

Package ‘grasp’

April 17, 2009

Version 2.5-7

Date 2008-10-10

Title Generalized Regression Analysis and Spatial Prediction for R

Author Anthony Lehmann (anthony.lehmann@unige.ch), Fabien Fivaz (fabien.fivaz@unine.ch), John Leathwick and Jake Overton, with contributions from many specialists from around the world

Description GRASP is a general method for making spatial predictions of response variables (RV) using point surveys of the RV and spatial coverages of Predictor variables (PV). Originally, GRASP was developed to analyse, model and predict vegetation distribution over New Zealand. It has been used in all sorts of applications since then. (A. Lehmann, J.R. Leathwick & J.McC. Overton, 2002. GRASP. Ecological Modelling, 157: 189-207)

Maintainer Fabien Fivaz <fabien.fivaz@unine.ch>

Depends R (>= 2.7), gam, mda

License GPL-2

URL <http://www.r-project.org>, <http://www.unige.ch/ia/climate/grasp>

Repository CRAN

Date/Publication 2008-10-12 09:02:12

R topics documented:

grasp	2
grasp.adm.dump	6
grasp.data.read	6
grasp.exp.cormat	7
grasp.exp.datamap	8
grasp.exp.distry	8
grasp.exp.histo	9
grasp.exp.RvsP	10
grasp.exp.summary	10
grasp.exp.xpredplot	11

grasp.in	12
grasp.int.contplot	13
grasp.int.contrib	14
grasp.int.plot	14
grasp.int.response	15
grasp.mod.anova	16
grasp.mod.validate	17
grasp.msk.limits	17
grasp.options	18
grasp.pred	19
grasp.pred.export	20
grasp.pred.lookup	21
grasp.pred.plot	21
grasp.step	22
grasp.step.aic	23
grasp.step.anova	23
grasp.step.bic	24
grasp.step.bruto	25
grasp.step.contrib	26
grasp.step.cross	26
grasp.step.crossplot	27
grasp.step.crossval	28
grasp.step.full	28
grasp.step.rocplot	29
grasp.step.scope	29
grasp.step.start	30
XXX	30
XXXpred	31
YYY	32

Index	33
--------------	-----------

grasp

Generalized Regression Analysis and Spatial Prediction

Description

GRASP is a general method for making spatial predictions of response variables (RV) using point surveys of the RV and spatial coverages of Predictor variables (PV). GRASP uses Trevor Hastie's gam library.

Usage

```
grasp(selected.responses, selected.predictors, title = "GRASP: ", path = "", save.o
gr.fam = "binomial", weights = FALSE, set.limits = FALSE, use.limits = FALSE, reset
make.summary = FALSE, plot.maps = FALSE, plot.distry = FALSE, plot.histograms = FAI
plot.correlation = FALSE, corlim = 100, use.correlation = FALSE, reset.cor = TRUE,
test = "F", plimit = 0.05, direction = "both", df1 = 0, df2 = 4, startwith = gr.sel
```

```

contributions = FALSE, plot.contributions = FALSE, plot.models = FALSE, response.cu
predictions = FALSE, export.predictions = FALSE, plot.predictions = FALSE, lookup.t
plotpar = c(3, 3), prescol= "black", abscol="grey", nbar = 10, show.details = FALSE

```

Arguments

```

selected.responses      Vector of column numbers or column names of YYY table, which contains re-
                        sponse variables
selected.predictors     Vector of column numbers or column names of XXX table, which contains pre-
                        dictor variables
title                  Title used in GRASP outputs
path                  Working directory path
save.outputs          Send outputs for each response to a specific log file in your working directory
save.PNG              Save graphs as PNG files in your working directory
gr.fam                Choice of statistical distribution for Ys: "binomial", "poisson", "quasibinomial",
                        "quasipoisson", "gaussian"
weights              Recalculate weights of 0s so that their sum is equal to sum of 1s
set.limits            Calculate and save limits of the environmental envelop of each responses (Ys)
use.limits            Use the saved limits in anylyses
reset.lim             Reset saved limits
selxlim              Vector of column numbers or column names of XXX table used to define limits
npast                 Number of absences (0s) kept after the last presence (1) along each selected
                        predictors used for calculating limits
make.summary          Calculate the data summary of selected Ys and Xs
plot.maps             Plot the spatial distribution of Ys
plot.distry           Histograms of the distribution of Ys
plot.histograms      Plot the distribution of Ys on histograms of Xs
plot.respvspred       Plot the relationship between Ys and Xs
plot.xpred            maps the predictors contained in XXXpred
plot.correlation      Calculate the correlations between Xs and eliminate Xs from candidate predic-
                        tors if maximum correlation has been set with corlim
corlim                Maximum correlation allowed between predictors, if higher correlation found
                        predictors are withdrawn from Xs candidate list. If correlated, the last selected
                        Xs is removed
use.correlation       Select a model using only the uncorrelated predictors selected earlier
reset.cor             resetting candidate predictors (Xs) to all selected Xs

