statistic = "SpatialMean", stratification = "CompactStratificationEqualArea",
samplingPattern = "SamplingPatternRandomComposite", data = "data.frame"): estimates the spatial mean, 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  Class "SpatialVariance"

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

spsample-methods  Spatial Sampling of Compact Strata

Description
Methods for sampling in compact strata.
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 'sf'-package.
if (require(sf)) {

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

  # 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 \(k\)-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**

```r
## 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

When the projection attribute of a map is set to EPSG:4326 (lat/lon), great circle distances will be used for stratification. Otherwise, Euclidean distances will be used.

References


See Also

spsample for sampling, and estimate for inference.
stratify-methods

Examples

# Note: the example below requires the 'sf'-package
if (require(sf)) {

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

    # 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

## classes
- CompactStratification-class, 4
- CompactStratificationEqualArea-class, 6
- CompactStratificationPriorPoints-class, 7
- SamplingPattern-class, 12
- SamplingPatternCentroids-class, 12
- SamplingPatternPriorPoints-class, 13
- SamplingPatternPurposive-class, 14
- SamplingPatternRandom-class, 14
- SamplingPatternRandomComposite-class, 15
- SamplingPatternRandomSamplingUnits-class, 16
- SamplingVariance-class, 17
- SpatialCumulativeDistributionFunction-class, 18
- SpatialMean-class, 19
- SpatialVariance-class, 20
- StandardError-class, 21
- Statistic-class, 22
- Stratification-class, 23

## methods
- estimate-methods, 8
- getArea-methods, 9
- getCentroid-methods, 9
- getNumberOfStrata-methods, 10
- getObjectiveFunctionValue-methods, 10
- getRelativeArea-methods, 10
- getSampleSize-methods, 11
- plot-methods, 11
- spsample-methods, 20
- stratify-methods, 23

## package
- spcosa-package, 3

coerce,CompactStratification,SpatialPixels-method (CompactStratification-class), 4
coeerce,CompactStratification,SpatialPixelsDataFrame-method (CompactStratification-class), 4
coeerce,CompactStratification,SamplingPattern-class (SamplingPattern-class), 12
coeerce,CompactStratification,SamplingPatternCentroids-class (SamplingPattern-class), 12
coeerce,CompactStratification,SamplingPatternPriorPoints-class (SamplingPattern-class), 12
coeerce,CompactStratification,SamplingPatternPurposive-class (SamplingPattern-class), 12
coeerce,CompactStratification,SamplingPatternRandom-class (SamplingPattern-class), 12
coeerce,CompactStratification,SamplingPatternRandomComposite-class (SamplingPattern-class), 12
coeerce,CompactStratification,SamplingPatternRandomSamplingUnits-class (SamplingPattern-class), 12

estimate, data.frame, 4, 21, 24
estimate(estimate-methods), 8
estimate,character,CompactStratification,SamplingPatternRandomSamplingUnits-class (estimate-methods), 8
estimate,character,CompactStratification,SamplingPatternRandomComposite-class (estimate-methods), 8
estimate,SamplingVariance,CompactStratification,SamplingPatternRandomSamplingUnits-class (estimate-methods), 8
estimate,SamplingVariance,CompactStratification,SamplingPatternRandomComposite-class (estimate-methods), 8
estimate,SpatialCumulativeDistributionFunction,CompactStratification,SamplingPatternRandomSamplingUnits-class (estimate-methods), 8
estimate,SpatialCumulativeDistributionFunction,CompactStratification,SamplingPatternRandomComposite-class (estimate-methods), 8
estimate, SpatialMean, CompactStratification, SamplingPatternRandomSamplingUnits

getArea

getRelativeArea

getSampleSize

kmeans

plot, CompactStratification, missing

plot, SamplingPatternRandomComposite, missing

SamplingPattern, 13–16

SamplingVariance, 5, 6, 8, 15, 16, 22

SpatialMean, 5, 6, 8, 15, 16

SpatialPoints, 4–7, 24

SpatialPolygons, 24

SpatialVariance, 5, 8, 17
SpatialVariance-class, 20
spcosa (spcosa-package), 3
spcosa-package, 3
spsample, 4, 12, 13, 15, 16, 21, 24
spsample (spsample-methods), 20
spsample, CompactStratification, missing, missing-method
 (spsample-methods), 20
spsample, CompactStratification, numeric, missing-method
 (spsample-methods), 20
spsample, CompactStratificationEqualArea, numeric, character-method
 (spsample-methods), 20
spsample, CompactStratificationPriorPoints, missing, missing-method
 (spsample-methods), 20
spsample-methods, 20
StandardError, 5, 8, 17
StandardError-class, 21
Statistic, 17–20, 22
Statistic-class, 22
Stratification, 4, 6, 7
Stratification-class, 23
stratify, 4, 6, 7, 21
stratify (stratify-methods), 23
stratify, SpatialGrid-method
 (stratify-methods), 23
stratify, SpatialPixels-method
 (stratify-methods), 23
stratify, SpatialPolygons-method
 (stratify-methods), 23
stratify-methods, 23