Package 'spcosa'

December 5, 2012

Type Package

Title Spatial Coverage Sampling and Random Sampling from Compact Geographical Strata

Version 0.3-2

Date 2012-12-05

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)

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 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 method_show-SamplingPattern.R method_show-Statistic.R onLoad.R

R topics documented:

`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 (>= 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\RtmpE7TCVd\Rinst170c1aba6522\spcosa\FAQ

Author(s)

D.J.J. Walvoort, D.J. Brus, J.J. de Gruijter,

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

References

Brus, D. J., Spatjens, L. E. E. M., and de Gruijter, J. J. (1999). A sampling scheme for estimating the mean extractable phosphorus concentration of fields for environmental regulation. Geoderma 89:129-148

de Gruijter, J. J., Brus, D. J., Bierkens, M. F. P., and Knotters, M. (2006). *Sampling for Natural Resource Monitoring* Berlin: Springer-Verlag.

Walvoort, D., Brus, D. and de Gruijter, J. (2009). Spatial Coverage Sampling on Various Spatial Scales. Pedometron 26:20-22

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

$$k$$

-means. Computers & Geosciences 36: 1261-1267 (http://dx.doi.org/10.1016/j.cageo.2010.04.005)

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", samp` estimates the sampling variance. See "SamplingVariance" for more details.

estimate `signature(statistic = "SpatialCumulativeDistributionFunction", stratification = "Compact` estimates the spatial cumulative distribution function (SCDF). See "SpatialCumulativeDistributionFunction" for more details.

estimate `signature(statistic = "SpatialMean", stratification = "CompactStratification", samplingF` estimates the spatial mean. See "SpatialMean" for more details.

estimate `signature(statistic = "SpatialVariance", stratification = "CompactStratification", sampl` estimates the spatial variance. See "SpatialVariance" for more details.

estimate `signature(statistic = "StandardError", stratification = "CompactStratification", samplir` estimates the standard error of the spatial mean. See "StandardError" for more details.

estimate `signature(statistic = "character", stratification = "CompactStratification", samplingPat` estimates `statistic`, one of `spatial mean`, `spatial variance`, `SCDF`, `sampling variance`, or `standard error`.

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`.

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.

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, centro`
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 `"CompactStratification"`, directly. Class `"Stratification"`, by class "CompactStratification", distance 2.

Methods

estimate `signature(statistic = "SamplingVariance", stratification = "CompactStratificationEqualAr`
estimates the sampling variance. See `"SamplingVariance"` for more details.

estimate `signature(statistic = "SpatialMean", stratification = "CompactStratificationEqualArea",`
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, cent`
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

Class `"CompactStratification"`, directly. Class `"Stratification"`, by class "CompactStratification", distance 2.

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 = "SamplingPatternRandomS
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 = "SamplingPatter
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 = "SamplingPatternR
estimates the sampling variance. See "SamplingVariance" for more details.

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

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

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

statistic = "SpatialVariance", stratification = "CompactStratification", samplingPattern = "SamplingPatternRan
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

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

Class "SamplingPatternPurposive", directly. Class "SamplingPattern", by class "SamplingPatternPurposive", distance 2.

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 is the location is a prior point, and FALSE if it is not.

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

Extends

Class "SamplingPatternPurposive", directly. Class "SamplingPattern", by class "SamplingPatternPurposive", distance 2.

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

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.

Author(s)

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

---

`SamplingPatternRandomComposite-class`
Class "SamplingPatternRandomComposite"

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

Class `"SamplingPatternRandom"`, directly. Class `"SamplingPattern"`, by class "SamplingPatternRandom", distance 2.

Methods

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

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

estimate `signature(statistic = "SamplingVariance", stratification = "CompactStratificationEqualAr`
estimates the sampling variance. See `"SamplingVariance"` for more details.

estimate `signature(statistic = "SpatialMean", stratification = "CompactStratificationEqualArea",`
estimates the spatial mean. See `"SpatialMean"` for more details.

getSampleSize `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

Class `"SamplingPatternRandom"`, directly. Class `"SamplingPattern"`, by class "SamplingPatternRandom", distance 2.