```

<code>full.models</code>	Select a model for each Ys forcing all candidate Xs in the models
<code>stepwise.models</code>	Select a model for each Ys from candidate Xs using the different possible options below
<code>test</code>	Choice of criterias (AIC, BIC) or tests (CHI and F (for quasi models))
<code>plimit</code>	Probability limits for CHI and F tests
<code>direction</code>	Set the direction of selection, default is both
<code>df1</code>	Optional smoother degree of freedom to be tested in variable selection
<code>df2</code>	First smoother degree of freedom to be tested in variable selection
<code>startwith</code>	Vector of column numbers of predictors used to start stepwise procedure
<code>resetstart</code>	Resetting starting predictors (Xs) to all selected Xs
<code>contributions</code>	Calculate contributions of univariate models (alone), when dropping each predictor from the selected model (drop) and within the model (model)
<code>plot.contributions</code>	Plot the calculated contributions as bars
<code>plot.models</code>	Plot response curves of the selected models (using linear predictor scale before transformation into the response scale)
<code>response.curves</code>	Plot combined response curves of the selected models (using response scale after transformation by the inverse link function)
<code>model.anova</code>	Calculate an ANOVA table by removing each term from the full model
<code>validate.models</code>	Validate the selected models for each Ys using both simple validation and cross-validation
<code>predictions</code>	Calculate predictions for each Ys using selected model and new data (XXXpred) and save them in <code>gr.predmat</code>
<code>export.predictions</code>	Export saved predictions from <code>gr.predmat</code> to a text file that can be directly imported in Arcview as an ASCII GRID
<code>plot.predictions</code>	Plot saved predictions from <code>gr.predmat</code>
<code>lookup.tables</code>	Calculate lookup tables describing partial responses curves at the linear predictor scale for each Ys using minimum and maximum values for each Xs contained in XXXpred. These values are derived from min and max values of the GRIDS that are used to build spatial predictions in Arcview.
<code>plotpar</code>	number of rows and columns for multiple graphical outputs
<code>prescol</code>	Color for presences in graph outputs
<code>abscol</code>	Color for absences in graph outputs
<code>nbar</code>	Number of bars in histograms
<code>show.details</code>	Verbose output of stepwise selection
<code>StdError</code>	Show pointwise error around partial plot
<code>cvgroups</code>	Number of groups for cross validation
<code>resolution</code>	Resolution for gids maps

Value

MODELCALLS	Calls of selected models
VALIDATION	Validation statistics for selected models
ALONE.CONTRIB	Predictor alone contributions for selected models
DROP.CONTRIB	Predictor drop contributions for selected models
MODEL.CONTRIB	Predictor model contributions for selected models

Author(s)

Anthony.Lehmann@unige.ch

References

A. Lehmann, J.R. Leathwick & J.McC. Overton, 2002. GRASP. *Ecological Modelling*, 157: 189-207

HOMEPAGE: <http://www.unige.ch/ia/climate/grasp>

RELATED: <http://www.spacedoor.net>

See Also

grasp.in

Examples

```
data(YYY) # reads in YYY,XXX and XXXpred demo dataset
data(XXX)
data(XXXpred)
grasp.in(YYY,XXX,XXXpred) # initialize a new grasp session

grasp(2:3,c(4:6,8:9), title = "GRASP: ", path = "", gr.fam = "binomial", weights = TRUE, mak
# run a full grasp analysis on 2 responses with 5 predictors
```

`grasp.adm.dump` *Internal GRASP function*

Description

This function saves the current version of your GRASP functions in a dump file that can be transported to another R session or to another computer.

