Package 'modEvA'

November 4, 2014

Type Package

Title Model Evaluation and Analysis

Version 0.7.2

Date 2014-10-10

Author Barbosa A.M., Brown J.A., Jimenez-Valverde A., Real R.

Maintainer A. Marcia Barbosa <barbosa@uevora.pt>

Description The modEvA package can build and analyse species distribution models and evaluate their performance. Includes functions for performing variation partitioning, assessing the false discovery rate, analysing multicollinearity among variables, calculating several measures of model discrimination and calibration, optimizing prediction thresholds based on a number of criteria, performing multivariate environmental similarity surface (MESS) analysis, and displaying various analytical plots. Includes also a sample data set with species occurrences and predictor variables.

License GPL-3

R topics documented:

---

modEvA-package *Model Evaluation and Analysis*

Description

The modEvA package can build and analyse species distribution models and evaluate their performance. Includes functions for performing variation partitioning, assessing the false discovery rate, analysing multicollinearity among variables, calculating several measures of model discrimination and calibration, optimizing prediction thresholds based on a number of criteria, performing multivariate environmental similarity surface (MESS) analysis, and displaying various analytical plots. Includes also a sample data set with species occurrences and predictor variables.

Details

Package:	modEvA
Type:	Package
Version:	0.7.2
Date:	2014-11-04
License:	GPL-3

Author(s)

Barbosa A.M., Brown J.A., Jimenez-Valverde A., Real R.

A. Marcia Barbosa <barbosa@uevora.pt>

References

Barbosa A.M., Real R., Munoz A.R. & Brown J.A. (2013) New measures for assessing model equilibrium and prediction mismatch in species distribution models. Diversity and Distributions 19: 1333-1338 (DOI: 10.1111/ddi.12100)

See Also

PresenceAbsence, **ROCR**, **verification**, **dismo**

Examples

```
# load sample data and see the column names:
data(rotif.env)
names(rotif.env)

# analyse multicollinearity among predictor variables:
multicol(rotif.env[ , 5:17])

# build models for presence-absence of rotifer species:
mods <- multGLM(data = rotif.env, sp.cols = 18:47, var.cols = 5:17,
step = FALSE, trim = TRUE)

# analyse the resulting predictions table:
head(mods$predictions)

# plot the 1st model:
plotGLM(model = mods$models[[1]])

# calculate the area under the ROC curve for the 1st model:
AUC(model = mods$models[[1]])

# calculate some more threshold-based measures for this model:
threshMeasures(model = mods$models[[1]], thresh = "preval")

# calculate optimal thresholds based on several criteria:
optiThresh(model = mods$models[[1]])

# calculate the optimal threshold balancing two evaluation measures:
optiPair(model = mods$models[[1]], measures = c("Omission", "Commission"))

# calculate the explained deviance, Hosmer-Lemeshow goodness-of-fit,
# Miller's calibration stats, and (pseudo) R-squared values for the 1st model:

Dsquared(model = mods$models[[1]])

HLfit(model = mods$models[[1]])

MillerCalib(model = mods$models[[1]])
```

```
RsqGLM(model = mods$models[[1]])
```

---

arrangePlots *Arrange plots*

Description

Get an appropriate row/column combination (for par(mfrow)) for arranging a given number of plots within a plotting window.

Usage

```
arrangePlots(n.plots, landscape = FALSE)
```

Arguments

n.plots	number of plots to be placed in the graphics device.
landscape	logical, whether the plotting window should be landscape/horizontal (number of columns larger than the number of rows) or not. The value does not make a difference if the number of plots makes for a square plotting window.

Details

This function is used internally by optiThresh, but can also be useful outside it.

Value

An integer vector of the form c(nr, nc) indicating, respectively, the number of rows and of columns of plots to set in the graphics device.

Author(s)

A. Marcia Barbosa

See Also

plot, layout

Examples

```
arrangePlots(10)

arrangePlots(10, landscape = TRUE)


# a more practical example:

data(rotif.env)
```

```
names(rotif.env)

# say you want to plot columns 5 to 9 in a nicely arranged plotting window:

par(mfrow = arrangePlots(length(5:9)))

for (i in 5:9) {
  plot(1:nrow(rotif.env), rotif.env[, i])
}
```

AUC *Area Under the ROC Curve*

Description

This function calculates the Area Under the Curve of the receiver operating characteristic (ROC) plot, for either a model object of class "glm", or two matching vectors of observed (binary, 1 for occurrence vs. 0 for non-occurrence) and predicted (continuous, e.g. occurrence probability) values, respectively. The AUC is a measure of the overall discrimination power of the predictions, or the probability that an occurrence site has a higher predicted value than a non-ocurrence site.

Usage

```
AUC(obs = NULL, pred = NULL, model = NULL, interval = 0.01, simplif = FALSE,
plot = TRUE, plot.values = TRUE, plot.preds = FALSE, grid = FALSE,
xlab = c("False positive rate", "(1-specificity)"),
ylab = c("True positive rate", "(sensitivity)"), main = "ROC curve", ...)
```

Arguments

obs	a vector of observed presences (1) and absences (0) or another binary response variable.
pred	a vector with the corresponding predicted values of presence probability, habitat suitability, environmental favourability or alike. Must be of the same length and in the same order as obs.
model	instead of (and overriding) obs and pred, you can provide a model object of class "glm".
interval	interval of threshold values at which to calculate the true and false positive and negative rates. Defaults to 0.01. Ignored if simplif = TRUE. Note that this does not affect the obtained AUC value (although it can affect the size of the plotted ROC curve, especially when prevalence is low), as the AUC is calculated with the Mann-Whitney U statistic and is therefore threshold-independent.
simplif	logical, whether to use a faster version that outputs only the AUC value (may be useful for calculating AUC for large numbers of species in a loop, for example). Note that the ROC plot is not drawn when simplif = TRUE.
plot	logical, whether or not to plot the ROC curve. Defaults to TRUE.

plot.values	logical, whether or not to show in the plot the values associated to the curve (e.g., the AUC). Defaults to TRUE.
plot.preds	logical, whether or not to plot the proportion of analysed model predictions (through proportionally sized circles) at each threshold. Experimental. Defaults to FALSE.
grid	logical, whether or not to add a grid to the plot, marking the analysed thresholds. Defaults to FALSE.
xlab	label for the x axis.
ylab	label for the y axis.
main	title for the plot.
...	further arguments to be passed to the plot function.

Details

Like the model evaluation measures calculated by the threshMeasures function, the area under the ROC curve (AUC) assesses the discrimination performance of a model; but unlike them, it does not require the choice of a particular threshold above which to consider that a model predicts species presence, but rather averages discrimination performance over all possible thresholds. Mind that the AUC has been widely criticized (e.g. Lobo et al. 2008, Jimenez-Valverde et al. 2013), yet it is still among the most widely used metrics in model evaluation. It is highly correlated with species prevalence, so this value is also provided by the AUC function (if simplif = FALSE, the default) for reference. Although there are functions to calculate the AUC in other R packages (e.g. **ROCR**, **PresenceAbsence**, **verification**, **Epi**), the AUC function is more compatible with the remaining functions in **modEvA** and can be applied not only to a set of observed versus predicted values, but also directly to a model object of class "glm".

Value

If simplif = TRUE, the function returns only the AUC value (a numeric value between 0 and 1). Otherwise (the default), it plots the curve and returns a list with the following components:

thresholds	a data frame of the true and false positives, the sensitivity and specificity of the predictions, and the number of predicted values at each analysed threshold.
N	the total number of obervations.
prevalence	the proportion of occurrences in the data (which correlates with the AUC).
AUC	the value of the AUC).
AUCratio	the ratio of the obtained AUC value to the null expectation (0.5).

Author(s)

A. Marcia Barbosa

References

Lobo, J.M., Jimenez-Valverde, A. & Real, R. (2008). AUC: a misleading measure of the performance of predictive distribution models. Global Ecology and Biogeography 17: 145-151

Jimenez-Valverde, A., Acevedo, P., Barbosa, A.M., Lobo, J.M. & Real, R. (2013). Discrimination capacity in species distribution models depends on the representativeness of the environmental domain. Global Ecology and Biogeography 22: 508-516

See Also

threshMeasures

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))

AUC(model = mod, simplif = TRUE)

AUC(model = mod)

AUC(model = mod, grid = TRUE, plot.preds = TRUE)


# you can also use AUC with vectors of observed and predicted values
# instead of with a model object:

prob <- predict(mod, data = rotif.env, type = "response")

AUC(obs = rotif.env$Abrigh, pred = prob)
```

Dsquared *Proportion of deviance explained by a GLM*

Description

This function calculates the (adjusted) amount of deviance accounted for by a generalized linear model.

Usage

```
Dsquared(obs = NULL, pred = NULL, model = NULL, adjust = FALSE)
```

Arguments

obs	a vector of presence/absence or other binary (1-0) observed data.
pred	a vector of the values predicted by a GLM of the binary observed data.
model	instead of (and overriding) obs and pred, a model object of class "glm".
adjust	logical, whether or not to adjust the value for the number of observations and parameters in the model (see Details). Only works if a model object is provided, otherwise a message will be returned. Defaults to FALSE.

Details

Linear models come with an R-squared value that measures the proportion of variation that the model accounts for. The R-squared is provided with summary(model) in R. For generalized linear models (GLMs), the equivalent is the amount of deviance accounted for (D-squared; Guisan & Zimmermann 2000), but this value is not normally provided with the model summary. The Dsquared function calculates it. There is also an option to calculate the adjusted D-squared, which takes into account the number of observations and the number of predictors, thus allowing direct comparison among different models (Weisberg 1980, Guisan & Zimmermann 2000).

Value

a numeric value indicating the (adjusted) proportion of deviance accounted for by the model.

Author(s)

A. Marcia Barbosa

References

Guisan, A. & Zimmermann, N.E. (2000) Predictive habitat distribution models in ecology. Ecological Modelling 135: 147-186

Weisberg, S. (1980) Applied Linear Regression. Wiley, New York

See Also

glm, plotGLM

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))

Dsquared(model = mod)

Dsquared(model = mod, adjust = TRUE)
```

```
# you can also use \code{Dsquared} with vectors of observed and predicted values
# instead of with a model object
# but in this case you won't obtain an adjusted D-squared value:

prob <- predict(object = mod, data = rotif.env, type = "response")

Dsquared(obs = rotif.env$Abrigh, pred = prob, adjust = TRUE)

Dsquared(obs = rotif.env$Abrigh, pred = prob, adjust = FALSE)
```

---

evaluate *Evaluate a GLM based on the elements of a confusion matrix.*

Description

This functions evaluates the discrimination performance of a model based on the values of a confusion matrix obtained at a particular threshold.

Usage

```
evaluate(a, b, c, d, N = NULL, measure = "CCR")
```

Arguments

a	number of correctly predicted presences
b	number of absences incorrectly predicted as presences
c	number of presences incorrectly predicted as absences
d	number of correctly predicted absences
N	total number of cases. If NULL (the dafault) it is calculated automatically by adding up a, b, c and d.)
measure	a character vector of length 1 indicating the the evaluation measure to use. Type `modEvAmethods("threshMeasures")` for available options.

