

Package ‘sklarsomega’

May 8, 2020

Type Package

Title Measuring Agreement Using Sklar's Omega Coefficient

Version 2.0-1

Date 2020-05-07

Author John Hughes <drjphughesjr@gmail.com>

Maintainer John Hughes <drjphughesjr@gmail.com>

Imports extraDistr, hash, LaplacesDemon, Matrix, mcmcse, numDeriv,
spam

Suggests lattice, parallel, pbapply

Description Provides tools for applying Sklar's Omega (Hughes, 2018) <arXiv:1803.02734> methodology to nominal, ordinal, interval, or ratio scores. The framework can accommodate any number of units, any number of coders, and missingness; and can be used to measure agreement with a gold standard, intra-coder agreement, and/or inter-coder agreement. Frequentist inference is supported for all levels of measurement. Bayesian inference is supported for interval and ratio scores.

License GPL (>= 2)

RoxygenNote 5.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-05-08 08:40:06 UTC

R topics documented:

baplot	2
build.R	3
cartilage	4
check.colnames	4
influence.sklarsomega	5
logLik.sklarsomega	7
nobs.sklarsomega	7
pcat	8

qcat	9
residuals.sklarsomega	9
simulate.sklarsomega	10
sklars.omega	11
sklars.omega.bayes	14
summary.sklarsomega	17
vcov.sklarsomega	18

Index	20
--------------	-----------

baplot	<i>Produce a Bland-Altman plot.</i>
--------	-------------------------------------

Description

Produce a Bland-Altman plot.

Usage

```
baplot(x, y, pch = 20, col = "black", bg = "black", main = "",
       xlab = "Mean", ylab = "Difference", lwd1 = 1, lwd2 = 1, cex = 1,
       lcol = "black")
```

Arguments

x	the first vector of outcomes.
y	the second vector of outcomes.
pch	the plotting character. Defaults to 20, a bullet.
col	the foreground color. Defaults to black.
bg	the background color. Defaults to black.
main	a title for the plot. Defaults to no title, i.e., "".
xlab	a title for the x axis. Defaults to "Mean".
ylab	a title for the y axis. Defaults to "Difference".
lwd1	the line width for the scatter plot. Defaults to 1.
lwd2	the line width for the confidence band. Defaults to 1.
cex	scaling factor for the scatter plot. Defaults to 1.
lcol	line color for the confidence band. Defaults to black.

Details

This function produces rather customizable Bland-Altman plots, using the [plot](#) and [abline](#) functions. The former is used to create the scatter plot. The latter is used to display the confidence band.

References

Altman, D. G. and Bland, J. M. (1983). Measurement in medicine: The analysis of method comparison studies. *The Statistician*, 307–317.

Nissi, M. J., Mortazavi, S., Hughes, J., Morgan, P., and Ellermann, J. (2015). T2* relaxation time of acetabular and femoral cartilage with and without intra-articular Gd-DTPA2 in patients with femoroacetabular impingement. *American Journal of Roentgenology*, **204**(6), W695.

Examples

```
# Reproduce the plot from Figure 4 of the package vignette.

data(cartilage)
baplot(cartilage$pre, cartilage$post, pch = 21, col = "navy", bg = "darkorange", lwd1 = 2,
        lwd2 = 2, lcol = "navy")
```

build.R

Build a Sklar's Omega correlation matrix.

Description

Build a Sklar's Omega correlation matrix.

Usage

```
build.R(data)
```

Arguments

`data` a matrix of scores. Each row corresponds to a unit, each column a coder.

Details

This function accepts a data matrix and uses its column names to build the appropriate block-diagonal correlation matrix. If gold-standard scores are included, they should be in the first column of the matrix, and that column should be named 'g'. For a given coder, the column name should begin with 'c', and then the coder number and score number should follow, separated by '.' (so that multi-digit numbers can be accommodated). For example, 'c.12.2' denotes the second score for coder 12.

See the package vignette for detailed information regarding the structure of the correlation matrix.

Note that this function is called by `sklars.omega` and so is not a user-level function, per se. We expose the function so that interested users can more easily carry out simulation studies.

Value

`build.R` returns a list comprising two elements.