Usage

```
grasp.adm.dump()
```

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

`grasp.data.read` *Internal GRASP function*

Description

Read and load `YYY,XXX, XXXpred` text tables from disk

Usage

```
grasp.data.read(path="")
```

Arguments

`path` The path (directory) in which the files are located

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.exp.cormat *Internal GRASP function*

Description

This function calculates the correlation between PVs and plots a correlation matrix, eliminating too highly correlated PVs.

Usage

```
grasp.exp.cormat(gr.Yi, cols = gr.selX, thin = OPTIONS$CORTHIN, corplot = TRUE)
```

Arguments

<code>gr.Yi</code>	A vector containing the selected responses
<code>cols</code>	A vector containing the selected predictors
<code>thin</code>	Acceptable level of correlation
<code>corplot</code>	Plot it or not ?

Details

The first aim of this function is to calculate correlation between PVs and to plot a matrix of correlation. The second aim is to eliminate PVs too highly correlated. The acceptable level of correlation is decided in the `OPTIONS`. Limits defined by `grasp.msk.limits()` are applied to the data shown in these plots if desired. Selecting the reset correlation for Ys option will remove all correlation selection define in earlier session for the selected Y. This is to make sure that no correlation limits are applied by mistake.

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.exp.datamap *Internal GRASP function*

Description

This function is a visual check of the spatial distribution of the response data

Usage

```
grasp.exp.datamap(gr.Yi)
```

Arguments

gr.Yi A vector containing the selected responses

Details

This function has been introduced in GRASP as a visual check of the spatial distribution of the input response data. Limits defined by `grasp.msk.limits()` are applied to the data shown in these maps. Geographic coordinates of points must be contained in the XXX table under column names (XXX\$CX, XXX\$CY).

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.exp.distry *Internal GRASP function*

Description

This function plots the distribution of responses as histograms with additional statistics supplied

Usage

```
grasp.exp.distry(gr.Yi, nbar = OPTIONS$NBAR)
```

Arguments

gr.Yi A vector containing the selected responses
nbar Number of histogram bars in the plot

Details

This function allows plotting the distribution of responses as histograms. Additional statistics are supplied. This is an important step in deciding which family of distribution to use: - presence-absence data >> binomial - frequency, continuous between 0 and 1 >> binomial - counts, richness, positive integers >> poisson - weights, size, biomass, continuous symmetrical >> gaussian

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

grasp.exp.histo *Internal GRASP function*

Description

This function plots the responses vs. the predictors.

Usage

```
grasp.exp.histo(gr.Yi, sX = gr.selX, nbar = OPTIONS$NBAR)
```

Arguments

gr.Yi	A vector containing the selected responses
sX	A vector of the selected predictors
nbar	Number of histogram bars in the plot

Details

This function plots responses (RV) versus predictor variables (PV). Limits defined by `grasp.msk.limits()` are applied to the data shown in these histograms. A smoothing function is applied (spline) to the relationship between the responses and each continuous predictor. The degrees of freedom for the smooth function is the same as the one required to build the model and saved in `OPTIONS$DF2`. Weights are applied when `WEIGHTS` matrix contains values different from 1

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

grasp.exp.RvsP *Internal GRASP function*

Description

This function is a visual check of both response and predictor variables.

Usage

```
grasp.exp.RvsP(gr.Yi, sX = gr.selX)
```

Arguments

gr.Yi	A vector of the selected responses
sX	A vector of the selected predictors

Details

This function is another visual check of both response (RV) and predictor variables (PV). Limits defined by `grasp.msk.limits()` can be applied to the data shown in these histograms if desired. First a simple histogram of the selected explanatory variables is produced. With binomial data, the number of response observations observed in each bar is drawn and written on top of each of them. A dashed line is added representing the overall proportion of response observations greater than 0 (presences) in comparison with the total number of observations. Finally, a continuous line is added showing whether the proportion of presences in each bar is higher or lower than the overall proportion.

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

grasp.exp.summary *Internal GRASP function*

Description

`grasp.exp.summary` prints summaries of XXX and YYY in order to check for missing data (NA) and distribution of min and max values of predictors and responses.

