Package ‘NCSampling’

October 12, 2022

Type Package

Title Nearest Centroid (NC) Sampling

Version 1.0

Date 2017-06-26

Author GJ Melville

Maintainer Gavin Melville <gavin.melville@dpi.nsw.gov.au>

Imports yaImpute, lattice, randomForest

Description Provides functionality for performing Nearest Centroid (NC) Sampling. The NC sampling procedure was developed for forestry applications and selects plots for ground measurement so as to maximize the efficiency of imputation estimates. It uses multiple auxiliary variables and multivariate clustering to search for an optimal sample. Further details are given in Melville G. & Stone C. (2016) <doi:10.1080/00049158.2016.1218265>.

NeedsCompilation no

Repository CRAN

Date/Publication 2017-06-27 06:14:25 UTC

R topics documented:

`NCSampling-package` *Nearest Centroid (NC) Sampling*

Description

Suite of functions to perform NC sampling. Used by forestry practitioners to select reference plots for imputation using remotely sensed data, for example aerial laser scanning (ALS) data.

Details

Package: NCSampling
Type: Package
Version: 1.0
Date: 2017-06-26
License: GPL-2

Depending on the application, the functions are usually called in the following order:-
Check.pop - check population file for errors
Alloc - allocate sample numbers to strata
Existing - determine the virtual plots, in the target set, which are neighbours to pre-existing plots
Alloc - re-allocate sample numbers to strata, taking into account pre-existing plots and their neighbours
NC.sample - select reference plots from the candidate set, using the internal functions Centroids and NC.select.
Spatial.plot - display the virtual plots, including the NC sample plots, as an x-y graph.
DesVar - calculate approximate design variances for each stratum and for the whole population.

Author(s)

G Melville Maintainer: <gavin.melville@dpi.nsw.gov.au>

References

G. Melville & C. Stone. (2016) Optimising nearest neighbour information - a simple, efficient sampling strategy for forestry plot imputation using remotely sensed data. Australian Forestry, 79:3, 217:228, DOI: 10.1080/00049158.2016.1218265.

`Addz` *Addz*

Description

Add variable/s to the population file which are good predictors of the variables/s of interest

Usage

```
Addz(popfile, training, yvars, xvars, pool)
```

Arguments

`popfile`	dataframe containing population data - as a minimum there must be columns named 'PID' (plot identifier), 'Strata' and 'plot_type'.
`training`	dataframe containing training data. Must contain auxiliary variables and variable/s of interest.
`yvars`	vector containing the name of each variable of interest (dependent variable).
`xvars`	vector containing the names of the auxiliary variables.
`pool`	logical value - should the training data be pooled across strata prior to fitting the regression model?

Details

The predictor variable for the each variable of interest (dependent variable) is obtained by performing random forest regression on the training data using the designated auxiliary variables. The training data can be pooled across strata (pool=T), or fitted separately within each strata (the default). Not normally called directly.

Value

A list with components:-

`popfile`	population file - data frame, as above, with predictor variable/s added to the file
`r.sqared`	dataframe containing the R-squared values obtained from the random forest regression/s

Author(s)

G. Melville

References

Random forest regression is performed using the randomForest package.

See Also

`DesVar`, `randomForest`.

Examples

```
## Addz(popfile, training, yvars, xvars)
```

`Alloc` *Allocation*

Description

Allocate sample among several strata, using proportional allocation. Inputs population file and total sample size. Outputs sample sizes for each stratum

Usage

```
Alloc(popfile, ntotal)
```

Arguments

`popfile`	dataframe containing population data - as a minimum there must be columns named 'PID' (plot identifier), 'Strata' and 'plot_type'.
`ntotal`	total sample size - required number of reference plots for all strata combined.

Details

Performs a proportional allocation, by calculating the required sample size for each stratum (i) using the formula n_i = n * N_i / N, where n is the sample size (number of reference plots) and N is the number of target plots.

Value

A vector of sample sizes, one for each stratum in the population file.

Author(s)

G. Melville

See Also

`Existing` and `NC.sample`.

Examples

```
popfile<-data.frame(PID=1:20, Strata=rep(c('A', 'B'),c(12,8)),
  plot_type=rep('B',20))
tot.samp<-6
Alloc(popfile, tot.samp)
```

`Centroids` *Calculate centroids*

Description

Separates a single stratum of the population file into n clusters and finds the centroid of each cluster, where n is the sample size. Not intended to be called directly.

