Package ‘SCGLR’

March 16, 2016

Type Package

Title Supervised Component Generalized Linear Regression

Version 2.0.3

Date 2016-03-07

Description The Fisher Scoring Algorithm is extended so as to combine Partial Least Squares regression with Generalized Linear Model estimation in the multivariate context.

License CeCILL-2 | GPL-2

Depends R (>= 3.0.0)

Imports Matrix,Formula,expm,graphics,grid,ggplot2 (>= 2.0.0),pROC,scales

Suggests parallel,gridExtra,knitr

LazyData yes

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Author Guillaume Cornu [cre, aut], Frederic Mortier [aut], Catherine Trottier [aut], Xavier Bry [aut]

Maintainer Guillaume Cornu <gcornu@cirad.fr>

Repository CRAN

Date/Publication 2016-03-16 18:16:10

R topics documented:

scglr-package .................................................. 2
barplot.SCGLR .................................................. 3
critConvergence .................................................. 3
customize .................................................. 4
SCGLR implements a new Partial Least Squares regression approach in the multivariate generalized linear framework. The method allows the joint modeling of random variables from different exponential family distributions, searching for common PLS-type components. `scglr` and `scglrCrossVal` are the two main functions. The former constructs the components and performs the parameter estimation, while the latter selects the appropriate number of components by cross-validation. Dedicated plots, print, and summary functions are available. The package contains also an ecological dataset dealing with the abundance of multiple tree genera given a large number of geo-referenced environmental variables.

**Author(s)**

Mortier F., Trottier C., Cornu G., Bry X.

**References**

**Description**

A custom plot for SCGLR objects.

**Usage**

```r
## S3 method for class 'SCGLR'
barplot(height, ..., plane = NULL)
```

**Arguments**

- `height`: object of class 'SCGLR', usually a result of running `scglr`.
- `plane`: a size-2 vector (or comma separated string) indicating which components are plotted (eg: c(1,2) or "1,2").
- `...`: optional arguments.

**Value**

an object of class `ggplot`.

**See Also**

For barplot application see examples in `plot.SCGLR`.

---

**Description**

Auxiliary function for controlling SCGLR fitting

**Usage**

```r
critConvergence(tol = 1e-06, maxit = 50)
```

**Arguments**

- `tol`: positive convergence threshold.
- `maxit`: integer, maximum number of iterations.

**Value**

a list containing elements named as the arguments.
### Customize

#### Plot customization

**Description**

Parameters used to choose what to plot and how. These parameters are given to `plot.SCGLR` and `pairs.SCGLR`.

**Details**

Parameter name can be abbreviated (e.g. pred.col will be understood as predictors.color). Options can be set globally using `options(“plot.SCGLR”).` It will then provide default values that can be further overridden by giving explicit parameter value.