Details

A number of measures can be used to evaluate continuous model predictions against observed binary occurrence data (Fielding & Bell 1997; Liu et al. 2011; Barbosa et al. 2013). The `evaluate` function can calculate a few threshold-based discrimination measures from the values of a confusion matrix obtained at a particular threshold. The `evaluate` function is used internally by `threshMeasures`. It can also be accessed directly by the user, but it is usually more practical to use `threshMeasures`, which calculates the confusion matrix automatically.

Value

The value of the specified evaluation measure.

Author(s)

A. Marcia Barbosa

References

Barbosa A.M., Real R., Munoz A.R. & Brown J.A. (2013) New measures for assessing model equilibrium and prediction mismatch in species distribution models. Diversity and Distributions, 19: 1333-1338

Fielding A.H. & Bell J.F. (1997) A review of methods for the assessment of prediction errors in conservation presence/absence models. Environmental Conservation 24: 38-49

Liu C., White M., & Newell G. (2011) Measuring and comparing the accuracy of species distribution models with presence-absence data. Ecography, 34, 232-243.

See Also

threshMeasures

Examples

```
evaluate(23, 44, 21, 34)

evaluate(23, 44, 21, 34, measure = "TSS")
```

---

evenness *Evenness in a binary vector.*

Description

For building and evaluating species distribution models, the porportion of presences (prevalence) of a species and the balance between the number of presences and absences may be issues to take into account (e.g. Jimenez-Valverde & Lobo 2006, Barbosa et al. 2013). The evenness function calculates the presence-absence balance in a binary (e.g., presence/absence) vector.

Usage

```
evenness(obs)
```

Arguments

obs	a vector of binary observations (e.g. 1 or 0, male or female, disease or no disease, etc.)

Value

A number ranging between 0 when all values are the same, and 1 when there are the same number of cases with each value in obs.

Author(s)

A. Marcia Barbosa

References

Barbosa A.M., Real R., Munoz A.R. & Brown J.A. (2013) New measures for assessing model equilibrium and prediction mismatch in species distribution models. Diversity and Distributions, 19: 1333-1338

Jimenez-Valverde A. & Lobo J.M. (2006) The ghost of unbalanced species distribution data in geographical model predictions. Diversity and Distributions, 12: 521-524.

See Also

prevalence

Examples

```
(x <- rep(c(0, 1), each = 5))
(y <- c(rep(0, 3), rep(1, 7)))
(z <- c(rep(0, 7), rep(1, 3)))

prevalence(x)
evenness(x)

prevalence(y)
evenness(y)

prevalence(z)
evenness(z)
```

Fav *Favourability*

Description

Environmental (prevalence-independent) favourability for a species' presence

Usage

```
Fav(obs = NULL, pred = NULL, n1n0 = NULL, model = NULL, sample.preval = NULL,
method = "RBV", true.preval = NULL)
```

Arguments

obs	a vector of 1-0 values of a modelled binary variable.
pred	a vector of predicted probability values for obs, given e.g. by logistic regression.
n1n0	alternatively to obs, an integer vector of length 2 providing the total number of ones and zeros, in this order. This argument is ignored if the obs vector is provided.
model	alternatively to all of the above, a model object of class "glm". If provided, will override any values provided in the arguments described above.
sample.preval	alternatively to obs or n1n0, the prevalence (proportion of positive cases) of the modelled binary variable in the modelled data.
method	either "RBV" for the original Real, Barbosa & Vargas (2006) procedure, or "AT"" for the modification proposed by Albert & Thuiller (2008) (but see Acevedo & Real 2012)
true.preval	the true prevalence (as opposed to sample prevalence), necessary if you want to use the AT method.

Details

Logistic regression (Generalised Linear Model with binomial error distribution and a logit link) is widely used for modelling species' potential distributions using presence/absence data and a set of categorical or continuous predictor variables. However, this GLM incorporates the prevalence (relative proportion of presences and absences) of the species in the training sample, which affects the probability values produced.

Barbosa (2006) and Real, Barbosa & Vargas (2006) proposed an environmental favourability function which is based on logistic regression but cancels out uneven proportions of presences and absences in the modelled data. Favourability thus assesses the extent to which the environmental conditions change the probability of occurrence of a species with respect to its overall prevalence in the study area. Model predictions can, therefore, be directly compared between species with different prevalences. The favourability function is implemented in the modEvA package and is also in the SAM (Spatial Analysis in Macroecology) software (Rangel et al. 2010).

Using simulated data, Albert & Thuiller (2008) proposed a modification to the favourability function, but it requires knowing the true prevalence of the species (not just the prevalence in the studied sample), which is rarely possible in real-world modelling. Besides, this suggestion was based on the misunderstanding that the favourability function was a way to obtain the probability of occurrence when prevalence differs from 50%, which is incorrect (see Acevedo & Real 2012).

To get environmental favourability with either the Real, Barbosa & Vargas ("RBV") or the Albert & Thuiller ("AT") method, you just need to get a probabilistic model (e.g. logistic regression) from your data and then use the Fav function. Input data for this function are either a model object resulting from the glm function, or the presences-absences (1-0) of your species and the corresponding presence probability values, obtained e.g. with predict(mymodel, mydata, type = "response"). Alternatively to the presences-absences, you can provide either the sample prevalence or the numbers of presences and absences. In case you want to use the "AT" method, you also need to provide the true (absolute) prevalence of your species.

Value

A numeric vector of the favourability values corresponding to the input probability values.

Author(s)

A. Marcia Barbosa

References

Acevedo P. & Real R. (2012) Favourability: concept, distinctive characteristics and potential usefulness. Naturwissenschaften 99: 515-522

Albert C.H. & Thuiller W. (2008) Favourability functions versus probability of presence: advantages and misuses. Ecography 31: 417-422.

Barbosa A.M.E. (2006) Modelacion de relaciones biogeograficas entre predadores, presas y parasitos: implicaciones para la conservacion de mamiferos en la Peninsula Iberica. PhD Thesis, University of Malaga (Spain).

Rangel T.F.L.V.B, Diniz-Filho J.A.F & Bini L.M. (2010) SAM: a comprehensive application for Spatial Analysis in Macroecology. Ecography 33: 46-50.

Real R., Barbosa A.M. & Vargas J.M. (2006) Obtaining environmental favourability functions from logistic regression. Environmental and Ecological Statistics 13: 237-245.

See Also

glm, multGLM, prevalence

Examples

```
# obtain a probability model and its predictions:

data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))

prob <- predict(mod, data = rotif.env, type = "response")


# obtain predicted favourability in different ways:

Fav(model = mod)

Fav(obs = rotif.env$Abrigh, pred = prob)

Fav(pred = mod$fitted.values, n1n0 = c(112, 179))

Fav(pred = mod$fitted.values, sample.preval = 0.3849)
```

FDR *False Discovery Rate*

Description

Calculate the false discovery rate (type I error) under repeated testing and determine which variables to select and to exclude from multivariate analysis.

Usage

FDR(data = NULL, sp.cols = NULL, var.cols = NULL, pvalues = NULL, model.type, family = "binomial", correction = "fdr", q = 0.05, verbose = TRUE)

Arguments

data	a data frame containing the response and predictor variables (one in each column).
sp.cols	index number of the column containing the response variable (currently implemented for only one response variable at a time).
var.cols	index numbers of the columns containing the predictor variables.
pvalues	optionally, instead of data, sp.cols and var.cols, a data frame with the names of the predictor variables in the first column and their bivariate p-values (obtained elsewhere) in the second column. Example: pvalues <- data.frame(var = letters[1:5], pval = c(0.02, 0.004, 0.07, 0.03, 0.05)).
model.type	either "LM" (linear model, for continuous response variables) or "GLM" (generalized linear models, for binary or other variables for which such models are more appropriate)
family	if model.type = "GLM", the error distribution and link function (see glm or family for details); defaults to "binomial" (for binary logistic regression).
correction	the correction procedure to apply to the p-values; see p.adjust.methods for available options and p.adjust for more information. The default is "fdr".
q	the threshold value of FDR-corrected significance above which to reject variables. Defaults to 0.05.
verbose	logical, whether to report a short description of the results.

Details

It is common in ecology to search for statistical relationships between species' occurrence and a set of predictor variables. However, when a large number of variables is analysed (compared to the number of observations), false findings may arise due to repeated testing. Garcia (2003) recommended controlling the false discovery rate (FDR; Benjamini & Hochberg 1995) in ecological studies. The p.adjust R function performs this and other corrections to the significance (p) values of variables under repeated testing. The FDR function performs repeated regressions (either linear or binary logistic) or uses already-obtained p values for a set of variables; calculates the FDR with

p.adjust; and shows which variables should be retained for or excluded from further multivariate analysis according to their corrected p values (see, for example, Barbosa, Real & Vargas 2009).

The FDR function uses the Benjamini & Hochberg ("BH") correction by default, but check the p.adjust documentation for other available methods. Input data may be the response variable (for example, the presence-absence or abundance of a species) and the predictors (a table with one independent variable in each column, with the same number of rows and in the same order as the response); there should be no missing values in the data. Model type can be either "LM" (linear model) for continuous response variables such as abundance, or "GLM" (generalized linear model) for binary responses such as presence-absence. Alternatively, you may already have performed the univariate regressions and have a set of variables and corresponding p values which you want to correct with FDR; in this case, get a table with your variables' names in the first column and their p values in the second column, and supply it as the pvalues argument to the FDR function (no need to provide response or predictors in this case).

Value

A list with the following components:

exclude	a data frame of the variables to exclude under the chosen criteria, including the variables' names, their bivariate coefficients against the response, their p-value and adjusted p-value.
select	a data frame similar to the above for the variables to select.

Author(s)

A. Marcia Barbosa

References

Barbosa A.M., Real R. & Vargas J.M (2009) Transferability of environmental favourability models in geographic space: The case of the Iberian desman (Galemys pyrenaicus) in Portugal and Spain. Ecological Modelling 220: 747-754

Benjamini Y. & Hochberg Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society, Series B 57: 289-300

Garcia L.V. (2003) Controlling the false discovery rate in ecological research. Trends in Ecology and Evolution 18: 553-554

See Also

p.adjust

Examples

```
data(rotif.env)

names(rotif.env)

FDR(data = rotif.env, sp.cols = 18, var.cols = 5:17, model.type = "GLM")
```

getBins *Get bins of data.*

Description

Get continuous predicted values into bins according to specific criteria.

Usage

getBins(obs = NULL, pred = NULL, model = NULL, id = NULL, bin.method = "quantiles", n.bins = 10, fixed.bin.size = FALSE, min.bin.size = 15, min.prob.interval = 0.1, simplif = FALSE)

Arguments

obs	a vector of 1-0 values of a modelled binary variable.
pred	a vector of the corresponding predicted values.
model	instead of (and overriding) obs and pred, a model object of class "glm".
id	optional vector of row identifiers; must be of the same length and in the same order of obs and pred (or of the cases used to build model)
bin.method	the method with which to divide the values into bins. Type modEvAmethods("getBins") for available options.
n.bins	the number of bins in which to divide the data.
fixed.bin.size	logical, whether all bins should have the same size.
min.bin.size	integer value defining the minimum number of observations to include in each bin. The default minimum is 15, following Jovani & Tella (2006).
min.prob.interval	minimum range of probability values in each bin. Defaults to 0.1.
simplif	logical, whether to calculate a simplified faster version (usd internally in other functions). The default is FALSE.