Usage

```
grasp.exp.summary(gr.Yi)
```

Arguments

`gr.Yi` A vector containing the selected responses

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

`grasp.exp.xpredplot`

Internal GRASP function

Description

This function maps the predictors stored in XXXpred at the resolution set in the GUI

Usage

```
grasp.exp.xpredplot (gr.Xi)
```

Arguments

`gr.Xi` A vector containing the selected predictors

Details

Aggregation of pixels is calculated if the chosen resolution is coarser than the actual resolution in XXXpred.

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

`grasp.in`*Generalized Regression Analysis and Spatial Prediction, initializing session*

Description

This function is used only when beginning with new data or when modifying names of dimensions of `YYY`, `XXX`, `XXXpred`

Usage

```
grasp.in(Ymat = YYY, Xmat = XXX, Xpred = XXXpred, Xlut = NULL)
```

Arguments

<code>Ymat</code>	is the dataframe containing the response variables (RV): Each column represents a variable and each row an observation. Columns must have unique names not included in other variable names. Data type: single, double.
<code>Xmat</code>	is the dataframe containing the predictor variables (PV): Each column represents a variable and each rows an observation. The number of rows must be the same as for <code>YYY</code> . Columns must have unique names not included in other variable names. Data type: single, double, integer, factor.
<code>Xpred</code>	is the dataframe containing the explanatory data (PV) to predict from: Each column represents a variable and each rows a new observation to predict from. Columns must have unique names identical to those in <code>XXX</code> and missing data must be avoided
<code>Xlut</code>	If predictions are made on large datasets (more than 250000 pixels) it is better to use lookup tables to describe models and build predictions in Arcview. <code>XXXlut</code> must contain 2 rows (rows 1 and 2) with the minimum and maximum values for each variable in the prediction dataset built as Grids in Arcview.

Author(s)

Anthony.Lehmann@unige.ch

References

A. Lehmann, J.R. Leathwick & J.McC. Overton, 2002. GRASP. Ecological Modelling, 157: 189-207

HOME PAGE: <http://www.unige.ch/ia/climate/grasp>

RELATED: <http://www.spacedoor.net>

See Also

grasp

Examples

```
data(YYY) # reads in YYY,XXX and XXXpred demo dataset
data(XXX)
data(XXXpred)

grasp.in(YYY,XXX,XXXpred) # initialize a new grasp session

grasp( 2:3,c(4:6,8:9), title = "GRASP: ", path = "", gr.fam = "binomial", weights = TRUE, ma
plot.histograms = TRUE, plot.resvpspred = TRUE, plot.xpred = TRUE,plot.correlation = TRUE, s
```

grasp.int.contplot *Internal GRASP function*

Description

grasp.cont.plot() allows plotting of the drop, model and alone contributions calculated by grasp.contrib() and saved in DROP.CONTRIB, MODEL.CONTRIB and ALONE.CONTRIB dataframes.

Usage

```
grasp.int.contplot(gr.Yi)
```

Arguments

gr.Yi A vector containing the selected responses

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.int.contrib *Internal GRASP function*

Description

Calculates the contributions...

Usage

```
grasp.int.contrib(gr.Yi)
```

Arguments

gr.Yi A vector containing the selected responses

Details

Three methods are used to calculate contributions: -Drop contributions are obtained by dropping each explanatory variable from the model and by calculating the associated change in deviance. Drop contributions indicate the marginal contribution of each variable. -Alone contributions are calculated by creating new models with only one predictor. Model contributions are calculated by the range between maximum and minimum contribution of each variable in the linear predictor scale before transformation by the link function. Alone contributions better reflect the potential of each variable. -Model contribution is the only measure which gives an indication of the contribution of the variable within the selected model. It corresponds to the possible range of variation on the scale of the linear predictor as shown on the vertical axes of plots produced by grasp.int.plot (see in next paragraphs)

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

grasp.int.plot *Internal GRASP function*

Description

Models resulting from grasp.step() and saved in MODELCALLS are plotted with the partial response curves of each selected explanatory variable.

Usage

```
grasp.int.plot(gr.Yi)
```

Arguments

gr.Yi A vector containing the selected responses

Details

Each response curves is the result of the backfitting algorithm used by Splus function GAM to calculate the additive contribution of each variable using non parametric smoothing methods. Results are expressed in the scale of the "additive predictor" before transformation into the prediction scale by the inverse link function. Dashed lines represent upper and lower point wise twice-standard-error curves.