Usage

```
Centroids(popfile, nrefs, desvars, ctype, imax, nst)
```

Arguments

`popfile`	population file - dataframe containing information relating to all plots in the stratum.
`nrefs`	scalar defining the number of reference plots - required sample size for the stratum.
`desvars`	character vector containing the names of the design variables.
`ctype`	clustering type - either k-means ('km') or Ward's D2 ('WD').
`imax`	maximum number of iterations when calling the k-means clustering procedure.
`nst`	number of random initial centroid sets when calling the k-means clustering procedure.

Details

The virtual plots are partitioned so as to minimise the sums of squares of distances from plots to cluster centroids. This is done by using a multivariate clustering procedure such as k-means clustering (Hartigan & Wong, 1979) or Ward's D2 clustering (Murtagh & Legendre, 2013), using standardized design variables and a Euclidean distance metric.

Value

`centroids`	dataframe containing centroids.
`cmns`	dataframe containing centroid means.

Author(s)

G Melville

References

Hartigan & Wong (1979) Algorithm AS 136: a K-means clustering algorithm. Applied Statistics 28, 100-108, DOI:10.2307/2346830.

Murtagh, M & Legendre, P. (2014) Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? Journal of Classification, 31, 274-295, DOI: 10.1007/s00357-014-9161-z.

See Also

`Existing`, `NC.sample` and `kmeans`.

Examples

```
## Centroids(popfile, nrefs, desvars, ctype='km', imax=200, nst=20)
```

`Check.pop` *Check population file*

Description

Carries out a range of checks on the population file to detect the most commonly encountered errors. Provides a barchart showing the population structure.

Usage

```
Check.pop(popfile, desvars)
```

Arguments

`popfile`	dataframe containing information for all plots in the population.
`desvars`	vector containing the names of the design variables.

Value

Reports on any errors found and produces a barchart.

Author(s)

G. Melville

See Also

`NC.sample`.

Examples

```
## Check.pop(popfile, desvars)
```

DesVar *Design variances for NC sample.*

Description

For each stratum ,and for the population as a whole, approximate design variances are calculated.

Usage

```
DesVar(popfile, nrefs, desvars, yvars, kvalue, B=1000, zvars=NULL,

training=NULL, xvars=NULL, pool=F)
```

Arguments

popfile	dataframe containing information on all plots in the population.
nrefs	vector containing the sample size of each stratum.
desvars	vector containing the names of the design variables.
yvars	character vector containing the name of each variable of interest (dependent variable) for which design variances are required.
kvalue	scalar specifying the value of k for the k-nn imputation.
B	number of re-samples used to calculate the design variances.
zvars	character vector containing the name/s of the predictor variables.
training	dataframe containing the data needed to determine the predictor variable. Must contain the necessary yvars and xvars. If missing, predictor variables are supplied by the user (zvars)
xvars	character vector containing the name/s of the predictor variables.
pool	logical value - should strata be pooled prior to fitting regression model?

Details

Approximate design variances are calculated using a re-sampling procedure in conjunction with a predictor variable. The predictor variable can be user-supplied or determined by the program using random forest regression based on a set of training data. The regression model can be fitted separately for each strata (pool=F), the default, or based on pooled training data with stratum included in the regression model as a factor.

Value

A dataframe containing the design variances for each stratum and for the whole population.

Author(s)

G. Melville

See Also

NC.sample.

Examples

```
## DesVar(popfile, nrefs, desvars, yvars, B=1000, zvars=NULL,
##   training=NULL, xvars=NULL, pool=F)
```

DVar *Design variances for single stratum.*

Description

For a single stratum approximate design variances are calculated. Not intended to be called directly.

Usage

```
DVar(popfile, nrefs, yvars, desvars, kvalue, B=1000)
```

Arguments

popfile	dataframe containing information on stratum of interest.
nrefs	scalar containing the sample size of the stratum.
yvars	character vector containing the name of each variable of interest (dependent variable) for which design variances are required.
desvars	character vector containing the names of the design variables.
kvalue	scalar specifying the value of k for the k-nn imputation.
B	number of re-samples used to calculate the design variances.