Value

The output of getBins is a list with the following components:

prob.bin	the first and last value of each bin
bins.table	a data frame with the numbers of presences, absences, prevalence, mean probability, etc. in each bin.
N	the total number of observations in the analysis.
n.bins	the total number of bins obtained.

Author(s)

A. Marcia Barbosa

References

Jovani R. & Tella J.L. (2006) Parasite prevalence and sample size: misconceptions and solutions. Trends in Parasitology 22: 214-218.

See Also

HLfit

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))

getBins(model = mod)
```

---

getModEqn *Get model equation*

Description

This function retrieves the equation of a model, to print or apply elsewhere.

Usage

```
getModEqn(model, type = "y", digits = NULL, prefix = NULL, suffix = NULL)
```

Arguments

model	a model object of class 'lm' or glm'.
type	the type of equation to get; can be either "y" (the default, for the linear model equation), "P" (for probabiity) or "F" (for favourability).
digits	the number of digits to which to round the coefficient estimates in the equation.
prefix	the prefix to add to each variable name in the equation.
suffix	the suffix to add to each variable name in the equation.

Details

The summary of a model in R gives you a table of the coefficient estimates and other parameters. Sometimes it may be useful to have a string of text with the model's equation, so that you can present it in an article (e.g. Real et al. 2005) or apply it in a (raster map) calculation, either in R (although here you can usually use the 'predict' function for this) or in a GIS software (e.g. Barbosa et al. 2010). The getModEqn function gets this equation for linear or generalized linear models.

By default it prints the "y" linear equation, but for generalized linear models you can also set type = "P" (for the equation of probability) or type = "F" (for favourability, which corrects the intercept to eliminate the effect of prevalence - see Real et al. 2006).

If the variables to which you want to apply the model have a prefix or suffix (e.g. prefix = "raster.stack$" for the R raster package, or prefix = "mydata$" for a data frame, or suffix = "@1" in Quantum GIS, or suffix = "@mapset" in GRASS), you can get these in the equation too, using the prefix and/or the suffix argument.

Value

A charachter string of model the equation.

Author(s)

A. Marcia Barbosa

References

Barbosa A.M., Real R. & Vargas J.M. (2010) Use of coarse-resolution models of species' distributions to guide local conservation inferences. Conservation Biology 24: 1378-87

Real R., Barbosa A.M., Martinez-Solano I. & Garcia-Paris, M. (2005) Distinguishing the distributions of two cryptic frogs (Anura: Discoglossidae) using molecular data and environmental modeling. Canadian Journal of Zoology 83: 536-545

Real R., Barbosa A.M. & Vargas J.M. (2006) Obtaining environmental favourability functions from logistic regression. Environmental and Ecological Statistics 13: 237-245

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))

getModEqn(mod)

getModEqn(mod, type = "P", digits = 3, suffix = "@mapset")

getModEqn(mod, type = "F", digits = 2)
```

getPreds *Get model predictions*

Description

This function allows getting the predictions of multiple models when applied to a given dataset. It can be useful if you have a list of model objects (e.g. resulting from multGLM) and want to apply them to a new data set containing the same variables for another region or time period. There are options to include the logit link (y) and/or Favourability (see Fav).

Usage

```
getPreds(data, models, id.col = NULL, Y = FALSE, P = TRUE, Favourability = FALSE,
incl.input = TRUE)
```

Arguments

data	the data frame to which to apply the models to get their predictions; must contain all variables (with the same names, case-sensitive) included in any of the models.
models	a list of model objects obtained e.g. with function glm or multGLM.
id.col	optionally, the index number of a column of data containing row identifiers, to be included in the result. Ignored if incl.input = TRUE.
Y	logical, whether to include the logit link (y) value in the predictions.
P	logical, whether to include the probability value in the predictions.
Favourability	logical, whether to include Favourability in the predictions (see Fav).
incl.input	logical, whether to include input columns in the output. The default is TRUE.

Value

This function returns a data frame containing the model predictions, next to the id.col if provided, and to the input data if incl.input = TRUE.

Author(s)

A. Marcia Barbosa

See Also

multGLM, predict

Examples

```
data(rotif.env)

names(rotif.env)


# identify rotifer data corresponding to the Eastern and Western hemispheres:

unique(rotif.env$CONTINENT)

rotif.env$HEMISPHERE <- "Eastern"

rotif.env$HEMISPHERE[rotif.env$CONTINENT %in%
c("NORTHERN_AMERICA", "SOUTHERN_AMERICA")] <- "Western"

head(rotif.env)


# separate the rotifer data into hemispheres

east.hem <- rotif.env[rotif.env$HEMISPHERE == "Eastern", ]
west.hem <- rotif.env[rotif.env$HEMISPHERE == "Western", ]


# make models for 3 of the species in rotif.env based on their distribution
# in the Eastern hemisphere:

mods <- multGLM(east.hem, sp.cols = 18:20, var.cols = 5:17, id.col = 1,
step = FALSE, FDR = FALSE, trim = FALSE)


# get the models' predictions for the Western hemisphere dataset:

preds <- getPreds(west.hem, models = mods$models, P = TRUE, Favourability = TRUE)

head(preds)
```

HLfit *Hosmer-Lemeshow goodness of fit*

Description

Calculate a model's calibration performance (reliability) with the Hosmer & Lemeshow goodness-of-fit statistic.

Usage

```
HLfit(obs = NULL, pred = NULL, model = NULL, bin.method = "quantiles", n.bins = 10,
```

fixed.bin.size = FALSE, min.bin.size = 15, min.prob.interval = 0.1, simplif = FALSE, alpha = 0.05, plot = TRUE, plot.values = TRUE, plot.bin.size = TRUE, xlab = "Predicted probability", ylab = "Observed prevalence", ...)

Arguments

obs	a vector of observed presences (1) and absences (0) or another binary response variable
pred	a vector with the corresponding predicted probabilities as given e.g. by logistic regression
model	instead of (and overriding) obs and pred, a model object of class "glm"
bin.method	the method for grouping the records into bins within which to compare predicted probability to observed prevalence; type modEvAmethods("getBins") for available options.
n.bins	the number of bins to use if bin.method = n.bins.
fixed.bin.size	logical, whether to force bins to have the same size whenever possible.
min.bin.size	the minimum number of records in each bin (minimum for significant comparisons is at least 15 according to Jovani & Tella 2006).
min.prob.interval	the minimum interval (range) of probability values within each bin.
simplif	logical, wheter to perform a faster version returning only the basic statistics.
alpha	alpha value for confidence intervals if plot = TRUE.
plot	logical, whether to produce a plot of the results.
plot.values	logical, whether to report measure values in the plot.
plot.bin.size	logical, whether to report bin sizes in the plot.
xlab	label for the x axis.
ylab	label for the y axis.
...	further arguments to pass to the plot function.

Details

Most of the commonly used measures for evaluating model performance focus on discrimination capacity, i.e., how well the model is capable of distinguishing presences from absences (after the model's continuous predictions of presence probability or alike are converted to binary predictions of presence or absence). However, there is another important facet of model evaluation: calibration or reliability, i.e., the relationship between predicted probability and observed prevalence (Pearce & Ferrier 2000; Jimenez-Valverde et al. 2013). The HLfit function measures model reliability with the Hosmer & Lemeshow goodness-of-fit statistic (Hosmer & Lemeshow 1980).

Note that this statistic has some limitations and caveats (see e.g. http://www.statisticalhorizons.com/hosmer-lemeshow), mainly due to the need to group the values into bins within which to compare probability and prevalence, and the influence of the binning method on the result. The HLfit function can use several binning methods, which are implemented in the getBins function and can be accessed by typing modEvAmethods("getBins"). You should therefore evaluate your model using different binning methods, to see if the results change significantly.

Value

HLfit returns a list with the following components:

bins.table	a data frame of the obtained bins and the values resulting from the hosmer-Lemeshow goodness-of-fit analysis.
chi.sq	the value of the Chi-squared test.
DF	the number of degrees of freedom.
p.value	the p-value of the Hosmer-Lemeshow test. Note that this is one of those tests for which higher p-values are better.

Note

The 4 lines of code from "observed" to "p.value" were adapted from the 'hosmerlem' function available at http://www.stat.sc.edu/~hitchcock/diseaseoutbreakRexample704.txt. The plotting code was loosely based on the calibration.plot function in package **PresenceAbsence**. HLfit still needs some code simplification, and may fail for some datasets and binning methods. Fixes are being applied. Feedback is welcome.

Author(s)

A. Marcia Barbosa

References

Hosmer D.W. & Lemeshow S. (1980) A goodness-of-fit test for the multiple logistic regression model. Communications in Statistics, A10: 1043-1069

Jimenez-Valverde A., Acevedo P., Barbosa A.M., Lobo J.M. & Real R. (2013) Discrimination capacity in species distribution models depends on the representativeness of the environmental domain. Global Ecology and Biogeography, 22: 508-516

Jovani R. & Tella J.L. (2006) Parasite prevalence and sample size: misconceptions and solutions. Trends in Parasitology 22: 214-218

Pearce J. & Ferrier S. (2000) Evaluating the Predictive Performance of Habitat Models Developed using Logistic Regression. Ecological Modeling, 133: 225-245

See Also

getBins, threshMeasures, AUC

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))

prob <- predict(mod, data = rotif.env, type = "response")
```

```
HLfit(model = mod, bin.method = "n.bins", min.prob.interval = 0.1)

HLfit(obs = rotif.env$Abrigh, pred = prob, bin.method = "round.prob", n.bins = 10)
```

---

integerCols *Classify integer columns*

Description

This function detects which numeric columns in a data frame contain only whole numbers, and converts those columns to integer class, so that they take up less space.

Usage

```
integerCols(data)
```

Arguments

data	a data frame containing possibly integer columns classified as numeric.

Value

The function returns a data frame with the same columns as data, but with those that are numeric and contain only whole numbers (possibly including NA) now classified as integer.

Author(s)

A. Marcia Barbosa

See Also

is.integer, as.integer, multConvert

Examples

```
dat <- data.frame(
  var1 = 1:10,
  var2 = as.numeric(1:10),
  var3 = as.numeric(c(1:4, NA, 6:10)),
  var4 = as.numeric(c(1:3, NaN, 5, Inf, 7, -Inf, 9:10)),
  var5 = as.character(1:10),
  var6 = seq(0.1, 1, by = 0.1),
  var7 = letters[1:10]
)  # creates a sample data frame

dat

str(dat)
```

```
# var2 classified as 'numeric' but contains only whole numbers
# var3 same as var2 but containing also NA values
# var4 same as var2 but containing also NaN and infinite values
# var5 contains only whole numbers but initially classified as factor

dat <- integerCols(dat)

str(dat)
# var2 and var3 now classified as 'integer'
# var4 remains as numeric because contains infinite and NaN (not integer) values
# var5 remains as factor
```

MESS *Multivariate Environmental Similarity Surfaces based on a data frame*

Description

This function performs the MESS analysis of Elith et al. (2010) to determine the extent of the environmental differences between model training and model projection (extrapolation) data. It is applicable to variables in a matrix or data frame.

Usage