<table>
<thead>
<tr>
<th>parameter name</th>
<th>type (default value)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>string (NULL).</td>
<td>Main title of plot (override built-in).</td>
</tr>
<tr>
<td>labels.auto</td>
<td>logical (TRUE).</td>
<td>Should covariate or predictor labels be aligned with arrows.</td>
</tr>
<tr>
<td>labels.offset</td>
<td>numeric (0.01).</td>
<td>Offset by which labels should be moved from tip of arrows.</td>
</tr>
<tr>
<td>labels.size</td>
<td>numeric (1).</td>
<td>Relative size for labels. Use it to globally alter label size.</td>
</tr>
<tr>
<td>expand</td>
<td>numeric (1).</td>
<td>Expand factor for windows size. Use it for example to make room for clipped labels.</td>
</tr>
<tr>
<td>threshold</td>
<td>numeric (1).</td>
<td>Should we draw observations.</td>
</tr>
<tr>
<td>observations</td>
<td>numeric (1).</td>
<td>Point size.</td>
</tr>
<tr>
<td>observations.size</td>
<td>character (&quot;black&quot;).</td>
<td>Point color.</td>
</tr>
<tr>
<td>observations.color</td>
<td>character (&quot;black&quot;).</td>
<td>Point transparency.</td>
</tr>
<tr>
<td>observations.factor</td>
<td>logical (FALSE).</td>
<td>Paint observations according to factor (specify factor).</td>
</tr>
<tr>
<td>predictors</td>
<td>logical or array of characters (TRUE).</td>
<td>Should we draw predictors and optionally which one (TRUE means all).</td>
</tr>
<tr>
<td>predictors.color</td>
<td>string (&quot;red&quot;).</td>
<td>Base color used to draw predictors.</td>
</tr>
<tr>
<td>predictors.alpha</td>
<td>numeric (1).</td>
<td>Overall transparency for predictors (0 is transparent, 1 is opaque).</td>
</tr>
<tr>
<td>predictors.arrows</td>
<td>logical (TRUE).</td>
<td>Should we draw arrows for predictors.</td>
</tr>
<tr>
<td>predictors.arrows.color</td>
<td>string (predictors.color).</td>
<td>Specific color for predictor arrows.</td>
</tr>
<tr>
<td>predictors.arrows.alpha</td>
<td>numeric (predictors.alpha).</td>
<td>Transparency for predictor arrows.</td>
</tr>
<tr>
<td>predictors.labels</td>
<td>logical (TRUE).</td>
<td>Should we draw labels for predictors.</td>
</tr>
<tr>
<td>predictors.labels.color</td>
<td>string (predictors.color).</td>
<td>Specific color for predictor labels.</td>
</tr>
<tr>
<td>predictors.labels.size</td>
<td>numeric (labels.size).</td>
<td>Specific size for predictor labels.</td>
</tr>
<tr>
<td>predictors.labels.auto</td>
<td>logical (labels.auto).</td>
<td>Should predictor labels be aligned with arrows.</td>
</tr>
<tr>
<td>covariates</td>
<td>logical or array of characters (TRUE).</td>
<td>Should we draw covariates and optionally which one (TRUE means all).</td>
</tr>
<tr>
<td>covariates.color</td>
<td>string (&quot;black&quot;).</td>
<td>Base color used to draw covariates.</td>
</tr>
<tr>
<td>covariates.alpha</td>
<td>numeric (1).</td>
<td>Overall transparency for covariates (0 is transparent, 1 is opaque).</td>
</tr>
<tr>
<td>covariates.arrows</td>
<td>logical (TRUE).</td>
<td>Should we draw arrows for covariates.</td>
</tr>
<tr>
<td>covariates.arrows.color</td>
<td>string (covariates.color).</td>
<td>Specific color for covariate arrows.</td>
</tr>
<tr>
<td>covariates.arrows.alpha</td>
<td>numeric (covariates.alpha).</td>
<td>Transparency for covariate arrows.</td>
</tr>
<tr>
<td>covariates.labels</td>
<td>logical (TRUE).</td>
<td>Should we draw labels for predictors.</td>
</tr>
<tr>
<td>covariates.labels.color</td>
<td>string (covariates.color).</td>
<td>Specific color for predictor labels.</td>
</tr>
<tr>
<td>covariates.labels.alpha</td>
<td>numeric (covariates.alpha).</td>
<td>Transparency for covariate labels.</td>
</tr>
<tr>
<td>covariates.labels.size</td>
<td>numeric (labels.size).</td>
<td>Specific size for covariate labels.</td>
</tr>
</tbody>
</table>
genus

Sample dataset of abundance of genera in tropical moist forest

Description

Genus gives the abundance of 27 common tree genera in the tropical moist forest of the Congo-Basin and 40 geo-referenced environmental variables on one thousand 8 by 8 km plots (observations). Each plot’s data was obtained by aggregating the data measured on a variable number of previously sampled 0.5 ha sub-plots. Geo-referenced environmental variables were used to describe the physical factors as well as vegetation characteristics. 14 physical factors were used pertaining the description of topography, geology and rainfall of each plot. Vegetation is characterized through 16-days enhanced vegetation index (EVI) data.

