

Package ‘IndRSA’

July 3, 2018

Version 0.0.0.9000

Title What the Package Does (One Line, Title Case)

Description What the package does (one paragraph).

License What license is it under?

Encoding UTF-8

LazyData true

ByteCompile true

RoxygenNote 6.0.1.9000

Suggests testthat

Import randomForest

R topics documented:

aictab_ind	2
bad_fit	2
bad_fit1	3
cmodel	4
eval_ratio	4
goats	5
ind_coef	5
ind_se	6
kfoldRSF	6
kfold_ind	7
pop_avg	8
resample_rsf	9
rf	9
rm_bad_fit	10
rm_bad_fit1	10
rm_conv_fit	11
rm_conv_fit1	11
rsf_ind	12
rsf_mod	12
Index	14

aictab_ind	<i>Perform model selection over all individuals</i>
------------	---

Description

Perform AIC model selection over all individuals by adding up likelihood of individual model (based code partly taken from package AICcmodavg)

Usage

```
aictab_ind(mod_ls, cutoff = -1)
```

Arguments

mod_ls	A list of list of model generated by rsf_ind
cutoff	A cutoff value to exclude individuals with bad fit, default = -1 indicating model that did not converge will be excluded. Values > 0 will exclude based on coefficient values

Value

A AIC model selection table

Examples

```
data(goats)
ls1<-list()
ls1[[1]]<-as.formula(STATUS~ELEVATION+SLOPE+ET+ASPECT+HLI+TASP)
ls1[[2]]<-as.formula(STATUS~ET+ASPECT+HLI+TASP)
out<-rsf_ind(goats$ID, data=goats, form_ls=ls1)
aictab_ind(out)
```

bad_fit	<i>Identify potential individual with bad fits based on coefficients and model convergence</i>
---------	--

Description

Identify potential individual with bad fits based on coefficients and model convergence

Usage

```
bad_fit(mod_ls, cutoff = 1000)
```

Arguments

mod_ls	A list of list of model generated by rsf_ind
cutoff	Value a coefficient may take to indicate bad fit (default=1000)

Value

A list with first element giving individuals with bad fits based on coefficients and second element containing individuals with bad fit based on convergence

Examples

```
data(goats)
ls1<-list()
ls1[[1]]<-as.formula(STATUS~ELEVATION+SLOPE+ET+ASPECT+HLI+TASP)
ls1[[2]]<-as.formula(STATUS~ET+ASPECT+HLI+TASP)
out<-rsf_ind(goats$ID, data=goats, form_ls=ls1)
bad_fit(out, cutoff=20) #None
```

bad_fit1	<i>Identify potential individual with bad fits based on coefficients and model convergence for a specific model</i>
----------	---

Description

Identify potential individual with bad fits based on coefficients and model convergence

Usage

```
bad_fit1(mod_ls, cutoff = 1000, m = 1)
```

Arguments

mod_ls	A list of list of model generated by rsf_ind
cutoff	Value a coefficient may take to indicate bad fit (default=1000)
m	model number (based on number in list of formula provided to rsf_ind)

Value

A list with first element giving individuals with bad fits based on coefficients and second element containing individuals with bad fit based on convergence

Examples

```
data(goats)
ls1<-list()
ls1[[1]]<-as.formula(STATUS~ELEVATION+SLOPE+ET+ASPECT+HLI+TASP)
ls1[[2]]<-as.formula(STATUS~ET+ASPECT+HLI+TASP)
out<-rsf_ind(goats$ID, data=goats, form_ls=ls1)
bad_fit(out, cutoff=20, m=1) #None
```

cmodel	<i>Remove elements from glm object to save space</i>
--------	--

Description

Remove elements from glm objects (Taken from: <https://www.r-bloggers.com/trimming-the-fat-from-glm-models-in-r/>)

Usage

```
cmodel(cm)
```

Arguments

cm	A glm object
----	--------------

Value

A glm object

eval_ratio	<i>Evaluate ratio of used and random locations of individuals in a RSF table</i>
------------	--

Description

Evaluate ratio of used and random locations of individuals in a RSF table

Usage

```
eval_ratio(id, value)
```

Arguments

id	A vector of individual for each observation
value	A vector indicating if each observation is used (=1) or random(=0)

Value

A list indicating the range in ratio, range in random locations, and range in used location.

goats	<i>goats - Mountain goats data set</i>
-------	--

Description

GPS collar data of mountain goats (*Oreamnos americanus*) from Lele and Keim (2006).

Usage

```
goats
```

Format

A data frame with 19014 rows and 8 variables

ind_coef	<i>Extract individual coefficients</i>
----------	--

Description

Extract individual coefficients.