```
MESS(V, P)
```

Arguments

V	a matrix or data frame containing the variables (one in each column) in the reference (training) dataset.
P	a matrix or data frame containing the same variables in the area to which the model(s) will be projected. Variables (columns) must be in the same order as in V, and colnames(P) must exist.

Details

When model predictions are projected into regions, times or spatial resolutions not analysed in the training data, it may be important to measure the similarity between the new environments and those in the training sample (Elith et al. 2010), as models are not so reliable when predicting outside their domain (Barbosa et al. 2009). The Multivariate Environmental Similarity Surfaces (MESS) analysis measures the similarity in the analysed variables between any given locality in the projection dataset and the localities in the reference (training) dataset (Elith et al. 2010).

MESS analysis is implemented in the MAXENT software (Phillips et al. 2006) and in the **dismo** R package, but there it requires input variables in raster format. This implies not only the use of complex spatial data structures, but also that the units of analysis are rectangular pixels, whereas we often need to model distribution data recorded on irregular units (e.g. provinces, river basins), or on equal-area units (e.g. UTM cells, equal-area hexagons), whereas pixels vary in area between the equator and the poles. The MESS function computes this analysis for variables in a data frame, where localities (in rows) may be of various sizes and shapes.

Value

The function returns a data frame with the same column names as P, plus a column named TOTAL, quantifying the similarity between each point in the projection dataset and those in the reference dataset. Negative values indicate localities that are environmentally dissimilar from the reference region. The last column, MoD, indicates which of the column names of P corresponds to the most dissimilar variable, i.e., the limiting factor or the variable that drives the MESS in that locality (Elith et al. 2010).

Note

A new, apparently more complete method for analysing environmental dissimilarities has recently become available (see https://www.climond.org/ExDet.aspx). We are planning on implementing a dataframe version of that too in the **modEvA** package.

Author(s)

Alberto Jimenez-Valverde, A. Marcia Barbosa

References

Barbosa A.M., Real R. & Vargas J.M. (2009) Transferability of environmental favourability models in geographic space: the case of the Iberian desman (Galemys pyrenaicus) in Portugal and Spain. Ecological Modelling 220: 747-754

Elith J., Kearney M. & Phillips S. (2010) The art of modelling range-shifting species. Methods in Ecology and Evolution 1: 330-342

Phillips S.J., Anderson R.P. & Schapire R.E. (2006) Maximum entropy modeling of species geographic distributions. Ecological Modelling 190: 231-259

See Also

mess in package **dismo**; OA

Examples

```
# load the sample data:

data(rotif.env)


# add a column specifying the hemisphere:

unique(rotif.env$CONTINENT)

rotif.env$HEMISPHERE <- "Eastern"

rotif.env$HEMISPHERE[rotif.env$CONTINENT %in%
c("NORTHERN_AMERICA", "SOUTHERN_AMERICA")] <- "Western"

head(rotif.env)
```

```
# perform a MESS analysis
# suppose you'll extrapolate models from the Western hemisphere (Americas)
# to the Eastern hemisphere (rest of the world):

names(rotif.env)  # variables are in columns 5:17

mess <- MESS(V = subset(rotif.env, HEMISPHERE == "Western", select = 5:17),
P = subset(rotif.env, HEMISPHERE == "Eastern", select = 5:17))

head(mess)
range(mess[ , "TOTAL"])
```

MillerCalib *Miller's calibration satistics for logistic regression models*

Description

This function calculates Miller's (1991) calibration statistics for a generalized linear model with binomial distribution and logistic link, namely the intercept and slope of the regression of the response variable on the logit of predicted probabilities. Optionally and by default, it also plots the corresponding regression line (black) over the reference diagonal (grey).

Usage

```
MillerCalib(obs = NULL, pred = NULL, model = NULL, plot = TRUE, plot.values = TRUE, digits = 4, xlab = "",
```

Arguments

obs	a vector of observed presences (1) and absences (0) or another binary response variable.
pred	a vector with the corresponding predicted values of presence probability. Must be of the same length and in the same order as obs.
model	instead of (and overriding) obs and pred, you can provide a model object of class "glm" and family "binomial".
plot	logical, whether to plot the regression line over the reference diagonal. Defaults to TRUE.
plot.values	logical value indicating whether to report the statistics in the plot. Defaults to TRUE.
digits	integer number indicating the number of digits to which the values in the plot should be rounded. Dafaults to 4. Ignored if plot or plot.values are set to FALSE.
xlab	label for the x axis.
ylab	label for the y axis.
main	title for the plot.
...	additional arguments to pass to plot.

Details

Calibration or reliability measures how a model's predicted probabilities relate to observed species prevalence or proportion of presences in the modelled data (Pearce & Ferrier 2000; Wintle et al. 2005; Franklin 2010). If predictions are perfectly calibrated, the slope will equal 1 and the intercept will equal 0, so the model's calibation line will perfectly overlap with the reference diagonal. Note that Miller's statistics assess the model globally: a model is well calibrated if the average of all predicted probabilities equals the proportion of presences in the modelled data. Good calibration is always attained on the same data used for building the model (Miller 1991).

Value

This function returns a list of two integer values:

intercept	the calibration intercept.
slope	the calibration slope.

If plot = TRUE, a plot will be produced with the calibration line (black) and the reference diagonal (dashed grey).

Author(s)

A. Marcia Barbosa

References

Franklin, J. (2010) Mapping Species Distributions: Spatial Inference and Prediction. Cambridge University Press, Cambridge.

Miller M.E., Hui S.L. & Tierney W.M. (1991) Validation techniques for logistic regression models. Statistics in Medicine, 10: 1213-1226

Pearce J. & Ferrier S. (2000) Evaluating the predictive performance of habitat models developed using logistic regression. Ecological Modelling, 133: 225-245

Wintle B.A., Elith J. & Potts J.M. (2005) Fauna habitat modelling and mapping: A review and case study in the Lower Hunter Central Coast region of NSW. Austral Ecology, 30: 719-738

See Also

HLfit, RsqGLM

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))

MillerCalib(model = mod)
```

```
# you can also use MillerCalib with vectors of observed and predicted values
# instead of a model object:

prob <- predict(mod, data = rotif.env, type = "response")

MillerCalib(obs = rotif.env$Abrigh, pred = prob)
```

---

modelTrim *Trim off non-significant variables from a model*

Description

This function performs a stepwise removal of non-significant variables from a model.

Usage

```
modelTrim(model, method = "summary", alpha = 0.05)
```

Arguments

model	a model object.
method	the method for getting the individual p-values. Can be either "summary" for the p-values of the coefficient estimates, or "anova" for the p-values of the variables themselves (see Details).
alpha	the p-value above which a variable is removed.

Details

Stepwise variable selection is a common procedure for simplifying models. It maximizes predictive efficiency in an objective and reproducible way, and is useful when the individual importance of the predictors is not known a priori (Hosmer & Lemeshow, 2000). The step R function performs such procedure using an information criterion (AIC) to select the variables, but it often leaves variables that are not significant in the model. Such variables can be subsequently removed with a manual stepwise procedure (e.g. Crawley 2007, p. 442; Barbosa & Real 2010, 2012; Estrada & Arroyo 2012). The modelTrim function performs such removal automatically until all remaining variables are significant. It can also be applied to a full model (i.e., without previous use of the step function), as it serves as a backward stepwise selection procedure based on the significance of the coefficients (if method = "summary", the default) or on the significance of the variables themselves (if method = "anova", better when there are categorical variables in the model).

Value

The input model object after removal of non-significant variables.

Author(s)

A. Marcia Barbosa

References

Barbosa A.M. & Real R. (2010) Favourable areas for expansion and reintroduction of Iberian lynx accounting for distribution trends and genetic diversity of the European rabbit. Wildlife Biology in Practice 6: 34-47

Barbosa A.M. & Real R. (2012) Applying fuzzy logic to comparative distribution modelling: a case study with two sympatric amphibians. The Scientific World Journal, Article ID 428206

Crawley M.J. (2007) The R Book. John Wiley & Sons, Chichester (UK)

Estrada A. & Arroyo B. (2012) Occurrence vs abundance models: Differences between species with varying aggregation patterns. Biological Conservation, 152: 37-45

Hosmer D. W. & Lemeshow S. (2000) Applied Logistic Regression (2nd ed). John Wiley and Sons, New York

See Also

step

Examples

```
# load sample data:

data(rotif.env)

names(rotif.env)


# build a stepwise model of a species' occurrence based on some of the variables:

mod <- with(rotif.env, step(glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial)))


# examine the model:

summary(mod)  # contains non-significant variables


# use modelTrim to get rid of non-significan effects:

mod <- modelTrim(mod)

summary(mod)  # only significant variables now
```

---

modEvAmethods *Methods implemented in modEvA functions*

Description

This function allows retrieving the methods available for some of the functions in modEvA, such as threshMeasures, optiThresh, multModEv and getBins.

Usage

```
modEvAmethods(fun)
```

Arguments

fun	a character vector of length 1 specifying the name (in quotes) of the function for which to obtain the available methods.

Value

a character vector of the available methods for the specified function.

Author(s)

A. Marcia Barbosa

See Also

threshMeasures, optiThresh, getBins, multModEv

Examples

```
modEvAmethods("threshMeasures")

modEvAmethods("multModEv")

modEvAmethods("optiThresh")

modEvAmethods("getBins")
```

---

multConvert *Multiple conversion*

Description

This function can simultaneously convert multiple columns of a matrix or data frame.

Usage

```
multConvert(data, conversion, cols = 1:ncol(data))
```

Arguments

data	A matrix or data frame containing columns that need to be converted
conversion	the conversion to apply, e.g. as.factor or a custom-made fucntion
cols	the columns of data to convert

Details

Sometimes we need to change the data type (class, mode) of a variable in R. There are various possible conversions, performed by functions like as.integer, as.factor or as.character. If we need to perform the same conversion on a number of variables (columns) in a data frame, we can convert them all simultaneously using this function. By default it converts all the columns in the data frame, but you can specify just a few of them. multConvert can also be used to apply other kinds of transformations - for example, if you need to divide some of your columns by 100, just write a function to do this and then use multConvert to apply this function to any group of columns.

Value

The input data with the specified columns converted as asked.

Author(s)

A. Marcia Barbosa

Examples

```
data(rotif.env)

str(rotif.env)

# convert the first 4 columns to character:
converted.rotif.env <- multConvert(data = rotif.env, conversion = as.character, cols = 1:4)

str(converted.rotif.env)


names(rotif.env)

# divide some columns by 100:

div100 <- function(x) x / 100

rotif.env.cent <- multConvert(data = rotif.env, conversion = div100, cols = c(6:10, 12:17))

head(rotif.env.cent)
```

multGLM *Multiple GLMs*

Description

This function calculates generalized linear models for a set of species in a data frame.