Format

gen1 to gen27 abundance of the 27 common genera.
altitude above-sea level in meters.
pluvio_yr mean annual rainfall.
forest classified into seven classes.
pluvio_1 to pluvio_12 monthly rainfalls.

covariates.labels.auto logical (labels.auto). Should covariate labels be aligned with arrows.
factor logical or character (FALSE). Should we draw a factor chosen among additionnal variables (TRUE mean first one).
factor.points logical (TRUE). Should symbol be drawn for factors.
factor.points.size numeric (4). Symbol size.
factor.points.shape numeric (13). Point shape.
factor.labels logical (TRUE). Should factor labels be drawn.
factor.labels.color string ("black"). Color used to draw labels.
factor.labels.size numeric (labels.size). Specific size for factor labels.

Examples

```r
## Not run:
# setting parameters
plot(genus.scglr)
plot(genus.scglr, predictors=TRUE)
plot(genus.scglr, predictors=TRUE, pred.arrows=FALSE)

# setting global style
options(plot.scglr=list(predictors=TRUE, pred.arrows=FALSE))
plot(genus.scglr)

# setting custom style
myStyle <- list(predictors=TRUE, pred.arrows=FALSE)
plot(genus.scglr, style=myStyle)

## End(Not run)
```
geology 5-level geological substrate.
evi_1 to evi_23 16-days enhanced vegetation indexes.
lon and lat position of the plot centers.
surface sampled area.

Note
The use of this dataset for publication must make reference to the CoForChange project.

Author(s)
CoForChange project

References

infoCriterion  Function that calculates cross-validation selection criteria

Description
Function that calculates cross-validation selection criteria

Usage
infoCriterion(ynew, pred, family, type, size = NULL, npar = 0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ynew</td>
<td>data matrix corresponding to the observations used as test sample.</td>
</tr>
<tr>
<td>pred</td>
<td>predicted value of the linear predictor obtained from Xnew and the estimated parameters.</td>
</tr>
<tr>
<td>family</td>
<td>a vector of the same length as the number of responses containing characters identifying the distribution families of the dependent variables. &quot;bernoulli&quot;, &quot;binomial&quot;, &quot;poisson&quot; or &quot;gaussian&quot; are allowed.</td>
</tr>
<tr>
<td>type</td>
<td>information criterion used. Likelihood, aic, bic, aicc or Mean Square Prediction Error (mspe) are defined. Area Under ROC Curve (auc) also defined for Bernoulli cases only.</td>
</tr>
<tr>
<td>size</td>
<td>describes the number of trials for the binomial dependent variables. A (number of statistical units * number of binomial dependent variables) matrix is expected.</td>
</tr>
<tr>
<td>npar</td>
<td>number of parameters used for penalisation.</td>
</tr>
</tbody>
</table>
Methods

Value

a matrix containing the criterion value for each dependent variable (row) and each number of components (column).

<table>
<thead>
<tr>
<th>Methods</th>
<th>Regularization criterion types</th>
</tr>
</thead>
</table>

Description

- LPLS for PLS-type SCGLR
- SR Method iterative normed gradient (ING) for Structural Relevance

Usage

```r
methodLPLS()
methodSR(phi = "vpi", l = 1, s = 1/2, maxiter = 1000, epsilon = 1e-06, bailout = 10)
```

Arguments

- `phi` character string describing structure relevance used in the regularization process. Allowed values are "vpi" for Variable Powered Inertia and "cv" for Component Variance. Default to "vpi".
- `l` is a numeric argument (>1) tuning the importance of variable bundle locality.
- `s` is a numeric argument (in [0,1]) tuning the strength of structural relevance with respect to goodness of fit.
- `maxiter` integer for maximum number of iterations of SR function
- `epsilon` positive convergence threshold
- `bailout` integer argument

multivariateFormula

Formula construction

Description

Helper function for building multivariate scglr formula.