Usage

```
ind_coef(m = 1, mod_ls, cutoff = 0)
```

Arguments

m	model number (based on number in list of formula provided to rsf_ind)
mod_ls	A list of list of model generated by rsf_ind
cutoff	A cutoff value to exclude individuals with bad fit, default = -1 indicating model that did not converge will be excluded. Values > 0 will exclude based on coefficient

Value

A table of individual coefficients

Examples

```
data(goats)
ls1<-list()
ls1[[1]]<-as.formula(STATUS~ELEVATION+SLOPE+ET+ASPECT+HLI+TASP)
ls1[[2]]<-as.formula(STATUS~ET+ASPECT+HLI+TASP)
out<-rsf_ind(goats$ID, data=goats, form_ls=ls1)
ind_coef(m=1, out)
```

ind_se	<i>Extract individual standard errors</i>
--------	---

Description

Extract individual standard errors

Usage

```
ind_se(m = 1, mod_ls, cutoff = 0)
```

Arguments

m	model number (based on number in list of formula provided to rsf_ind)
mod_ls	A list of list of model generated by rsf_ind
cutoff	A cutoff value to exclude individuals with bad fit, default = -1 indicating model that did not converge will be excluded. Values > 0 will exclude based on coefficient

Value

A table of individual standard errors for each coefficients

Examples

```
data(goats)
ls1<-list()
ls1[[1]]<-as.formula(STATUS~ELEVATION+SLOPE+ET+ASPECT+HLI+TASP)
ls1[[2]]<-as.formula(STATUS~ET+ASPECT+HLI+TASP)
out<-rsf_ind(goats$ID, data=goats, form_ls=ls1)
ind_se(m=1, out)
```

kfoldRSF	<i>Perform kfold cross-validation on a RSF output.</i>
----------	--

Description

Perform kfold cross-validation on a RSF output. Similar to what is recommended in Boyce 2002. Function developped with Mathieu Basille

Usage

```
kfoldRSF(mod, k = 5, nrepet = 10, nbins = 10, jitter = TRUE,
  random = TRUE, method = method, x = m, form_ls = ls,
  reproducible = TRUE)
```

Arguments

mod	A RSF model (glm or glmer)
k	number of fold (default = 5)
nrepet	Number of repetitions (default =10)
nbins	Number of bins (default =10)
jitter	Logical, whether to add some random noise to the predictions (useful when the model is fitted on categorica variables, which can produces error in the ranking process).
reproducible	Logical, whether to use a fixed seed for each repetition.

Value

A data frame with the correlations (cor) and the type of value (type).

kfold_ind	<i>Perform kfold cross-validation at the individual level .</i>
-----------	---

Description

Perform kfold cross-validation at the individual level and return histogram, mean kfold accros individual and min/max value

Usage

```
kfold_ind(m = 1, mod_ls, ls = ls, cutoff = 0, k = 5, nrepet = 5,
          nbins = 10, grph = T)
```

Arguments

m	model number (based on number in list of formula provided to rsf_ind)
mod_ls	A list of list of model generated by rsf_ind
cutoff	A cutoff value to exclude individuals with bad fit, default = -1 indicating model that did not converge will be excluded. Values > 0 will exclude based on coefficient
k	number of fold (default = 5)
nrepet	Number of repetitions (default =10)
nbins	Number of bins (default =10)
jitter	Logical, whether to add some random noise to the predictions (useful when the model is fitted on categorica variables, which can produces error in the ranking process).
reproducible	Logical, whether to use a fixed seed for each repetition.

Value

A data frame with the correlations (cor) and the type of value (type).

Examples

```
data(goats)
ls1<-list()
ls1[[1]]<-as.formula(STATUS~ELEVATION+SLOPE+ET+ASPECT+HLI+TASP)
ls1[[2]]<-as.formula(STATUS~ET+ASPECT+HLI+TASP)
out<-rsf_ind(goats$ID, data=goats, form_ls=ls1)
kfold_ind(m=1, out, ls=ls1)
```

pop_avg	<i>Extract population average of top model and extract individual coefficients</i>
---------	--

Description

Extract population average of top model and extract individual coefficients. Population average can be calculated based on bootstrap (Prokopenko et al. 2016 JAppEco) or weighted based on standard errors (Murtaugh 2007 Ecology)

Usage

```
pop_avg(m = 1, mod_ls, cutoff = 0, method = "boot", nboot = 1000)
```

Arguments

m	model number (based on number in list of formula provided to rsf_ind)
mod_ls	A list of list of model generated by rsf_ind
cutoff	A cutoff value to exclude individuals with bad fit, default = -1 indicating model that did not converge will be excluded. Values > 0 will exclude based on coefficient
method	If = "boot", population average is based on bootstrap, if = "murtaugh" based on standard errors weighting. See Prokopenko et al 2016 or Murtaugh 2007 for details.
nboot	Number of bootstrap iterations, default = 1000. Only applicable if method = "boot".

Value

A list containing a table population average with confidence intervals and a table of individual coefficients

Examples

```
data(goats)
ls1<-list()
ls1[[1]]<-as.formula(STATUS~ELEVATION+SLOPE+ET+ASPECT+HLI+TASP)
ls1[[2]]<-as.formula(STATUS~ET+ASPECT+HLI+TASP)
out<-rsf_ind(goats$ID, data=goats, form_ls=ls1)
pop_avg(m=1, out, method="murtaugh")
```

resample_rsf	<i>Resample a RSF table to keep constant ratio of used/random locations accross individuals</i>
--------------	---

Description

Resample a RSF table to keep constant ratio of used/random locations accross individuals. Resampling is done with replacement.