Usage

```
multGLM(data, sp.cols, var.cols, id.col = NULL, family = "binomial",
test.sample = 0, FDR = FALSE, step = TRUE, trace = 0, start = "null.model",
direction = "both", Y = FALSE, P = TRUE, Favourability = TRUE, sep = "_",
group.preds = TRUE, trim = TRUE, ...)
```

Arguments

data	a data frame containing your species' binary (0/1) occurrence data and the predictor variables.
sp.cols	index numbers of the columns containing the species data to be modelled.
var.cols	index numbers of the columns containing the predictor variables to be used.
id.col	(optional) index number of column containing the row identifiers (if defined, it will be included in the output predictions data frame).
family	argument to be passed to the glm function; only 'binomial' is implemented in multGLM so far.
test.sample	a subset of data to set aside for subsequent model testing. Can be a value between 0 and 1 for a proportion of the data to choose randomly (e.g. 0.2 for 20%), or an integer number for a particular number of cases to choose randomly among the records in data, or a vector of integers for the index numbers of the particular rows to set aside, or "Huberty" for his rule of thumb based on the number of variables (Huberty 1994, Fielding & Bell 1997).
FDR	logical, whether to do a preliminary exclusion of variables based on their bivariate relationship with the response and the false discovery rate (see FDR).
step	logical, whether to use the step function to perform a first stepwise variable selection based on AIC.
trace	if positive, information is printed during the running of step. Larger values may give more detailed information.
start	logical, whether to start with the 'null.model' (so variable selection starts forward) or with the 'full.model' (so selection starts backward). Used only if step = TRUE.
direction	argument to be passed to step specifying the direction of variable selection ('forward', 'backward' or 'both'). Used only if step = TRUE.
Y	logical, whether to include in the output the response in the scale of the predictor variables (logit).

P	logical, whether to include in the output the response in the probability scale (response).
Favourability	logical, whether to apply the Favourability function (Real et al. 2006) and include its results in the output.
sep	separator for the predictions (Y, P and/or F) in the output table. The default is "_". If an 'illegal' separator is provided (such as "+", "-", ":", ","), R automatically converts it to a dot.
group.preds	logical, whether to group together predictions of similar type (Y, P or F) in the output predictions table (e.g. if FALSE: sp1_Y, sp1_P, sp1_F, sp2_Y, sp2_P, sp2_F; if TRUE: sp1_Y, , sp2_Y, sp1_P, sp2_P, sp1_F, sp2_F).
trim	logical, whether to trim non-significant variables off the models using the modelTrim function; can be used whether or not step is TRUE; works as a backward variable elimination procedure based on significance.
...	additional arguments to be passed to modelTrim.

Details

This function automatically calculates binomial GLMs for one or more species (or other binary variables) in a data frame. The function can optionally perform stepwise variable selection (and it does so by default) instead of forcing all variables into the models, starting from either the null model (the default, so selection starts forward) or from the full model (so selection starts backward) and using Akaike's information criterion (AIC) as a variable selection criterion. Instead or subsequently, it can also perform stepwise removal of non-significant variables from the models using the modelTrim function. There is also an optional preliminary selection of the variables with a significant bivariate relationship with the response, based on the false discovery rate (FDR), but note that some variables can be significant in a multivariate model even if they would not have been selected by FDR. Favourability is also calculated. By default all data are used in model training, but you can define an optional test.sample to be reserved for model testing afterwards.

The function will create a list of the resulting models (each with the name of the corresponding species column) and a data frame with their predictions (Y, P and/or F, all of which are optional). If you plan on representing these predictions in a GIS based on .dbf tables, remember that dbf only allows up to 10 characters in column names; multGLM predictions will add 2 characters (_y, _P and/or _F) to each of your species column names, so use species names/codes with up to 8 characters in the data set that you are modelling. You can create (sub)species name abbreviations with the spCodes function from the fuzzySim package.

Value

The function returns a list with the following components:

predictions	a data frame with the model predictions (if either of y, P, or Favourability are TRUE).
models	a list of the resulting modol objects.

Author(s)

A. Marcia Barbosa

References

Fielding A.H. & Bell J.F. (1997) A review of methods for the assessment of prediction errors in conservation presence/absence models. Environmental Conservation 24: 38-49

Huberty C.J. (1994) Applied Discriminant Analysis. Wiley, New York, 466 pp. Schaafsma W. & van Vark G.N. (1979) Classification and discrimination problems with applications. Part IIa. Statistica Neerlandica 33: 91-126

Real R., Barbosa A.M. & Vargas J.M. (2006) Obtaining environmental favourability functions from logistic regression. Environmental and Ecological Statistics 13: 237-245.

See Also

glm, Fav, step, modelTrim

Examples

```
data(rotif.env)

names(rotif.env)


# make models for 3 of the species in rotif.env:

mods <- multGLM(rotif.env, sp.cols = 45:47, var.cols = 5:17, id.col = 1,
step = TRUE, FDR = TRUE, trim = TRUE)

names(mods)

head(mods$predictions)

names(mods$models)

mods$models[[1]]

mods$models[["Ttetra"]]
```

---

multicol *Analyse the multicollinearity in a dataset, including VIF*

Description

This function analyses multicollinearity in a set of variables, including the R-squared, tolerance and variance inflation factor (VIF).

Usage

```
multicol(vars)
```

Arguments

vars	A matrix or data frame containing the numeric variables for which to calculate multicollinearity. Only the 'independent' (predictor, explanatory, right hand side) variables should be entered, as the result obtained for each variable depends on all the other variables present in the analysed data set.

Details

The multicol function calculates the degree of multicollinearity in a set of numeric variables, using three closely related measures: R squared (the coefficient of determination of a linear regression of each predictor variable on all other predictor variables, i.e., the amount of variation in each variable that is accounted for by other variables in the dataset); tolerance (1 - R squared), i.e. the amount of variation in each variable that is not included in the remaining variables; and the variance inflation factor: VIF = 1 / (1 - R squared), which reflects the degree to which the variance of an estimated regression coefficient (in a model using these variables as predictors) is increased due only to the correlations among covariates. (Marquardt 1970; Mansfield & Helms 1982). Such measures, especially the VIF, are widely used in ecological modelling and a range of other research areas, and several packages already calculate the VIF in R, but their results can be strikingly different (Estrada & Barbosa, submitted). The multicol function calculates multicollinearity statistics directly on the independent variables, producing the expected results and without (so far) clashing with other packages.

Value

The function returns a matrix with one row per analysed variable, the names of the variables as row names, and 3 columns: R-squared, Tolerance and VIF.

Author(s)

A. Marcia Barbosa

References

Estrada A. & Barbosa A.M. (submitted) Calculating the Variance Inflation Factor in R: cautions and recommendations.

Marquardt D.W. (1970) Generalized inverses, ridge regression, biased linear estimation, and non-linear estimation. Technometrics 12: 591-612.

Mansfield E.R. & Helms B.P. (1982) Detecting multicollinearity. The American Statistician 36: 158-160.

Examples

```
data(rotif.env)
names(rotif.env)

# calculate multicollinearity among the predictor variables:
multicol(rotif.env[ , 5:17])
```

```
# more examples uing R datasets:
multicol(trees)

# you'll get a warning and some NA results if any of the variables is not numeric:
multicol(OrchardSprays)

# so define the subset of numeric 'vars' to calculate 'multicol' for:
multicol(OrchardSprays[ , 1:3])
```

---

multModEv *Multiple model evaluation*

Description

If you have a list of models created e.g. with the multGLM function, or a data frame with presence-absence data and the corresponding predicted values for a set of species, you can use the multModEv function to get a set of evaluation measures for all models simultaneously. Note that all models must have the same sample size.

Usage

```
multModEv(obs.data = NULL, pred.data = NULL, models = NULL, Favourability = FALSE,
measures = modEvAmethods("multModEv"), thresh = "preval", standardize = FALSE,
bin.method = "quantiles", quiet = TRUE)
```

Arguments

obs.data	a data frame with observed binary data.
pred.data	a data frame with the corresponding predicted probability values, with both rows and columns in the same order as in obs.data.
models	instead of (and overriding) obs.data and pred.data, a list of model objects of class "glm" applied to the same data set.
Favourability	logical, whether to convert predicted probability to Favourability before calculating fixed-threshold measures; used only if model objects are provided (rather than obs.data + pred.data).
measures	character vector of the evaluation measures to calculate. The default is all implemented measures, which you can check by typing modEvAmethods("multModEv").
thresh	the threshold value to use for calculating threshold-based measures (the ones in modEvAmethods("threshMeasures")). Can be "preval" (the default, for species prevalence) or any number between 0 and 1.
standardize	logical, whether to standardize measures that vary between -1 and 1 to the 0-1 scale (standard01).
bin.method	the method with which to divide the data into groups or bins (for calibration or reliability measures such as HLfit); type modEvAmethods("getBins") for available options).
quiet	logical, whether to be wuiet or display messages.

Value

A data frame with the value of each evaluation measure for each model.

Author(s)

A. Marcia Barbosa

See Also

threshMeasures

Examples

```
data(rotif.env)

names(rotif.env)


# make models for 3 of the species in rotif.env:

mods <- multGLM(rotif.env, sp.cols = 18:20, var.cols = 5:17, id.col = 1, step = TRUE,
FDR = TRUE, trim = TRUE)

names(mods$predictions)


# calculate some evaluation measures for these models:

eval <- multModEv(obs.data = rotif.env[ , 18:20],
pred.data = mods$predictions[ , c("Abrigh_P", "Afissa_P", "Apriod_P")],
thresh = "preval")

head(eval)

eval2 <- multModEv(models = mods$models, thresh = 0.7,
measures = c("AUC", "CCR", "Sensitivity", "TSS"))

head(eval2)
```

OA *Overlap Analysis*

Description

This function analyses the range of values of the given environmental variables at the sites where a species has been recorded present.

Usage

```
OA(data, sp.cols, var.cols)
```

Arguments

data	a data frame with your species' occurrence data and the predictor variables.
sp.cols	index number of the column containing the occurrence data of the species to be modelled. Currently only one species can be analysed at a time.
var.cols	index numbers of the columns containing the predictor variables to be used.

Details

Overlap Analysis is one of the simplest forms of modelling species' distributions. It assesses the ranges of values of the given environmental variables at the sites where a species has been recorded present, and predicts where that species should be able to occur based on those presence data (e.g. Brito et al. 1999, Arntzen & Teixeira 2006).

OA can also be useful when extrapolating models outside their original scope (geographical area, time period or spatial resolution), as it can identify which localities are within the model's domain - i.e., within the analysed ranges of values of the variables, outside which the model may not be reliable (e.g. Barbosa et al. 2009). In this case, the response is not a species' presence, but rather the sites that have been included in the model. See also the MESS function for a comparison between modelled and extrapolation environments.

Input data for the OA function are a vector or column with ones and zeros (presences vs. absences of a species if we want to model its occurrence, or modelled vs. non-modelled sites if we want to know which non-modelled sites are within the modelled range), and a matrix or data frame with the corresponding values of the environmental variables to consider (one variable in each column, values in rows).

Value

A binary vector whith 1 where the values of all predictors lie within the ranges observed for the presence records, and 0 otherwise.

Author(s)

A. Marcia Barbosa

References

Arntzen J.W, Teixeira J. (2006) History and new developments in the mapping and modelling of the distribution of the golden-striped salamander, Chioglossa lusitanica. Zeitschrift fur Feldherpetologie, Supplement: 1-14.

Barbosa, A.M., Real, R. & Vargas, J.M. (2009) Transferability of environmental favourability models in geographic space: the case of the Iberian desman (Galemys pyrenaicus) in Portugal and Spain. Ecological Modelling 220: 747-754.

Brito J.C., Crespo E.G., Paulo O.S. (1999) Modelling wildlife distributions: Logistic Multiple Regression vs Overlap Analysis. Ecography 22: 251-260.