NOTE: Interactions involving factors are not allowed for now. For interactions between two quantitative variables, use `I(x*y)` as usual.

Usage

```r
multivariateFormula(namesY, namesX, namesAX = NULL)
```
**Arguments**

- **namesY**: a vector of character containing the names of the dependent variables.
- **namesX**: a vector of character containing the names of the covariates (X) involved in the components.
- **namesAX**: a vector of character containing the names of the additional covariates.

**Value**

- an object of class `Formula`.

---

**multivariateglmNfit  Multivariate generalized linear regression**

**Description**

`multivariateglm` is used to fit multivariate generalized linear models specified by a symbolic formula together with the distributions of the responses. This function performs a simple GLM fit for each dependent variable with the associated distribution.

**Usage**

```r
multivariateglmNfit(Y, comp, family, offset, size)
```

```r
multivariateglm(formula, data, family, size = NULL, offset = NULL, subset = NULL)
```

**Arguments**

- **Y**: matrix of dependent variables.
- **comp**: matrix of covariates.
- **family**: a vector of character giving the family distribution of each response.
- **offset**: used for the poisson dependent variables. A vector or a matrix of size: number of observations * number of Poisson dependent variables is expected.
- **size**: a matrix giving the number of trials for each Binomial dependent variable ncol(size) must be equal to the number of Binomial variables.
- **formula**: an object of class `Formula` (or one that can be coerced to that class): a symbolic description of the model to be fitted.
- **data**: the data frame to be modeled.
- **subset**: an optional vector specifying a subset of observations to be used in the fitting process.

**Value**

- the list, each item of which is the glm object associated with each response.
Examples

```r
## Not run:
library(SCGLR)

# load sample data
data(genus)

# get variable names from dataset
n <- names(genus)
ny <- n[grepl("^gen",n)] # Y <- names that begins with "gen"
mx <- n[-grepl("^gen",n)] # X <- remaining names

# remove "geology" and "surface" from nx as offset
# is offset and we want to use geology as additional covariate
nx <- nx[-!nx%in%c("geology","surface")]

# build multivariate formula
# we also add "lat*lon" as computed covariate
form <- multivariateFormula(ny,c(nx,"I(lat*lon)",c("geology")))

# split genus dataset
sub <- sample(1:nrow(genus),100,replace=FALSE)
sub_fit <- (1:nrow(genus))[-sub]

# define family
fam <- rep("poisson",length(ny))

# fit the model

# xnew, the design matrix associated to sub-sample used for prediction
# note rhs parameter is introduced to take into account that the
# covariate part of the formula is composed of two different sets
xnew <- model.matrix(form, data=genus[sub,], rhs=1:2)[-1]

# prediction based on the scglr approach
pred.scglr <- multivariatePredictGlm(xnew,family=fam,
  beta=genus.scglr$beta, offset=genus$surface[sub])
cor.scglr <- diag(cor(pred.scglr,genus[sub,ny]))
plot(cor.scglr, col="red", ylim=c(-1,1))

# prediction based on classical poisson glm
genus.glm <- multivariateGlm(formula=form, data=genus, family=fam,
  offset=genus$surface, subset=sub_fit)
coefs <- sapply(genus.glm,coef)

# rhs parameter is introduced to take into account that the
# covariate part of the formula is composed of two different sets
pred.glm <- multivariatePredictGlm(xnew,family=fam,beta=coefs,
  offset=genus$surface[sub])
cor.glm <- diag(cor(pred.glm,genus[sub,ny]))
```

multivariatePredictGlm

Function that predicts the responses from the covariates for a new sample

Description

Function that predicts the responses from the covariates for a new sample

Usage

multivariatePredictGlm(xnew, family, beta, offset = NULL)

Arguments

- `xnew`: a data frame containing the values of the covariates for the new sample.
- `family`: a vector of character specifying the distributions of the responses.
- `beta`: the matrix of coefficients estimated from the calibration sample.
- `offset`: used for the poisson dependent variables. A vector or a matrix of size: number of observations * number of Poisson dependent variables is expected.