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

grasp.int.response *Internal GRASP function*

Description

This complex function allows simultaneous plotting of predictions from different responses in relation to continuous predictor variables

Usage

```
grasp.int.response(sY, sX)
```

Arguments

sY A vector containing the selected responses

sX A vector of the selected predictors

Details

This complex function allows simultaneous plotting of predictions from different responses in relation to continuous predictor variables. The result is in the scale of the responses. Better results are obtained if the `OPTIONS$RECALCULATEWEIGHTS` (on page A) is set to `TRUE`, as this gives the same weights to presences and absences in the analyses. The contributions of all selected predictors in a model are set to 0 except for one predictor. A new prediction is made for that predictor given a 0 contribution of all other variables

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.mod.anova *Internal GRASP function*

Description

An ANOVA table is constructed by testing the significance of removing in turn each predictor from the selected model.

Usage

```
grasp.mod.anova (gr.Yi)
```

Arguments

`gr.Yi` A vector containing the selected responses

Details

This function gives a different result from the `anova` function of `Splus` `anova(model, test="F")`, which tests the successive inclusion of variables into the model and is therefore dependant on the order of variable entry.

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.mod.validate *Internal GRASP function*

Description

Function allows a visual check of the relationship between fitted and observed data.

Usage

```
grasp.mod.validate (gr.Yi)
```

Arguments

gr.Yi A vector containing the selected responses

Details

A cross-validation is made with subsets of the entire dataset, where each subset contains an equal number of randomly selected data points. Each subset is then dropped from the model, the model is recalculated and predictions made for the omitted data points. Combination of the predictions from the different subsets is then plotted against the observed data. A ROC (area under curve) and a COR (Spearman Correlation) statistics is used with binomial data.(see Fielding & Bell, 1997). A simple correlation coefficient is calculated for Poisson and normally distributed data.

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.msk.limits *Internal GRASP function*

Description

It restricts the data within limits defined for selected response variables by keeping a given number of 0 observations on each side of the last presence along selected predictors.

Usage

```
grasp.msk.limits (gr.Yi, sX = OPTIONS$SELXLIM, npast = OPTIONS$NPAST)
```

Arguments

gr.Yi	A vector containing the selected responses
sX	Defined in the options
npast	Defined in the options

Details

The `grasp.msk.limits()` function is optional. It restricts the data within limits defined for selected response variables by keeping a given number of 0 observations on each side of the last presence along selected predictors (Austin and Nichols, 1997; Leathwick, 1998). You can change the X variables used to set the limits and the number of observations kept on each side of the last positive observations.

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.options

Internal GRASP function

Description

Set main GRASP options

Usage

```
grasp.options(title = "GRASP: ", path = "c:/temp", plotpar = c(3, 3), npast = 0,
selxlim = gr.selX, corlim = 100, corplot = TRUE, nbar = 10, df1 = 0, df2 = 4, test
resetstart = TRUE, showdetail = FALSE, cvgroups = 5,
resolution = (max(XXXpred$CX) - min(XXXpred$CX)) / (length(unique(XXXpred$CX)) - 1),
StdError = TRUE)
```

Arguments

title	Title in the plots
path	path where outputs will be written
plotpar	A vector of the number of colonnes and lines in plots
npast	For limits, the number of data to be left out (limited)
selxlim	For limits, a vector of predictors to which limits will be applied
corlim	...
corplot	If true, correlation will be plotted

nbar	Number of bars in histograms
df1	First degrees of freedom
df2	Second degrees of freedom
test	Which test to use, F-test by default
plimit	...
direction	...
weights	If TRUE, weights will be used, they are stored in the WEIGHTS dataframe generated by grasp.in
mincont	...
startwith	...
resetstart	...
showdetail	...
cvgroups	...
resolution	...
StdError	...

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.pred

Internal GRASP function

Description

It predicts each Response Variables (RVs) using a new set of Predictor Variables (PVs) contained in XXXpred.