See Also

MESS

Examples

```
data(rotif.env)

names(rotif.env)

OA(rotif.env, sp.cols = 18, var.cols = 5:17)
```

---

optiPair — *Optimize the discrimination threshold for a pair of related model evaluation measures.*

Description

The optiPair function can optimize a model's discrimination threshold based on a pair of model evaluation measures that balance each other, such as sensitivity-specificity, omission-commission, or underprediction-overprediction (Fielding & Bell 1997; Liu et al. 2011; Barbosa et al. 2013). The function plots both measures in the given pair against all thresholds with a given interval, and calculates the optimal sum, difference and mean of the two measures.

Usage

```
optiPair(obs = NULL, pred = NULL, model = NULL,
measures = c("Sensitivity", "Specificity"), interval = 0.01,
plot = TRUE, plot.sum = FALSE, plot.diff = FALSE, ylim = NULL, ...)
```

Arguments

obs	a vector of observed presences (1) and absences (0) or another binary response variable.
pred	a vector with the corresponding predicted values of presence probability, habitat suitability, environmental favourability or alike.
model	instead of (and overriding) obs and pred, a model object of class "glm".
measures	a character vector of length 2 indicating the pair of measures whose curves to plot and whose thresholds to optimize. The default is c("Sensitivity", "Specificity").
interval	the interval of thresholds at which to calculate the measures. The default is 0.01.
plot	logical indicating whether or not to plot the pair of measures.
plot.sum	logical, whether to plot the sum (+) of both measures in the pair. Defaults to FALSE.
plot.diff	logical, whether to plot the difference (-) between both measures in the pair. Defaults to FALSE.

ylim	a character vector of length 2 indicating the lower and upper limits for the y axis. The default is NULL for an automatic definition of ylim based on the values of the measures and their sum and/or difference if any of these are set to TRUE.
...	additional arguments to be passed to the plot function.

Value

The output is a list with the following components:

measures.values	a data frame with the values of the chosen pair of measures, as well as their difference, sum and mean, at each threshold.
MinDiff	numeric value, the minimum difference between both measures.
ThreshDiff	numeric value, the threshold that minimizes the difference between both measures.
MaxSum	numeric value, the maximum sum of both measures.
ThreshSum	numeric value, the threshold that maximizes the sum of both measures.
MaxMean	numeric value, the maximum mean of both measures.
ThreshMean	numeric value, the threshold that maximizes the mean of both measures.

Author(s)

A. Marcia Barbosa

References

Barbosa, A.M., Real, R., Munoz, A.-R. & Brown, J.A. (2013) New measures for assessing model equilibrium and prediction mismatch in species distribution models. Diversity and Distributions 19: 1333-1338

Fielding A.H. & Bell J.F. (1997) A review of methods for the assessment of prediction errors in conservation presence/absence models. Environmental Conservation 24: 38-49

Liu C., White M., & Newell G. (2011) Measuring and comparing the accuracy of species distribution models with presence-absence data. Ecography, 34, 232-243.

See Also

optiThresh, threshMeasures

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))

optiPair(model = mod)
```

```
optiPair(model = mod, measures = c("UPR", "OPR"))


# you can also use optiPair with vectors of observed and predicted values
# instead of with a model object:

prob <- predict(mod, data = rotif.env, type = "response")

optiPair(obs = rotif.env$Abrigh, pred = prob, measures = c("Omission",
"Commission"))
```

---

optiThresh *Optimize threshold for model evaluation.*

Description

The optiThresh function calculates optimal thresholds for a number of model evaluation measures (see threshMeasures). Optimization is given for each measure, and/or for all measures according to particular criteria (e.g. Jimenez-Valverde & Lobo 2007, Nenzen & Araujo 2011). Results are given numerically and in plots.

Usage

```
optiThresh(obs = NULL, pred = NULL, model = NULL, interval = 0.01,
measures = modEvAmethods("threshMeasures"), optimize = modEvAmethods("optiThresh"),
simplif = FALSE, plot = TRUE, sep.plots = FALSE, xlab = "Threshold", ...)
```

Arguments

obs	a vector of observed presences (1) and absences (0) or another binary response variable.
pred	a vector with the corresponding predicted values of presence probability, habitat suitability, environmental favourability or alike.
model	instead of (and overriding) obs and pred, a model object of class "glm".
interval	numeric value between 0 and 1 indicating the interval between the thresholds at which to calculate the evaluation measures. Defaults to 0.01.
measures	character vector indicating the names of the model evaluation measures for which to calculate optimal thresholds. The default is using all measures available in modEvAmethods("threshMeasures").
optimize	character vector indicating the threshold optimization criteria to use; "each" calculates the optimal threshold for each model evaluation measure, while the remaining options optimize all measures according to the specified criterion. The default is using all criteria available in modEvAmethods("optiThresh").
simplif	logical, whether to calculate a faster simplified version. Used internally in other functions.
plot	logical, whether to plot the values of each evaluation measure at all thresholds.

sep.plots	logical. If TRUE, each plot is presented separately (you need to be recording R plot history to be able to browse through them all); if FALSE (the default), all plots are presented together in the same plotting window.
xlab	character vector indicating the label of the x axis.
...	additional arguments to pass to plot.

Value

This function returns a list with the following components:

all.thresholds	a data frame with the values of all analysed measures at all analysed thresholds.
optimals.each	if "each" is among the threshold criteria specified in optimize, optimals.each is output as a data frame with the value of each measure at its optimal threshold, as well as the type of optimal for that measure (which may be the maximum for measures of goodness such as "Sensitivity", or the minimum for measures of badness such as "Omission").
optimals.criteria	a data frame with the values of measure at the threshold that maximizes each of the criteria specified in optimize (except for "each", see above).

Note

Some measures cannot be calculated for thresholds at which there are zeros in the confusion matrix, hence the eventual 'NaN' or 'Inf' in results. Also, optimization may be deceiving for some measures; use plot = TRUE and inspect the plot(s).

Author(s)

A. Marcia Barbosa

References

Jimenez-Valverde A. & Lobo J.M. (2007) Threshold criteria for conversion of probability of species presence to either-or presence-absence. Acta Oecologica 31: 361-369.

Nenzen H.K. & Araujo M.B. (2011) Choice of threshold alters projections of species range shifts under climate change. Ecological Modelling 222: 3346-3354.

See Also

threshMeasures, optiPair

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))
```

```
optiThresh(model = mod)


# change some of the parameters:

optiThresh(model = mod, pch = 20, measures = c("CCR", "Sensitivity", "Specificity",
"UPR", "OPR", "kappa", "TSS"), ylim = c(0, 1))


# you can also use optiThresh with vectors of observed and predicted values
# instead of with a model object:

prob <- predict(mod, data = rotif.env, type = "response")

optiThresh(obs = rotif.env$Abrigh, pred = prob, pch = ".")
```

---

percentTestData *Percent test data*

Description

Based on the work of Schaafsma & van Vark (1979), Huberty (1994) provided a heuristic ("rule of thumb") for determining an adequate proportion of data to set aside for testing species presence/absence models, based on the number of predictor variables that are used (Fielding & Bell 1997). The percentTestData function calculates this proportion as a percentage.

Usage

```
percentTestData(nvar)
```

Arguments

nvar	the number of variables in the model.

Value

A numeric value of the percentage of data to leave out of the model for further model testing.

Author(s)

A. Marcia Barbosa

References

Huberty C.J. (1994) Applied Discriminant Analysis. Wiley, New York, 466 pp. Schaafsma W. & van Vark G.N. (1979) Classification and discrimination problems with applications. Part IIa. Statistica Neerlandica 33: 91-126

Fielding A. H. & Bell J. F. (1997) A review of methods for the assessment of prediction errors in conservation presence/absence models. Environmental Conservation 24: 38-49

See Also

multGLM

Examples

```
# say you're building a model with 15 variables:

percentTestData(15)

# the result tells you that 21% is an appropriate percentage of data
# to set aside for testing your model, so train it with 79% of the data
```

---

plotGLM *Plot a generalized linear model*

Description

This function plots the observed (presence/absence) data and the predicted (probability) values of a Generalized Linear Model against the y regression equation (logit) values. Only logistic regression (binomial response, logit link) is implemented. Optionally (if plot.values = TRUE) it can display in the plot the proportion of deviance explained (Weisberg 1980, Guisan & Zimmermann 2000).

Usage

```
plotGLM(obs = NULL, pred = NULL, model = NULL, link = "logit", plot.values = FALSE,
xlab = "Logit (Y)", ylab = "Predicted probability", main = "Model plot", ...)
```

Arguments

obs	a vector of presence/absence or other binary (1-0) observed data.
pred	a vector of the values predicted by a GLM of the binary observed data.
model	instead of (and overriding) obs and pred, a model object of class "glm".
link	the link function of the GLM; only 'logit' (the default) is implemented.
plot.values	logical, whether to report values associated with the model in the plot (see Details). Defaults to FALSE.
xlab	character string specifying the label for the x axis.
ylab	character string specifying the label for the y axis.
main	character string specifying the title for the plot.
...	additional arguments to pass to plot.

Details

If plot.values = TRUE, the plot will also display values such as deviance (and adjusted deviance if model is provided). Deviance is calculated with the Dsquared function. Additional measures will be implemented in the future.

Value

This function outputs a plot of the provided model.

Author(s)

A. Marcia Barbosa

References

Guisan A. & Zimmermann N.E. (2000) Predictive habitat distribution models in ecology. Ecological Modelling 135: 147-186

Weisberg S. (1980) Applied Linear Regression. Wiley, New York

See Also

glm, Dsquared

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))

plotGLM(model = mod)

plotGLM(model = mod, plot.values = TRUE)


# you can also use \code{plotGLM} with vectors of observed and predicted values
# instead of with a model object:

prob <- predict(object = mod, data = rotif.env, type = "response")

plotGLM(obs = rotif.env$Abrigh, pred = prob)
```

---

prevalence *Prevalence*

Description

For building and evaluating species distribution models, the porportion of presences of the species may be issues to take into account (e.g. Jimenez-Valverde & Lobo 2006, Barbosa et al. 2013). The prevalence function calculates this measure.

Usage

```
prevalence(obs, event = 1)
```

Arguments

obs	a vector of binary observations (e.g. 1 vs. 0, male vs. female, disease vs. no disease, etc.).
event	the value whose prevalence we want to calculate (e.g. 1, "present", etc.).

Value

Numeric value of the prevalence of event in the obs vector.

Author(s)

A. Marcia Barbosa

References

Barbosa A.M., Real R., Munoz A.R. & Brown J.A. (2013) New measures for assessing model equilibrium and prediction mismatch in species distribution models. Diversity and Distributions, in press

Jimenez-Valverde A. & Lobo J.M. (2006) The ghost of unbalanced species distribution data in geographical model predictions. Diversity and Distributions, 12: 521-524.

See Also

evenness

Examples

```
(x <- rep(c(0, 1), each = 5))

(y <- c(rep(0, 3), rep(1, 7)))

(z <- c(rep(0, 7), rep(1, 3)))

prevalence(x)

prevalence(y)

prevalence(z)
```

range01 *Shrink or stretch a vector to make it range between 0 and 1*

Description

This function re-scales a numeric vector so that it ranges between 0 and 1 (i.e., the lowest value becomes 0, the highest becomes 1, and the ones in the middle retain their rank and relative diference).

Usage

```
range01(x, na.rm = TRUE)
```

Arguments

x	a numeric vector.
na.rm	logical, whether to remove NA values.

Details

This function was borrowed from http://stackoverflow.com/questions/5468280/scale-a-series-between-two-points-in-r/5468527#5468527 and adapted to handle also missing values.

Value

A numeric vector of the same length as the input, now with the values ranging from 0 to 1.

Author(s)

A. Marcia Barbosa

See Also

standard01

Examples