Value

a matrix of predicted values.

Examples

```r
## Not run:
library(SCGLR)

data(genus)

# get variable names from dataset
n <- names(genus)
ny <- n[grep("gen",n)] # Y <- names that begins with "gen"
mx <- n[-grep("gen",n)] # X <- remaining names

# remove "geology" and "surface" from nx as surface is
# offset and we want to use geology as additional covariate
nx <- mx[!mx%in%c("geology","surface")]

# build multivariate formula
```
# we also add "lat*lon" as computed covariate
form <- multivariateFormula(ny,c(nx,"I(lat*lon")"),c("geology"))

# split genus dataset
sub <- sample(1:nrow(genus),100,replace=FALSE)
sub_fit <- (1:nrow(genus))[,-sub]

# define family
fam <- rep("poisson",length(ny))

# fit the model
genus.scglr <- scglr(formula=form, data=genus, family=fam, K=4,
                   offset=genus$surface, subset=sub_fit)

# xnew, the design matrix associated to sub-sample used for prediction
# rhs parameters is introduced to take into account that the covariate
# of the formula is composed of two differents sets
xnew <- model.matrix(form, data=genus[sub,], rhs=1:2)[,-1]

# prediction based on the scglr approach
pred.scglr <- multivariatePredictGlm(xnew,family=fam,
                                      beta=genus.scglr$beta,offset=genus$surface[sub])
cor.scglr <- diag(cor(pred.scglr,genus[sub,ny]))
plot(cor.scglr, col="red",ylim=c(-1,1))

# prediction based on classical poisson glm
genus.glm <- multivariateGlm(formula=form, data=genus, family=fam,
                             offset=genus$surface, subset=sub_fit)
coefs <- sapply(genus.glm$coef)

# rhs parameters is introduced to take into account that the covariate
# part of the formula is composed of two differents sets
pred.glm <- multivariatePredictGlm(xnew,family=fam,beta=coefs,
                                   offset=genus$surface[sub])
cor.glm <- diag(cor(pred.glm,genus[sub,ny]))

points(cor.glm, col="blue")

## End(Not run)

pairs.SCGLR  

Pairwise scglr plot on components

Description

Pairwise scglr plot on components

Usage

## S3 method for class 'SCGLR'
pairs(x, ..., nrow = NULL, ncol = NULL, components = NULL)
Arguments

x object of class 'SCGLR', usually a result of running scglr.
nrow number of rows of the grid layout.
ncol number of columns of the grid layout.
components vector of integers selecting components to plot (default is all components).
... optionally, further arguments forwarded to link{plot.SCGLR}.

See Also

For pairs application see examples in plot.SCGLR

Description

SCGLR generic plot

Usage

## S3 method for class 'SCGLR'
plot(x,..., style = getOption("plot.SCGLR"), plane = c(1,
2))

Arguments

x an object from SCGLR class.
style named list of values used to customize the plot (see customize)
plane a size-2 vector (or comma separated string) indicating which components are
plotted (eg: c(1,2) or "1,2").
... optional arguments (see customize).

Value

an object of class ggplot.

Examples

## Not run:
library(SCGLR)

# load sample data
data(genus)

# get variable names from dataset
n <- names(genus)
print.SCGLR

Description

Prints inertia per component and deviance for each Y.

Usage

## S3 method for class 'SCGLR'
print(x, ...)

Arguments

x object of class 'SCGLR', usually a result of running scglr.

... Not used.
scglr  

Function that fits the scglr model

Description
Calculates the components to predict all the dependent variables.