Methods

estimate `signature(statistic = "SamplingVariance", stratification = "CompactStratification", samp`
estimates the sampling variance. See `"SamplingVariance"` for more details.

estimate `signature(statistic = "SpatialCumulativeDistributionFunction", stratification = "Compact`
estimates the spatial cumulative distribution function (SCDF). See `"SamplingPatternRandomSamplingUnits"` for more details.

estimate `signature(statistic = "SpatialMean", stratification = "CompactStratification", samplingF`
estimates the spatial mean. See `"SpatialMean"` for more details.

estimate `signature(statistic = "SpatialVariance", stratification = "CompactStratification", sampl`
estimates the spatial variance. See `"SpatialVariance"` for more details.

estimate `signature(statistic = "StandardError", stratification = "CompactStratification", samplir`
estimates the standard error of the spatial mean. See `"StandardError"` for more details.

estimate `signature(statistic = "character", stratification = "CompactStratification", samplingPat`
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

The sampling variance is estimated by means of Equation 7.14 in *de Gruijter et al., (2006).*

Objects from the Class

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

Slots

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

Extends

Class `"Statistic"`, directly.

Methods

estimate `signature(statistic = "SamplingVariance", stratification = "CompactStratification", samp`
estimates the sampling variance, given a stratification, a sampling pattern and data.

estimate `signature(statistic = "SamplingVariance", stratification = "CompactStratificationEqualAr`
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

de Gruijter, J. J., Brus, D. J., Bierkens, M. F. P., and Knotters, M. (2006) *Sampling for Natural Resource Monitoring* Berlin: Springer-Verlag.

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 op the statistic.

Extends

Class `"Statistic"`, directly.

Methods

estimate `signature(statistic = "SpatialCumulativeDistributionFunction", stratification = "Compact` 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

de Gruijter, J. J., Brus, D. J., Bierkens, M. F. P., and Knotters, M. (2006) *Sampling for Natural Resource Monitoring* Berlin: Springer-Verlag.

SpatialMean-class *Class "SpatialMean"*

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", samplingP` estimates the spatial mean, given a stratification, a sampling pattern and data.

estimate `signature(statistic = "SpatialMean", stratification = "CompactStratificationEqualArea",` 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

de Gruijter, J. J., Brus, D. J., Bierkens, M. F. P., and Knotters, M. (2006) *Sampling for Natural Resource Monitoring* Berlin: Springer-Verlag.

`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 op the statistic.

Extends

Class `"Statistic"`, directly.

Methods

estimate `signature(statistic = "SpatialVariance", stratification = "CompactStratification", sampl`
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

de Gruijter, J. J., Brus, D. J., Bierkens, M. F. P., and Knotters, M. (2006) *Sampling for Natural Resource Monitoring* Berlin: Springer-Verlag.

`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

```
# 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

The standard error is estimated by means of the square root of Equation 7.14 in *de Gruijter et al., (2006).*

Objects from the Class

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

Slots

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

Extends

Class `"SamplingVariance"`, directly. Class `"Statistic"`, by class "SamplingVariance", distance 2.

Methods

estimate `signature(statistic = "StandardError", stratification = "CompactStratification", samplir` 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

de Gruijter, J. J., Brus, D. J., Bierkens, M. F. P., and Knotters, M. (2006) *Sampling for Natural Resource Monitoring* Berlin: Springer-Verlag.

`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 op 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

```
## 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

Brus, D. J., Spatjens, L. E. E. M., and de Gruijter, J. J. (1999). A sampling scheme for estimating the mean extractable phosphorus concentration of fields for environmental regulation. Geoderma 89:129-148

de Gruijter, J. J., Brus, D. J., Bierkens, M. F. P., and Knotters, M. (2006) *Sampling for Natural Resource Monitoring* Berlin: Springer-Verlag.

Walvoort, D., Brus, D. and de Gruijter, J. (2009). Spatial Coverage Sampling on Various Spatial Scales. Pedometron 26:20-22

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

$$k$$

-means. Computers & Geosciences 36: 1261-1267 (http://dx.doi.org/10.1016/j.cageo.2010.04.005)

See Also

spsample for sampling, and estimate for inference.

Examples

```
# 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