R	the correlation matrix.
onames	a character vector that contains names for the parameters of the correlation matrix.

See Also

[sklars.omega](#), [check.colnames](#)

cartilage

Cartilage data described in the package vignette.

Description

These data are presented and analyzed in Section 3.2 of the vignette.

Usage

```
data(cartilage)
```

Source

Nissi, M. J., Mortazavi, S., Hughes, J., Morgan, P., and Ellermann, J. (2015). T2* relaxation time of acetabular and femoral cartilage with and without intra-articular Gd-DTPA2 in patients with femoroacetabular impingement. *American Journal of Roentgenology*, **204**(6), W695.

Examples

```
data(cartilage)
```

check.colnames

Check the column names of a Sklar's Omega data matrix for correctness.

Description

Check the column names of a Sklar's Omega data matrix for correctness.

Usage

```
check.colnames(data)
```

Arguments

`data` a matrix of scores. Each row corresponds to a unit, each column a coder.

Details

This function performs a somewhat rudimentary validation of the column names. At most one column may be labeled 'g', and said column must be the first. The only other valid format is 'c.C.S', where 'C' denotes coder number and 'S' denotes the Sth score for coder C (both positive whole numbers). It is up to the user to ensure that the coder and score indices make sense and are ordered correctly, i.e., coders, and scores for a given coder, are numbered consecutively.

Value

check.colnames returns a list comprising two elements.

success logical; if TRUE, the column names passed the test.

cols if success is FALSE, vector cols contains the numbers of the problematic column names.

References

Krippendorff, K. (2013). Computing Krippendorff's alpha-reliability. Technical report, University of Pennsylvania.

Examples

The following data were presented in Krippendorff (2013).

```
data = matrix(c(1,2,3,3,2,1,4,1,2,NA,NA,NA,
               1,2,3,3,2,2,4,1,2,5,NA,3,
               NA,3,3,3,2,3,4,2,2,5,1,NA,
               1,2,3,3,2,4,4,1,2,5,1,NA), 12, 4)
colnames(data) = c("c.1.1", "c.2.1", "c.3.1", "c.4.1")
data
(check.colnames(data))
```

Introduce errors for columns 1 and 4.

```
colnames(data) = c("c.a.1", "c.2.1", "c.3.1", "C.4.1")
(check.colnames(data))
```

The following scenario passes the check but is illogical.

```
colnames(data) = c("g", "c.2.1", "c.1.47", "c.2.1")
(check.colnames(data))
```

influence.sklarsomega *Compute DFBETAs for units and/or coders.*

Description

Compute DFBETAs for units and/or coders.

Usage

```
## S3 method for class 'sklarsomega'
influence(model, units, coders, ...)
```

Arguments

model	a fitted model object.
units	a vector of integers. A DFBETA will be computed for each of the corresponding units.
coders	a vector of integers. A DFBETA will be computed for each of the corresponding coders.
...	additional arguments.

Details

This function computes DFBETAS for one or more units and/or one or more coders.

Value

A list comprising at most two elements.

dfbeta.units	a matrix, the columns of which contain DFBETAS for the units specified via argument units.
dfbeta.coders	a matrix, the columns of which contain DFBETAS for the coders specified via argument coders.

References

Young, D. S. (2017). *Handbook of Regression Methods*. CRC Press.

Krippendorff, K. (2013). Computing Krippendorff's alpha-reliability. Technical report, University of Pennsylvania.

Examples

```
# The following data were presented in Krippendorff (2013).

data = matrix(c(1,2,3,3,2,1,4,1,2,NA,NA,NA,
               1,2,3,3,2,2,4,1,2,5,NA,3,
               NA,3,3,3,2,3,4,2,2,5,1,NA,
               1,2,3,3,2,4,4,1,2,5,1,NA), 12, 4)
colnames(data) = c("c.1.1", "c.2.1", "c.3.1", "c.4.1")
fit = sklars.omega(data, level = "nominal", confint = "none")
summary(fit)
(inf = influence(fit, units = c(6, 11), coders = c(2, 3)))
```

logLik.sklarsomega *Return the maximum of the Sklar's Omega log objective function.*

Description

Return the maximum of the Sklar's Omega log objective function.