Usage
scglr(formula, data, family, K = 1, size = NULL, offset = NULL, subset = NULL, na.action = na.omit, crit = list(), method = methodSCGLR())

Arguments
- formula: an object of class Formula (or one that can be coerced to that class): a symbolic description of the model to be fitted.
- data: a data frame to be modeled.
- family: a vector of character of the same length as the number of dependent variables: "bernoulli", "binomial", "poisson" or "gaussian" is allowed.
- K: number of components, default is one.
- size: describes the number of trials for the binomial dependent variables. A (number of statistical units * number of binomial dependent variables) matrix is expected.
- offset: used for the poisson dependent variables. A vector or a matrix of size: number of observations * number of Poisson dependent variables is expected.
- subset: an optional vector specifying a subset of observations to be used in the fitting process.
- na.action: a function which indicates what should happen when the data contain NAs. The default is set to na.omit.
- crit: a list of two elements: maxit and tol, describing respectively the maximum number of iterations and the tolerance convergence criterion for the Fisher scoring algorithm. Default is set to 50 and 10e-6 respectively.
- method: Regularization criterion type. Object of class "method.SCGLR" built by methodLPLS for PLS-type approach or methodSR for Structural Relevance.

Value
an object of the SCGLR class.
The function summary (i.e., summary.SCGLR) can be used to obtain or print a summary of the results.
The generic accessor functions coef can be used to extract various useful features of the value returned by scglr.
An object of class "SCGLR" is a list containing following components:
u \text{ matrix of size (number of regressors } \times \text{ number of components), contains the component-loadings, i.e. the coefficients of the regressors in the linear combination giving each component.}

\textit{comp} \text{ matrix of size (number of statistical units } \times \text{ number of components) having the components as column vectors.}

\textit{compr} \text{ matrix of size (number of statistical units } \times \text{ number of components) having the standardized components as column vectors.}

\textit{gamma} \text{ list of length number of dependant variables. Each element is a matrix of coefficients, standard errors, z-values and p-values.}

\textit{beta} \text{ matrix of size (number of regressors } + 1 \text{ (intercept)} \times \text{ number of dependent variables), contains the coefficients of the regression on the original regressors } X.

\textit{lin.pred} \text{ data.frame of size (number of statistical units } \times \text{ number of dependent variables), the fitted linear predictor.}

\textit{xFactors} \text{ data.frame containing the nominal regressors.}

\textit{xNumeric} \text{ data.frame containing the quantitative regressors.}

\textit{inertia} \text{ matrix of size (number of components } \times 2), contains the percentage and cumulative percentage of the overall regressors’ variance, captured by each component.}

\textit{deviance} \text{ vector of length (number of dependent variables), gives the deviance of each } y_k \text{'s GLM on the components.}

References


Examples

```r
## Not run:
library(SCGLR)

data(genus)

# get variable names from dataset
n <- names(genus)
nx <- n[!grep("gen",n)]  # X <- remaining names
ny <- n[grep("gen",n)]   # y <- names that begins with "gen"

# remove "geology" and "surface" from nx
# as surface is offset and we want to use geology as additional covariate
nx <- nx[-grep("gen",nx)]

# build multivariate formula
# we also add "lat*lon" as computed covariate
form <- multivariateFormula(ny,c(nx,"I(lat*lon)"),c("geology"))

# define family
```
scglrCrossVal <- function(formula, data, family, K = 1, nfolds = 5, type = "mspe", size = NULL, offset = NULL, subset = NULL, na.action = na.omit, crit = list(), method = methodSR(), mc.cores = 1)