```
range01(0:10)

range01(-12.3 : 21.7)
```

rotif.env *Rotifers and environmental variables on TDWG level 4 regions of the world*

Description

These data were extracted from a database of monogonont rotifer species presence records on the geographical units used by the Biodiversity Information Standards (formerly Taxonomic Database Working Group, TDWG; base maps available at http://www.kew.org/science-research-data/kew-in-depth/gis/ resources-and-publications/data/tdwg/index.htm) and a few environmental (including human and spatial) variables on the same spatial units. The original data were compiled and published by Fontaneto et al. (2012) in long (narrow, stacked) format. Here they are presented in wide or unstacked format (presence -absence table, obtained with the splist2presabs function in the **fuzzySim** package), reduced to the species recorded in at least 100 different TDWG level 4 units, and with abbreviations of the species' names (obtained with the spCodes function in the **fuzzySim** package). Mind that this is not a complete picture of these species' distributions, due to insufficient sampling in many regions.

Usage

```
data(rotif.env)
```

Format

A data frame with 291 observations on the following 47 variables.

TDWG4 a factor with levels ABT-OO AFG-OO AGE-BA AGE-CH AGE-CN AGE-ER AGE-SF AGS-CB AGS-NE AGS-RN AGS-SC AGS-TF AGW-CA AGW-JU AGW-LR AGW-ME AGW-SA AGW-SE AGW-SJ AGW-SL AGW-TU ALG-OO ARI-OO ARK-OO ARU-OO ASK-OO ASS-AS ASS-MA ASS-ME ASS-MI ASS-NA ASS-TR ATP-OO AUT-AU AZO-OO BAH-OO BGM-BE BLR-OO BLT-ES BLT-KA BLT-LI BOL-OO BOR-KA BRC-OO BRY-OO BUL-OO BZC-MS BZC-MT BZE-MA BZE-PE BZL-ES BZL-MG BZL-RJ BZL-SP BZN-AM BZN-PA BZN-RM BZN-RO BZS-PR CAL-OO CAY-OO CBD-OO CHA-OO CHC-YN CHM-HJ CHN-SD CHQ-OO CHS-HE CHS-HN CLC-BI CLC-LA CLC-MA CLC-OH CLM-OO CLS-LL CMN-OO CNT-OO COL-OO COM-CO COS-OO CPP-EC CPP-NC CPP-WC CPV-OO CTM-OO CUB-OO CZE-CZ CZE-SL DEN-OO DOM-OO ECU-OO EGY-OO EHM-AP ELS-OO ETH-OO FIN-OO FLA-OO FRA-FR GAM-OO GEO-OO GER-OO GHA-OO GNL-OO GRB-OO GRC-OO GUA-OO GUI-OO GUY-OO HAW-HI HON-OO HUN-OO ICE-OO IDA-OO ILL-OO IND-AP IND-DE IND-GU IND-HA IND-JK IND-KE IND-KT IND-MP IND-MR IND-OR IND-PU IND-RA IND-TN IND-UP IND-WB INI-OO IOW-OO IRE-IR IRK-OO IRN-OO ITA-IT IVO-OO JAM-OO JAP-HK JAP-HN JAP-KY JAW-OO JNF-OO KAM-OO KAZ-OO KEG-OO KEN-OO KOR-SK KRA-OO KTY-OO LAB-OO LAO-OO LBS-SY LEE-AB LEE-AG LEE-GU LEE-MO LEE-NL LEE-SM LEE-VI LOU-OO LSI-BA LSI-LS MAI-OO MAS-OO MDG-OO MIC-OO MIN-OO MLI-OO MLW-OO MLY-PM MLY-SI MON-OO MOR-MO MOZ-OO MRY-OO MSO-OO MTN-OO MXC-DF MXC-ME MXE-GU MYA-OO NAT-OO NBR-OO NCA-OO NCS-DA NCS-KC NCS-KR NDA-OO NEB-OO NEP-OO

NET-OO NEV-OO NFL-NE NGA-OO NIC-OO NLA-BO NLA-CU NNS-OO NOR-OO NSC-OO NSW-NS NTA-OO NUN-OO NWG-PN NWH-OO NWJ-OO NWT-OO NWY-OO NZN-OO NZS-OO OFS-OO OHI-OO OMA-OO ONT-OO ORE-OO PAL-IS PAN-OO PAR-OO PEN-OO PER-OO PHI-OO POL-OO POR-OO PUE-OO QLD-QU QUE-OO REU-OO ROM-OO RUC-OO RUE-OO RUN-OO RUS-OO RUW-OO RWA-OO SAS-OO SCA-OO SEN-OO SEY-OO SIE-OO SOA-OO SPA-AN SPA-SP SRL-OO SUD-OO SUM-OO SUR-OO SWE-OO SWI-OO TAI-OO TAN-OO TAS-OO TCS-AR TCS-GR TEN-OO TEX-OO THA-OO TRT-OO TUE-OO TUN-OO TUR-OO TVL-GA TVL-MP TVL-NP TVL-NW UGA-OO UKR-UK URU-OO UZB-OO VEN-OO VER-OO VIC-OO VNA-OO VRG-OO WAS-OO WAU-WA WHM-JK WIN-BA WIN-DO WIN-GR WIN-MA WIS-OO WSB-OO YAK-OO YUG-CR YUG-MA YUG-SE YUG-SL YUK-OO ZAI-OO ZAM-OO ZIM-OO

LEVEL_NAME a factor with levels

REGION_NAME a factor with levels Arabian_Peninsula Australia Brazil Caribbean Caucasus Central_America China Eastern_Asia Eastern_Canada Eastern_Europe East_Tropical_Africa Indian_Subcontinent Indo-China Macaronesia Malesia Mexico Middle_Asia Middle_Europe Mongolia New_Zealand North-Central_Pacific North-Central_U.S.A. Northeastern_U.S.A. Northeast_Tropical_Africa Northern_Africa Northern_Europe Northern_South_America Northwestern_U.S.A. Papuasia Russian_Far_East Siberia South-Central_U.S.A. Southeastern_Europe Southeastern_U.S.A. Southern_Africa Southern_South_America South_Tropical_Africa Southwestern_Europe Southwestern_U.S.A. Subantarctic_Islands Subarctic_America West-Central_Tropical_Africa Western_Asia Western_Canada Western_Indian_Ocean Western_South_America West_Tropical_Africa

CONTINENT a factor with levels AFRICA ANTARCTIC ASIA-TEMPERATE ASIA-TROPICAL AUSTRALASIA EUROPE NORTHERN_AMERICA PACIFIC SOUTHERN_AMERICA

Area a numeric vector

Altitude a numeric vector

AltitudeRange a numeric vector

HabitatDiversity a numeric vector

HumanPopulation a numeric vector

Latitude a numeric vector

Longitude a numeric vector

Precipitation a numeric vector

PrecipitationSeasonality a numeric vector

TemperatureAnnualRange a numeric vector

Temperature a numeric vector

TemperatureSeasonality a numeric vector

UrbanArea a numeric vector

Abrigh a numeric vector

Afissa a numeric vector

Apriod a numeric vector

Bangul a numeric vector

Bcalyc a numeric vector

Bplica a numeric vector

Bquadr a numeric vector

Burceo a numeric vector

Cgibba a numeric vector

Edilat a numeric vector

Flongi a numeric vector

Kcochl a numeric vector

Kquadr a numeric vector

Ktropi a numeric vector

Lbulla a numeric vector

Lclost a numeric vector

Lhamat a numeric vector

Lluna a numeric vector

Llunar a numeric vector

Lovali a numeric vector

Lpatel a numeric vector

Lquadr a numeric vector

Mventr a numeric vector

Ppatul a numeric vector

Pquadr a numeric vector

Pvulga a numeric vector

Specti a numeric vector

Tpatin a numeric vector

Tsimil a numeric vector

Ttetra a numeric vector

Source

Fontaneto D., Barbosa A.M., Segers H. & Pautasso M. (2012) The 'rotiferologist' effect and other global correlates of species richness in monogonont rotifers. Ecography, 35: 174-182.

See Also

fuzzySim

Examples

```
data(rotif.env)

head(rotif.env)
```

RsqGLM *R-squared measures for GLMs*

Description

This function calculates some (pseudo) R-squared statistics for binomial Generalized Linear Models.

Usage

RsqGLM(obs = NULL, pred = NULL, model = NULL)

Arguments

obs	a vector of observed presences (1) and absences (0) or another binary response variable.
pred	a vector with the corresponding predicted values of presence probability. Must be of the same length and in the same order as obs.
model	instead of (and overriding) obs and pred, you can provide a model object of class "glm" and family "binomial".

Details

Implemented measures include the R-squareds of McFadden (1974), Cox-Snell (1989), Nagelkerke (1991, which corresponds to the corrected Cox-Snell, eliminating its upper bound), and Tjur (2009). See Allison (2014) for a brief review of these measures.

Value

The function returns a named list of the calculated R-squared values.

Author(s)

A. Marcia Barbosa

References

Allison P. (2014) Measures of fit for logistic regression. SAS Global Forum, Paper 1485-2014

Cox, D.R. & Snell E.J. (1989) The Analysis of Binary Data, 2nd ed. Chapman and Hall, London

McFadden, D. (1974) Conditional logit analysis of qualitative choice behavior. In: Zarembka P. (ed.) Frontiers in Economics. Academic Press, New York

Nagelkerke, N.J.D. (1991) A note on a general definition of the coefficient of determination. Biometrika, 78: 691-692

Tjur T. (2009) Coefficients of determination in logistic regression models - a new proposal: the coefficient of discrimination. The American Statistician, 63: 366-372.

See Also

AUC, threshMeasures, HLfit

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))

RsqGLM(model = mod)


# you can also use RsqGLM with vectors of observed and predicted values
# instead of a model object:

prob <- predict(mod, data = rotif.env, type = "response")

RsqGLM(obs = rotif.env$Abrigh, pred = prob)
```

---

standard01 *Standardize to 0-1 (or vice-versa)*

Description

This function converts the score of a measure that ranges from -1 to 1 (e.g. a kappa or TSS value obtained for a model) into its (linearly) corresponding value in 0-to-1 scale, so that it can be compared directly with measures that range between 0 and 1 (such as CCR or AUC). It can also perform the conversion in the opposite direction.

Usage

```
standard01(score, direction = c("-1+1to01", "01to-1+1"))
```

Arguments

score	numeric value indicating the score of the measure of interest.
direction	character value indicating the direction in which to perform the standardization. The default, "-1+1to01", can be switched to "01to-1+1".

Details

While most of the threshold-based measures of model evaluation range theoretically from 0 to 1, some of them (such as Cohen's kappa and the true skill statistic, TSS) may range from -1 to 1 (Allouche et al. 2006). Thus, the values of different measures may not be directly comparable. We do not usually get negative values of TSS or kappa (nor values under 0.5 for CCR or AUC, for

example) because that only happens when model predictions perform worse than random guesses; still, such values are mathematically possible, and can occur e.g. when extrapolating models to regions where where the species-environment relationships differ. This standardization is included as an option in the threshMeasures function.

Value

The numeric value of score when re-scaled to the 0-to-1 (or to the -1 to +1) scale.