Usage

```
## S3 method for class 'sklarsomega'  
logLik(object, ...)
```

Arguments

object a fitted model object.
... additional arguments.

Details

This function extracts the maximum value of the log objective function from a model fit, and is principally intended to be used in computing information criteria.

Value

This function returns an object of class logLik. This is a number with at least one attribute, "df" (degrees of freedom), giving the number of estimated parameters in the model.

See Also

[AIC](#)

nobs.sklarsomega *Return the number of observations for a Sklar's Omega fit.*

Description

Return the number of observations for a Sklar's Omega fit.

Usage

```
## S3 method for class 'sklarsomega'  
nobs(object, ...)
```

Arguments

object a fitted model object.
... additional arguments.

Details

This function extracts the number of observations from a model fit, and is principally intended to be used in computing information criteria.

Value

An integer.

See Also

[AIC](#)

pcat	<i>Compute the cumulative distribution function for a categorical distribution.</i>
------	---

Description

Compute the cumulative distribution function for a categorical distribution.

Usage

```
pcat(q, p)
```

Arguments

q	vector of quantiles.
p	vector of probabilities.

Details

This function uses [dcat](#) to compute the cdf for the categorical distribution with support $1, \dots, K$ and probabilities $p = (p_1, \dots, p_K)'$.

Value

pcat returns a probability or a vector of probabilities, depending on the dimension of q.

qcat *Compute the quantile function for a categorical distribution.*

Description

Compute the quantile function for a categorical distribution.

Usage

```
qcat(pr, p, lower.tail = TRUE, log.pr = FALSE)
```

Arguments

`pr` vector of probabilities.
`p` vector of probabilities.
`lower.tail` logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.
`log.pr` logical; if TRUE, probabilities pr are handled as $\log(pr)$.

Details

This function computes quantiles for the categorical distribution with support $1, \dots, K$ and probabilities $p = (p_1, \dots, p_K)'$.

Value

qcat returns a vector of quantiles.

`residuals.sklarsomega` *Extract model residuals.*

Description

Extract model residuals.

Usage

```
## S3 method for class 'sklarsomega'
residuals(object, ...)
```

Arguments

`object` an object of class sklarsomega, typically the result of a call to `sklars.omega`.
`...` additional arguments.

Details

Although our simulation studies suggest that residuals are not terribly useful in this context, we provide residuals nonetheless. Said residuals are computed by first applying the probability integral transform, then applying the inverse probability integral transform, then pre-multiplying by the inverse of the square root of the (fitted) copula correlation matrix. For nominal or ordinal scores, the distributional transform approximation is used.

Value

A vector of residuals.

References

Nissi, M. J., Mortazavi, S., Hughes, J., Morgan, P., and Ellermann, J. (2015). T2* relaxation time of acetabular and femoral cartilage with and without intra-articular Gd-DTPA2 in patients with femoroacetabular impingement. *American Journal of Roentgenology*, **204**(6), W695.

See Also

[sklars.omega](#)

Examples

```
# Fit a subset of the cartilage data, assuming a Laplace marginal distribution.

data(cartilage)
data = as.matrix(cartilage)[1:100, ]
colnames(data) = c("c.1.1", "c.2.1")
fit = sklars.omega(data, level = "interval", control = list(dist = "laplace"))
summary(fit)
res = residuals(fit)
qqnorm(res, pch = 20)
abline(0, 1, col = "red", lwd = 2)
```

simulate.sklarsomega *Simulate a Sklar's Omega dataset(s).*

Description

Simulate a Sklar's Omega dataset(s).

Usage

```
## S3 method for class 'sklarsomega'
simulate(object, nsim = 1, seed = NULL, ...)
```

Arguments

object	a fitted model object.
nsim	number of datasets to simulate. Defaults to 1.
seed	either NULL or an integer that will be used in a call to <code>set.seed</code> before simulating the response vector(s). If set, the value is saved as the "seed" attribute of the returned value. The default (NULL) will not change the random generator state, and <code>.Random.seed</code> will be returned as the "seed" attribute.
...	additional arguments.