Usage

```
grasp.pred(gr.Yi)
```

Arguments

gr.Yi A vector containing the selected responses

Details

This function is certainly the most demanding in time and computer resources. Its purpose is to predict each Response Variables (RVs) using a new set of Predictor Variables (PVs) contained in XXXpred. It uses the models selected in `grasp.step()` and stored in MODELCALLS. Predictions are calculated using the Splus function `pred.gam(MODELCALLS, XXXpred, type="response")`. When `type="response"` is selected, the predictions are on the scale of the response, and are monotone transformations of the additive predictors, using the inverse link function.

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

`grasp.pred.export` *Internal GRASP function*

Description

This function will export the predictions made in `grasp.pred()` and stored in `gr.predmat` into an ASCII file ready to be read by Import Grid in ArcView or ArcGIS.

Usage

```
grasp.pred.export (gr.Yi)
```

Arguments

`gr.Yi` A vector containing the selected responses

Details

The resolution of the exported GRID can be set on page R of the GUI. When observations are merging in a new cell a mean prediction is calculated (agglomeration).

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

grasp.pred.lookup *Internal GRASP function*

Description

`grasp.pred.lookup()` replaces the use of both `grasp.pred()` and `grasp.pred.export()`. This function prepares lookup tables from the models built in `grasp.step()` and stored in `MODELCALLS`.

Usage

```
grasp.pred.lookup(gr.Yi)
```

Arguments

`gr.Yi` A vector containing the selected responses

Details

The resulting lookup table are saved under `OPTIONS$PATHLUT` and can be read in Arcview by an Avenue script written to build spatial predictions by reclassifying the predictors maps corresponding to those selected in the model.

Author(s)

Anthony.Lehmann@unige.ch

See Also

`grasp` [grasp](#), `grasp.in` [grasp.in](#)

grasp.pred.plot *Internal GRASP function*

Description

This function maps within R the predictions made by `grasp.pred()` and stored in `gr.predmat`.

Usage

```
grasp.pred.plot(gr.Yi)
```

Arguments

`gr.Yi` A vector containing the selected responses

Details

The resolution of the exported GRID can be set with the resolution option.

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

grasp.step

Internal GRASP function

Description

The function selects significant predictors for each response variable

Usage

```
grasp.step(gr.Yi)
```

Arguments

gr.Yi A vector containing the selected responses

Details

This function is the central part of GRASP. It selects significant predictors for each response variable. By default the stepwise procedure starts with a full model containing all potential variables (selX), and goes in loop to try to eliminate one variable at the time. At each step, the less significant variable is dropped from the model, and the loop starts again with the remaining variables. Tests are used to decide whether a variable should be dropped, kept or reintroduced. Available tests are : AIC, BIC, F, Chisq, BRUTO and CROSS

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

grasp.step.aic *Internal GRASP function*

Description

Stepwise AIC selection: an Akaike IC criteria is used to select variables, very conservative (default test)

Usage

```
grasp.step.aic(object, scope, scale, direction, trace, keep, steps, ...)
```

Arguments

object	An object of class 'gam' or any of it's inheritants.
scope	...
scale	...
direction	...
trace	If true, verbose output
keep	...
steps	Number of stepwise steps
...	...

Author(s)

Function step.gam by Trevor Hastie (gam library). Modified by Anthony lehmann (Anthony.Lehmann@unige.ch).

See Also

[step.gam](#) for the generic function.

grasp.step.anova *Internal GRASP function*

Description

Stepwise F and Chisq test selection

Usage

```
grasp.step.anova(object, scope, scale, direction = c("both", "backward", "forward")  
                 trace = TRUE, keep = NULL, steps = 1000, ...)
```

Arguments

object	...
scope	...
scale	...
direction	...
trace	If true, verbose output
keep	...
steps	Number of stepwise steps
...	...

Details

Stepwise F and Chisq test selection: - F: an F test is used to test whether predictors are significant (for quasi models) - CHI: an Chi test is used to test whether predictors are significant variable

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.step.bic *Internal GRASP function*

Description

Stepwise BIC selection: a Bayesian IC criteria is used to select variables, very selective

Usage

```
grasp.step.bic(object, scope, scale, direction = c("both", "backward", "forward"),
```

Arguments

object	...
scope	...
scale	...
direction	...
trace	If true, verbose output
keep	...
steps	Number of stepwise steps
...	...