Note

Note that this is not the same as re-scaling a vector so that it ranges between 0 and 1, which is done by range01.

Author(s)

A. Marcia Barbosa

References

Allouche O., Tsoar A. & Kadmon R. (2006) Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). Journal of Applied Ecology 43: 1223-1232

See Also

threshMeasures, range01

Examples

```
standard01(0.6)

standard01(0.6, direction = "-1+1to01")

standard01(0.6, direction = "01to-1+1")
```

---

threshMeasures *Threshold-based measures of model evaluation*

Description

This function calculates a number of measures for evaluating the discrimination capacity of a species distribution (or ecological niche, or bioclimatic envelope...) model against observed presence-absence data (Fielding & Bell 1997; Liu et al. 2011; Barbosa et al. 2013).

Usage

```
threshMeasures(obs = NULL, pred = NULL, model = NULL, thresh = 0.5,
measures = modEvAmethods("threshMeasures"), simplif = FALSE, plot = TRUE,
plot.ordered = FALSE, standardize = TRUE, messages = TRUE, ...)
```

Arguments

obs	a vector of observed presences (1) and absences (0) or another binary response variable.
pred	a vector with the corresponding predicted values of presence probability, habitat suitability, environmental favourability or alike.
model	instead of (and overriding) obs and pred, you can provide a model object of class "glm".
thresh	numeric value of the threshold to separate predicted presences from predicted absences in pred; can be "preval" (to use the prevalence of obs as the threshold) or any real number between 0 and 1. The default is 0.5 (but see Details).
measures	character vector of the evaluation measures to use. By default, all measures available in modEvAmethods("threshMeasures") are calculated.
simplif	logical, whether to calculate a faster, simplified version. Used internally by other functions in the package. Defaults to FALSE.
plot	logical, whether to produce a barplot of the calculated measures. Defaults to TRUE.
plot.ordered	logical, whether to plot the measures in decreasing order rather than in input order. Defaults to FALSE.
standardize	logical, whether to change measures that may range between -1 and +1 (namely kappa and TSS) to their corresponding value in the 0-to-1 scale (skappa and sTSS), so that they can compare directly to other measures (see standard01). The default is TRUE but a message is displayed to let the user know about it.
messages	logical, whether to display messages.
...	additional arguments to be passed to plot.

Details

This function can also be used to calculate the agreement between different presence-absence (or other types of binary) data, as Barbosa et al. (2012) did for comparing mammal distribution data from atlas and range maps. Notice, however, that some of these measures, such as TSS or NMI, are not symmetrical (obs vs. pred is different from pred vs. obs).

Value

If simplif = TRUE, the output is a numeric matrix with the name and value of each measure. If simplif = FALSE (the default), the ouptut is a bar plot of the calculated measures and a list with the following components:

N	the number of observations (records) in the analysis.
Prevalence	the prevalence (proportion of presences) in obs.
Threshold	the threshold value used to calculate the measures.
ConfusionMatrix	the confusion matrix obtained with the used threshold.
ThreshMeasures	a numeric matrix with the name and value of each measure.

Note

Some of these measures (like NMI, UPR, OPR, PPP, NPP) cannot be calculated for thresholds at which there are zeros in the confusion matrix.

By default the threshold used is 0.5, but you may want to change this depending on a number of criteria (see e.g. Jimenez-Valverde & Lobo 2007, Nenzen & Araujo 2011). You can set thresh to "preval" (species' prevalence or proportion of presences in the input obs data) or calculate optimal threshold values according to several criteria with the optiThresh function.

While most of these threshold-based measures range from 0 to 1, some of them - such as kappa and TSS - may range from -1 to 1 (Allouche et al. 2006), so their raw scores are not exactly comparable. threshMeasures includes an option to standardize these measures to 0-1, for which you need the standard01 function, so that you obtain the standardized versions skappa and sTSS.

Author(s)

A. Marcia Barbosa

References

Allouche O., Tsoar A. & Kadmon R. (2006) Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). Journal of Applied Ecology 43: 1223-1232.

Barbosa A.M., Estrada A., Marquez A.L., Purvis A. & Orme C.D.L. (2012) Atlas versus range maps: robustness of chorological relationships to distribution data types in European mammals. Journal of Biogeography 39: 1391-1400

Barbosa A.M., Real R., Munoz A.R. & Brown J.A. (2013) New measures for assessing model equilibrium and prediction mismatch in species distribution models. Diversity and Distributions 19: 1333-1338

Fielding A.H. & Bell J.F. (1997) A review of methods for the assessment of prediction errors in conservation presence/absence models. Environmental Conservation 24: 38-49

Jimenez-Valverde A. & Lobo J.M. (2007) Threshold criteria for conversion of probability of species presence to either-or presence-absence. Acta Oecologica 31: 361-369

Liu C., White M., & Newell G. (2011) Measuring and comparing the accuracy of species distribution models with presence-absence data. Ecography 34: 232-243.

Nenzen H.K. & Araujo M.B. (2011) Choice of threshold alters projections of species range shifts under climate change. Ecological Modelling 222: 3346-3354

See Also

optiThresh, optiPair, AUC, HLfit

Examples

```
data(rotif.env)

names(rotif.env)

mod <- with(rotif.env, glm(Abrigh ~ Area + Altitude + AltitudeRange +
HabitatDiversity + HumanPopulation, family = binomial))
```

```
threshMeasures(model = mod, simplif = TRUE)

threshMeasures(model = mod)

threshMeasures(model = mod, plot.ordered = TRUE)

threshMeasures(model = mod, measures = c("CCR", "TSS", "kappa"))

threshMeasures(model = mod, plot.ordered = TRUE, thresh = "preval")


# you can also use threshMeasures with vectors of observed and predicted values
# instead of with a model object:

prob <- predict(mod, data = rotif.env, type = "response")

threshMeasures(obs = rotif.env$Abrigh, pred = prob)
```

---

timer *Timer*

Description

Reporting of time elapsed since a given start time. This function is used internally by other functions in the **modEvA** package.

Usage

```
timer(start.time)
```

Arguments

start.time	A date-time object of class POSIXct, e.g. as given by Sys.time.

Value

The function returns a message informing of the time elapsed since start.time.

Author(s)

A. Marcia Barbosa

See Also

Sys.time, proc.time, difftime

Examples

```
# get starting time:
start <- Sys.time()

# do some random analysis:
sapply(rnorm(50000), function(x) x*5)

# see how long it took:
timer(start)
```

---

varPart *Variation partitioning*

Description

This function performs variation partitioning (Borcard et al. 1992) among two factors (e.g. Ribas et al. 2006) or three factors (e.g. Real et al. 2003) for either multiple linear regression models (LM) or generalized linear models (GLM).

Usage

```
varPart(A, B, C = NULL, AB, BC = NULL, AC = NULL, ABC = NULL, model.type,
A.name = "A", B.name = "B", C.name = "C", plot = TRUE, plot.digits = 3,
cex.names = 1.2, cex.values = 1)
```

Arguments

A	numeric value of the R-squared of the regression of the resonse variable on the variables related to factor 'A'
B	numeric value of the R-squared of the regression of the resonse variable on the variables related to factor 'B'
C	(optionally, if there are 3 factors) numeric value of the R-squared of the regression of the response on the variables related to factor 'C'
AB	numeric value of the R-squared of the regression of the response on the variables of factors 'A' and 'B' simultaneously
BC	(if there are 3 factors) numeric value of the R-squared of the regression of the response on the variables of factors 'B' and 'C' simultaneously
AC	(if there are 3 factors) numeric value of the R-squared of the regression of the response on the variables of factors 'A' and 'C' simultaneously
ABC	(if there are 3 factors) numeric value of the R-squared of the regression of the response on the variables of factors 'A', 'B' and 'C' simultaneously
model.type	character value, either "LM" (linear model) or "GLM" (generalised linear model) depending on your model type.
A.name	optional character indicating the name of factor 'A'

B.name	optional character indicating the name of factor 'B'
C.name	optional character indicating the name of factor 'C' (if there are 3 factors)
plot	logical, whether to plot the variation partitionig diagram.
plot.digits	integer value of the number of digits to which to round the values in the plot.
cex.names	numeric value indicating character expansion factor to define the size of the names of the factors displayed in the plot.
cex.values	numeric value indicating character expansion factor to define the size of the values displayed in the plot.

Details

If you have linear models, input data for varPart are the coefficients of determination (R-squared values) of the linear regressions of the target variable on all the variables in the model, on the variables related to each particular factor, and (when there are 3 factors) on the variables related to each pair of factors. The outputs are the amounts of variance explained exclusively by each factor, the amounts explained exclusively by the overlapping effects of each pair of factors, and the amount explained by the overlap of the 3 factors if this is the case (e.g. Real et al. 2003). The amount of variation not explained by the complete model is also provided.

If you have generalized linear models (GLMs) such as logistic regression (see glm) or the favourability function (see Fav), you have no R-squared values from these models; inputs are then the squared coefficients of (Spearman's) correlation between the model predictions given by each factor or pair of factors and the predictions of the complete model (e.g. Munoz & Real 2006), or the R-squared values of the corresponding logit (y) functions (Real et al. 2013), or an adjusted R-squared (De Araujo et al. 2013). Consequently, output values are not the total amounts of variance (of the target variable) explained by factors and overlaps, but rather their proportional contribution to the total variation explained by the model. The amount of unexplained variation is not available for GLMs.

Value

The output consists of a data frame indicating the proportion of variance accounted for by each of the factors, and (if plot = TRUE a Venn diagram of the contributions of each factor.)

Author(s)

A. Marcia Barbosa

References

Borcard D, Legendre P, Drapeau P (1992) Partialling out the spatial component of ecological variation. Ecology 73: 1045-1055

De Araujo, C.B., Marcondes-Machado, L.O. & Costa, G.C. (2013) The importance of biotic interactions in species distribution models: a test of the Eltonian noise hypothesis using parrots. Journal of Biogeography, early view (DOI: 10.1111/jbi.12234)

Munoz, A.-R. & Real, R. (2006) Assessing the potential range expansion of the exotic monk parakeet in Spain. Diversity and Distributions 12: 656-665.

Real, R., Barbosa, A.M., Porras, D., Kin, M.S., Marquez, A.L., Guerrero, J.C., Palomo, L.J., Justo, E.R. & Vargas, J.M. (2003) Relative importance of environment, human activity and spatial situation in determining the distribution of terrestrial mammal diversity in Argentina. Journal of Biogeography 30: 939-947.

Real R., Romero D., Olivero J., Estrada A. & Marquez A.L. (2013) Estimating how inflated or obscured effects of climate affect forecasted species distribution. PLoS ONE 8: e53646. doi:10.1371/journal.pone.0053646

Ribas, A., Barbosa, A.M., Casanova, J.C., Real, R., Feliu, C. & Vargas, J.M. (2006) Geographical patterns of the species richness of helminth parasites of moles (Talpa spp.) in Spain: separating the effect of sampling effort from those of other conditioning factors. Vie et Milieu 56: 1-8.

Examples

```
varPart(A = 0.857537, B = 0.078095, C = 0.499474, AB = 0.89933, BC = 0.492119,
AC = 0.981672, model.type = "GLM", A.name = "Climatic", B.name = "Human",
C.name = "Topographic", plot.digits = 3, cex.names = 1.2, cex.values = 1)
```

Index