Arguments

- formula: an object of class "Formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
- data: the data frame to be modeled.
- family: a vector of character of length q specifying the distributions of the responses. Bernoulli, binomial, poisson and gaussian are allowed.
- K: number of components, default is one.
- nfolds: number of folds, default is 5. Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets.
- type: loss function to use for cross-validation. Currently six options are available depending on whether the responses are of the same distribution family. If the responses are all bernoulli distributed, then the prediction performance may be measured through the area under the ROC curve: type = "auc" In any other case one can choose among the following five options ("likelihood", ",aic", ",aicc", ",bic", ",mspe").
- size: specifies the number of trials of the binomial variables included in the model. A (n*qb) matrix is expected for qb binomial variables.
- offset: used for the poisson dependent variables. A vector or a matrix of size: number of observations * number of Poisson dependent variables is expected.
- subset: an optional vector specifying a subset of observations to be used in the fitting process.
na.action a function which indicates what should happen when the data contain NAs. The default is set to the \texttt{na.omit}.

crit a list of two elements: \texttt{maxit} and \texttt{tol}, describing respectively the maximum number of iterations and the tolerance convergence criterion for the Fisher scoring algorithm. Default is set to 50 and 10e-6 respectively.

method Regularization criterion type. Object of class "method.SCGLR" built by \texttt{methodLPLS} for PLS-type approach or \texttt{methodSR} for Structural Relevance.

mc.cores max number of cores to use when using parallelization (Not available in windows yet and strongly discouraged if in interactive mode).

Value
a matrix containing the criterion values for each response (rows) and each number of components (columns).

References

Examples

```r
## Not run:
library(SCGLR)

data(genus)

# get variable names from dataset
data(genus)
n <- names(genus)
ny <- n[grep("^{gen","n] # Y <- names that begins with "gen"
x <- n[grep("^{gen","n] # X <- remaining names

# remove "geology" and "surface" from nx
# as surface is offset and we want to use geology as additional covariate
nx <- nx[!nx%in%c("geology","surface")]

# build multivariate formula
# we also add "lat*lon" as computed covariate
form <- multivariateFormula(ny,c(nx,"I(lat*lon)"),c("geology"))

# define family
fam <- rep("poisson",length(ny))

# cross validation
genus.cv <- scglrCrossVal(formula=form, data=genus, family=fam, K=12,
offset=genus$surface)

# find best K
mean.crit <- t(apply(genus.cv,1,function(x) x/mean(x)))
```
mean.crit <- apply(mean.crit,2,mean)
K.cv <- which.min(mean.crit)-1

#plot(mean.crit, type="l")
## End(Not run)

summary.SCGLR              Summarizing SCGLR fits

Description

Summary method for class "SCGLR".

Usage

## S3 method for class 'SCGLR'
summary(object, ...)

## S3 method for class 'summary.SCGLR'
print(x, digits = 3, cutoff = 1, ...)

Arguments

object      an object of class "SCGLR", usually a result of a call to scglr.
x          an object of class "summary.SCGLR", usually a result of a call to summary.SCGLR.
digits    the number of significant digits to use when printing.
cutoff     print coefficients with pvalue lower than or equal to cutoff (default to 1).
...        Not used.

Value

an object of class "summary.SCGLR".
inertia     inertia per component.
deviance    deviance for each $Y_k$.
rho         squared correlations with numerical covariates.
rho.pred    squared correlations with linear predictors.
coefficients contains the coefficients of the regression on the components.
pvalue      contains the pvalues of the coefficients of the regression on the components.
Index

barplot.SCGLR, 3

coef, 14
critConvergence, 3
customize, 4, 12
genus, 5
ggplot, 12
infoCriterion, 6

methodLPLS, 14, 17
methodLPLS (Methods), 7
Methods, 7
methodSR, 14, 17
methodSR (Methods), 7
multivariateFormula, 7
multivariateGlm (multivariateGlm.fit), 8
multivariateGlm.fit, 8
multivariatePredictGlm, 10

pairs.SCGLR, 4, 11
plot.SCGLR, 3, 4, 12, 12
print.SCGLR, 13
print.summary.SCGLR (summary.SCGLR), 18

scglr, 2, 3, 12, 13, 14, 18
scglr-package, 2
scglrCrossVal, 2, 16
summary, 14
summary.SCGLR, 14, 18