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

`grasp.step.bruto` *Internal GRASP function*

Description

Stepwise BRUTO selection

Usage

```
grasp.step.bruto(gr.Yi, selectedX)
```

Arguments

<code>gr.Yi</code>	A vector containing the selected responses
<code>selectedX</code>	Selected predictors

Details

Use the bruto methods found in library mda at: <http://www.stats.ox.ac.uk/pub/MASS4/Software.html>.

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

grasp.step.contrib *Internal GRASP function*

Description

Stepwise CONTRIB selection: Stepwise selection based on minimum percentage of Model contribution.

Usage

```
grasp.step.contrib(object, scope, scale, direction = c("both", "backward", "forward"))
```

Arguments

object	...
scope	...
scale	...
direction	...
trace	If true, verbose output
keep	...
steps	Number of stepwise steps
...	Additional arguments to be passed on to 'keep'

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

grasp.step.cross *Internal GRASP function*

Description

Stepwise CROSS selection: A cross validation method is used at each step of a BIC selection to keep at the end the model presenting the highest statistics of cross validation.

Usage

```
grasp.step.cross(object, scope, scale, direction = c("both", "backward", "forward"))
```

Arguments

object	...
scope	...
scale	...
direction	...
trace	If true, verbose output
keep	...
steps	Number of stepwise steps
...	...

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

`grasp.step.crossplot`
Internal GRASP function

Description

None

Usage

`grasp.step.crossplot (gr.Yi)`

Arguments

`gr.Yi` A vector containing the selected responses

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), [grasp.in](#) [grasp.in](#)

`grasp.step.crossval`
Internal GRASP function

Description

None

Usage

`grasp.step.crossval(tested.model)`

Arguments

`tested.model` ...

Author(s)

Anthony.Lehmann@unige.ch

See Also

`grasp` [grasp](#), `grasp.in` [grasp.in](#)

`grasp.step.full` *Internal GRASP function*

Description

This function create models containing all candidates predictors.

Usage

`grasp.step.full(gr.Yi)`

Arguments

`gr.Yi` A vector containing the selected responses

Author(s)

Anthony.Lehmann@unige.ch

See Also

`grasp` [grasp](#), `grasp.in` [grasp.in](#)

`grasp.step.rocplot` *Internal GRASP function*

Description

None

Usage

```
grasp.step.rocplot(z1, z2)
```

Arguments

<code>z1</code>	...
<code>z2</code>	...

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

`grasp.step.scope` *Internal GRASP function*

Description

`grasp.step.scope()` creates a model scope for the stepwise selection of variables with all selected Predictor Variables including factors and smoothed continuous variables, using both degree of freedom (DF1 and DF2)

Usage

```
grasp.step.scope(sX)
```

Arguments

<code>sX</code>	...
-----------------	-----

Author(s)

Anthony.Lehmann@unige.ch

See Also

grasp [grasp](#), grasp.in [grasp.in](#)

`grasp.step.start` *Internal GRASP function*

Description

`grasp.step.start()` creates a starting formula with all selected Predictor Variables including factors and smoothed continuous variables.

Usage

```
grasp.step.start(sX)
```

Arguments

`sX` ...

Author(s)

Anthony.Lehmann@unige.ch

See Also

`grasp` [grasp](#), `grasp.in` [grasp.in](#)

`XXX` *XXX: contains predictor variables for GRASP*

Description

`XXX` is the dataframe containing the predictor variables (PV): Each column represents a variable and each rows an observation.

Usage

```
data(XXX)
```

Format

A data frame with 7563 observations on the following 12 variables.

ID a numeric vector, Unique ID, in same order as in `YYY`

CX a numeric vector, X geographic coordinate

CY a numeric vector, Y geographic coordinate

TEMP a numeric vector, Mean annual temperature

RAD a numeric vector, Mean annual solar radiation

PREC a numeric vector, Precipitation
NNESS a numeric vector, Northness
SLOPE a numeric vector, Slope
ASPECT a factor with levels NORTH NULL SOUTH

Details

The number of rows must be the same as for `YYY`. Columns must have unique names not included in other variable names. Data type: single, double, integer, factor.