Usage

```
resample_rsf(data, id = "Id_Year", value = "Value", ratio = 3)
```

Arguments

data	The RSF dataset to resample
id	A vector of individual for each observation
value	A vector indicating if each observation is used (=1) or random(=0)
ratio	The ratio of random:used location (default =3, meaning 3 random locations for each used location)

Value

A RSF dataset

rf	<i>Extraction of proximity from random forest classification</i>
----	--

Description

Apply random forest classification

Usage

```
rf(coef, ntree = 10000, ...)
```

Arguments

coef	A matrix of model coefficient (from function ind_coef)
------	--

Value

A proximity matrix

Examples

```
data(goats)
ls1<-list()
ls1[[1]]<-as.formula(STATUS~ELEVATION+SLOPE+ET+ASPECT+HLI+TASP)
ls1[[2]]<-as.formula(STATUS~ET+ASPECT+HLI+TASP)
out<-rsf_ind(goats$ID, data=goats, form_ls=ls1)
coef<-ind_coef(m=1, out)
prox<-rf(coef)
```

rm_bad_fit

*Remove potential individual with bad fits based on coefficients***Description**

Remove potential individual with bad fits based on coefficients

Usage

```
rm_bad_fit(mod_ls, cutoff = 1000)
```

Arguments

mod_ls	A list of list of model generated by rsf_ind
cutoff	Value a coefficient may take to indicate bad fit (default=1000)

Value

A list excluding individual with bad fits.

rm_bad_fit1

*Remove potential individual with bad fits based on coefficients for a specific model***Description**

Remove potential individual with bad fits based on coefficients

Usage

```
rm_bad_fit1(mod_ls, cutoff = 1000, m = 1)
```

Arguments

mod_ls	A list of list of model generated by rsf_ind
cutoff	Value a coefficient may take to indicate bad fit (default=1000)
m	model number (based on number in list of formula provided to rsf_ind)

Value

A list excluding individual with bad fits.

rm_conv_fit	<i>Remove potential individual with bad fits based on model convergence</i>
-------------	---

Description

Remove potential individual with bad fits based on model convergence

Usage

```
rm_conv_fit(mod_ls)
```

Arguments

mod_ls A list of list of model generated by rsf_ind

Value

A list excluding individual with bad fits.

rm_conv_fit1	<i>Remove potential individual with bad fits based on model convergence for a specific model</i>
--------------	--

Description

Remove potential individual with bad fits based on model convergence

Usage

```
rm_conv_fit1(mod_ls, m = 1)
```

Arguments

m model number (based on number in list of formula provided to rsf_ind)

Value

A list excluding individual with bad fits.

rsf_ind	<i>Apply a list of candidate models to multiple individuals</i>
---------	---

Description

Apply rsf_mod to each individual of a dataset

Usage

```
rsf_ind(id, data, form_ls, cleanModel = F, method = "glm.fit")
```

Arguments

id	A vector indicating the individuals
data	The dataset containing all data
form_ls	A list of formulas for the different candidate models
method	Whether typical glm or bias-reduction glm should be fitted (default="glm.fit") (see package brglm)
cleanModel	Whether the model should be "cleaned" to save memory space (default = F)

Value

A list of list of glm objects

Examples

```
data(goats)
ls1<-list()
ls1[[1]]<-as.formula(STATUS~ELEVATION+SLOPE+ET+ASPECT+HLI+TASP)
ls1[[2]]<-as.formula(STATUS~ET+ASPECT+HLI+TASP)
out<-rsf_ind(goats$ID, data=goats, form_ls=ls1)
```

rsf_mod	<i>Apply a list of candidate models to a single individual</i>
---------	--

Description

Apply a list of candidate models to a single individual

Usage

```
rsf_mod(sub, form_ls, cleanModel = F, method = method)
```

Arguments

sub	A subset of data from a single individual
form_ls	A list of formulas for the different candidate models
method	Whether typical glm or bias-reduction glm should be fitted (see package brglm)
cleanModel	Whether the model should be "cleaned" to save memory space

rsf_mod

13

Value

A list of glm objects

Index

*Topic **datasets**

goats, [5](#)

aictab_ind, [2](#)

bad_fit, [2](#)

bad_fit1, [3](#)

cmodel, [4](#)

eval_ratio, [4](#)

goats, [5](#)

ind_coef, [5](#)

ind_se, [6](#)

kfold_ind, [7](#)

kfoldRSF, [6](#)

pop_avg, [8](#)

resample_rsf, [9](#)

rf, [9](#)

rm_bad_fit, [10](#)

rm_bad_fit1, [10](#)

rm_conv_fit, [11](#)

rm_conv_fit1, [11](#)

rsf_ind, [12](#)

rsf_mod, [12](#)