Value

A dataframe containing the design variances for the stratum of interest. Data used to calculate these are also returned.

Author(s)

G. Melville

See Also

NC.sample, DesVar.

Examples

```
## DesVar(popfile, nrefs, yvars, kvalue, desvars, B=1000)
```

`Existing` *Pre-existing plot neighbours*

Description

Determines the plots which are close, in the auxiliary space, to the pre-existing plots.

Usage

```
Existing(popfile, nrefs, desvars, draw.plot)
```

Arguments

`popfile`	dataframe containg information on all plots in the population file.
`nrefs`	vector containing the number of reference plots in each stratum.
`desvars`	vector containing the names of the design variables.
`draw.plot`	logical variable - should a bar graph be drawn to show the number of neighbours for each pre-existing plot?

Value

A list with components:-

`Nx`	vector containing the number of neighbours to existing plots in each stratum.
`Ng`	vector containing the number of target plots in each stratum.
`popfile`	dataframe containing the original population file with neighbours to pre-existing plots separately identified.

Author(s)

G Melville.

See Also

`NC.sample`.

Examples

```
## Existing(popfile, nrefs, desvars, draw.plot=T)
```

NC.sample *Nearest Centroid (NC) Sample*

Description

Selects NC sample in multiple strata.

Usage

```
NC.sample(popfile, nrefs, desvars, ctype, imax, nst)
```

Arguments

`popfile`	dataframe containing information on all plots in the population.
`nrefs`	vector containing the sample size of each stratum.
`desvars`	vector containing the names of the design variables.
`ctype`	clustering type - either k-means ('km') or Wards D ('WD').
`imax`	maximum number of iterations for the k-means procedure.
`nst`	number of initial random sets of cluster means for the k-means procedure.

Details

In each stratum the population of virtual plots is segregated into n clusters where n is the stratum sample size (number of reference plots). The virtual plots are partitioned so as to minimise the sums of squares of distances from plots to cluster centroids. This is achieved by using a multivariate clustering procedure such as k-means clustering (Hartigan & Wong, 1979) or Ward's D clustering (Murtagh & Legendre, 2013), using standardized design variables and a Euclidean distance metric. Following determination of the cluster centroids, the virtual plot, in the candidate set, closest to each centroid is selected as a reference plot.

Value

A list with components:-

`popfile`	population file - dataframe, as above, with reference plots designated as 'R'
`cmns`	centroid means

Author(s)

G. Melville

References

G. Melville & C. Stone. (2016) Optimising nearest neighbour information - a simple, efficient sampling strategy for forestry plot imputation using remotely sensed data. Australian Forestry, 79:3, 217:228, DOI: 10.1080/00049158.2016.1218265.

Hartigan & Wong (1979) Algorithm AS 136: a K-means clustering algorithm. Applied Statistics 28, 100-108, DOI:10.2307/2346830.

Murtagh, M & Legendre, P. (2013) Ward's hierarchical agglomerative clustering method: Which algorithms implement Ward's criterion? Journal of Classification.

See Also

See also `NC.sample`.

Examples

```
## NC.sample(popfile, nrefs, desvars, ctype='km', imax=200, nst=20)
```

NC.select — *Nearest Centroid (NC) Plot Selection*

Description

Select the reference plots closest, in the auxiliary space, to the target plot centroids. Not intended to be called directly.

Usage

```
NC.select(popfile, nrefs, desvars, centroids)
```

Arguments

`popfile`	dataframe containing information on all plots in the stratum.
`nrefs`	vector containing the number of reference plots in the stratum.
`desvars`	vector containing the names of the design variables.
`centroids`	dataframe containing the centroids for the stratum.

Value

A list with components:-

`refs`	dataframe containing reference plots
`exist`	dataframe containing pre-existing plots
`targs`	dataframe containing target plots

Author(s)

G. Melville

See Also

NC.sample.

Examples

```
## NC.select(popfile, nrefs, desvars, centroids)
```

---

nundle.sf *Nundle State Forest LiDAR data*

Description

LiDAR data from two strata acquired by over-flying the Nundle State Forest (SF), NSW, Australia in 2011

Usage

```
data(nundle.sf)
```

Format

A data frame with 2068 observations on the following 12 variables.