Source

<http://www.cscf.ch>

See Also

`YYY`, `XXXpred`

Examples

```
data(XXX)
summary(XXX)
```

XXXpred

XXXpred: contains predictor variables used in GRASP to predict from

Description

`XXXpred` is the dataframe containing the explanatory data (PV) to predict from: Each column represents a variable and each rows a new observation to predict from.

Usage

```
data(XXXpred)
```

Format

A data frame with 6620 observations on the following 12 variables.

ID a numeric vector, Unique ID, in same order as in `YYY`
CX a numeric vector, X geographic coordinate
CY a numeric vector, Y geographic coordinate
TEMP a numeric vector, Mean annual temperature
RAD a numeric vector, Mean annual solar radiation
PREC a numeric vector, Precipitation
NNESS a numeric vector, Northness
SLOPE a numeric vector, Slope
ASPECT a factor with levels NORTH NULL SOUTH

Details

Columns must have unique names identical to those in XXX and missing data must be avoided. Categorical variables must be transformed to factor variable in XXX and XXXpred.

Source

<http://www.cscf.ch>

Examples

```
data(XXXpred)
summary(XXXpred)
```

YYY

YYY: contains response variables for GRASP

Description

YYY is the trial dataframe containing the response variables (RV): Each column represents a variable and each row an observation.

Usage

```
data(YYY)
```

Format

A data frame with 7563 observations on the following 4 variables.

ID a numeric vector, Unique ID, in same order as in XXX

SP1 a numeric vector, example of binomial data with presence/absence of species 1

SP2 a numeric vector, example of binomial data with presence/absence of species 2

RICHNESS a numeric vector, example of Poisson data with species richness

Details

Columns must have unique name not included in other variable names. Data type: single, double. Data distribution: Binomial, Poisson, Gaussian

Source

<http://www.cscf.ch>

Examples

```
data(YYY)
summary(YYY)
```

Index

*Topic **datasets**

XXX, 29

XXXpred, 30

YYY, 31

*Topic **models**

grasp, 2

grasp.adm.dump, 5

grasp.data.read, 5

grasp.exp.cormat, 6

grasp.exp.datamap, 7

grasp.exp.distry, 7

grasp.exp.histo, 8

grasp.exp.RvsP, 9

grasp.exp.summary, 9

grasp.exp.xpredplot, 10

grasp.in, 11

grasp.int.contplot, 12

grasp.int.contrib, 13

grasp.int.plot, 13

grasp.int.response, 14

grasp.mod.anova, 15

grasp.mod.validate, 16

grasp.msk.limits, 16

grasp.options, 17

grasp.pred, 18

grasp.pred.export, 19

grasp.pred.lookup, 20

grasp.pred.plot, 20

grasp.step, 21

grasp.step.aic, 22

grasp.step.anova, 22

grasp.step.bic, 23

grasp.step.bruto, 24

grasp.step.contrib, 25

grasp.step.cross, 25

grasp.step.crossplot, 26

grasp.step.crossval, 27

grasp.step.full, 27

grasp.step.rocplot, 28

grasp.step.scope, 28

grasp.step.start, 29

grasp, 2, 5–10, 12–21, 23–29

grasp.adm.dump, 5

grasp.data.read, 5

grasp.exp.cormat, 6

grasp.exp.datamap, 7

grasp.exp.distry, 7

grasp.exp.histo, 8

grasp.exp.RvsP, 9

grasp.exp.summary, 9

grasp.exp.xpredplot, 10

grasp.in, 5–10, 11, 12–21, 23–29

grasp.int.contplot, 12

grasp.int.contrib, 13

grasp.int.plot, 13

grasp.int.response, 14

grasp.mod.anova, 15

grasp.mod.validate, 16

grasp.msk.limits, 16

grasp.options, 17

grasp.pred, 18

grasp.pred.export, 19

grasp.pred.lookup, 20

grasp.pred.plot, 20

grasp.step, 21

grasp.step.aic, 22

grasp.step.anova, 22

grasp.step.bic, 23

grasp.step.bruto, 24

grasp.step.contrib, 25

grasp.step.cross, 25

grasp.step.crossplot, 26

grasp.step.crossval, 27

grasp.step.full, 27

grasp.step.rocplot, 28

grasp.step.scope, 28

grasp.step.start, 29

step.gam, 22

xxx, 29

xxxpred, 30

yyy, 31