`PID` numeric vector containing unique plot IDs

`height` numeric vector containing LiDAR heights

`meanht` numeric vector containing LiDAR mean heights

`mam` a numeric vector containing mean above mean heights

`mdh` a numeric vector containing LiDAR mean dominant heights

`pstk` a numeric vector containing LiDAR stocking rate

`cc` a numeric vector containing LiDAR canopy cover

`OV` a numeric vector containing LiDAR occupied volume

`var` a numeric vector containing LiDAR height variances

`Strata` a factor with levels `O`, `Y`

`x` a numeric vector containing x-coordinates

`y` a numeric vector containing y-coordinates

Details

The LiDAR variables were calculated as outlined in Turner et al. (2011).

Source

Forestry Corporation of NSW

References

Melville G, Stone C, Turner R (2015). Application of LiDAR data to maximize the efficiency of inventory plots in softwood plantations. New Zealand Journal of Forestry Science, 45:9,1-16. doi:10.1186/s40490-015-0038-7.

Stone C, Penman T, Turner R (2011). Determining an optimal model for processing lidar data at the plot level: results for a Pinus radiata plantation in New SouthWales, Australia. New Zealand Journal of Forestry Science, 41, 191-205.

Turner R, Kathuria A, Stone C (2011). Building a case for lidar-derived structure stratification for Australian softwood plantations. In Proceedings of the SilviLaser 2011 conference, Hobart, Tasmania, Australia.

Examples

```
data(nundle.sf)
```

`R.sample1` *Random sample.*

Description

Selects random sample in a single stratum.

Usage

```
R.sample1(popfile, nrefs)
```

Arguments

`popfile`	dataframe containing information on all plots in the stratum.
`nrefs`	vector containing the required sample size of the stratum.

Details

A random sample of virtual plots is selected from the candidate set in the stratum of interest.

Value

A list with components:-

`popfile`	population file - dataframe, as above, with plot type of reference plots set to 'R'

Author(s)

G. Melville

See Also

NC.sample.

Examples

```
## R.sample1(popfile, nrefs)
```

Spatial.plot *Spatial Plot*

Description

Spatial (x-y) graph of candidate plots, target plots, pre-existing plots, reference plots and neighbours to pre-existing plots.

Usage

```
Spatial.plot(popfile, sampfile)
```

Arguments

popfile	dataframe containing information on all plots in the population prior to the sample.
sampfile	dataframe containing information on all plots in the population after the sample.

Value

Draws an x-y plot showing the location of different plots in each stratum.

Author(s)

G. Melville

See Also

See also NC.sample.

Examples

```
## Spatial.plot(popfile, sampfile)
```

training *Nundle State Forest LiDAR data*

Description

Contains LiDAR data for 200 plots from two strata acquired by over-flying the Nundle State Forest (SF), NSW, Australia in 2011

Usage

```
data(training)
```

Format

A data frame with 200 observations on the following 10 variables.

`OV` a numeric vector containing LiDAR occupied volume

`height` numeric vector containing LiDAR heights

`cc` a numeric vector containing LiDAR canopy cover

`pstk` a numeric vector containing LiDAR stocking rate

`var` a numeric vector containing LiDAR height variances

`x` a numeric vector containing x-coordinates

`y` a numeric vector containing y-coordinates

`Strata` a factor with levels `O` `Y`

`PID` numeric vector containing unique plot IDs

`plot_type` a factor with levels `B` `C` `T`

Details

The LiDAR variables were calculated as outlined in Turner et al. (2011).

Source

Forestry Corporation of NSW

References

Melville G, Stone C, Turner R (2015). Application of LiDAR data to maximize the efficiency of inventory plots in softwood plantations. New Zealand Journal of Forestry Science, 45:9,1-16. doi:10.1186/s40490-015-0038-7.

Stone C, Penman T, Turner R (2011). Determining an optimal model for processing lidar data at the plot level: results for a Pinus radiata plantation in New SouthWales, Australia. New Zealand Journal of Forestry Science, 41, 191-205.

Turner R, Kathuria A, Stone C (2011). Building a case for lidar-derived structure stratification for Australian softwood plantations. In Proceedings of the SilviLaser 2011 conference, Hobart, Tasmania, Australia.

Examples

```
data(training)
```

Index