Details

This function simulates one or more responses distributed according to the fitted model.

Value

A data frame having `nsim` columns, each of which contains a simulated response vector. Said data frame has a "seed" attribute, which takes the value of the `seed` argument or the value of `.Random.seed`.

Examples

```
# The following data were presented in Krippendorff (2013).

data = matrix(c(1,2,3,3,2,1,4,1,2,NA,NA,NA,
               1,2,3,3,2,2,4,1,2,5,NA,3,
               NA,3,3,3,2,3,4,2,2,5,1,NA,
               1,2,3,3,2,4,4,1,2,5,1,NA), 12, 4)
colnames(data) = c("c.1.1", "c.2.1", "c.3.1", "c.4.1")
fit = sklars.omega(data, level = "nominal", confint = "none")
summary(fit)

# Simulate three datasets from the fitted model, and then
# display the first dataset in matrix form.

sim = simulate(fit, nsim = 3, seed = 42)
data.sim = t(fit$data)
data.sim[! is.na(data.sim)] = sim[, 1]
data.sim = t(data.sim)
data.sim
```

sklars.omega

Apply Sklar's Omega.

Description

Apply Sklar's Omega.

Usage

```
sklars.omega(data, level = c("nominal", "ordinal", "interval", "ratio"),
  confint = c("none", "bootstrap", "asymptotic"), verbose = FALSE,
  control = list())
```

Arguments

data	a matrix of scores. Each row corresponds to a unit, each column a coder. The columns must be named appropriately so that the correct copula correlation matrix can be constructed. See build.R for details regarding column naming.
level	the level of measurement, one of "nominal", "ordinal", "interval", or "ratio".
confint	the method for computing confidence intervals, one of "none", "bootstrap", or "asymptotic".
verbose	logical; if TRUE, various messages are printed to the console.
control	a list of control parameters.
bootit	the size of the (parametric) bootstrap sample. This applies when <code>confint = "bootstrap"</code> , or when <code>confint = "asymptotic"</code> and <code>level = "nominal"</code> or <code>level = "ordinal"</code> . Defaults to 1,000.
dist	when <code>level = "interval"</code> , one of "gaussian", "laplace", "t", "gamma", or "empirical"; when <code>level = "ratio"</code> , one of "beta" or "kumaraswamy".
nodes	the desired number of nodes in the cluster.
parallel	logical; if TRUE (the default), bootstrapping is done in parallel.
type	one of the supported cluster types for makeCluster . Defaults to "SOCK".

Details

This is the package's flagship function. It applies the Sklar's Omega methodology to nominal, ordinal, interval, or ratio outcomes, and, if desired, produces confidence intervals. Parallel computing is supported, when applicable, and other measures (e.g., sparse matrix operations) are taken in the interest of computational efficiency.

If the level of measurement is nominal or ordinal, the scores (which must take values in $1, \dots, K$) are assumed to share a common categorical marginal distribution. If K is less than 5, a composite marginal likelihood (CML) approach is used. If K is greater than or equal to 5, the distributional transform (DT) approximation is used. In either case, two types of confidence interval are available: bootstrap or asymptotic. See the package vignette for details.

If the level of measurement is interval or ratio, control parameter `dist` must be used to select a marginal distribution from among "gaussian", "laplace", "t", "gamma", and "empirical", or from among "beta" or "kumaraswamy", respectively. The ML method is used unless `dist = "empirical"`, in which case conditional maximum likelihood is used, i.e., the copula parameters are estimated conditional on the sample distribution function of the scores. For the ML method, both bootstrap and asymptotic confidence intervals are available. When `dist = "empirical"`, only bootstrap intervals are available.

When applicable, functions of appropriate sample quantities are used as starting values for marginal parameters, regardless of the level of measurement. Details are provided in the package vignettes.

Value

Function `sklars.omega` returns an object of class `"sklarsomega"`, which is a list comprising the following elements.

AIC	the value of AIC for the fit, if <code>level = "interval"</code> and <code>dist != "empirical"</code> , or if <code>level = "ratio"</code> .
BIC	the value of BIC for the fit, if <code>level = "interval"</code> and <code>dist != "empirical"</code> , or if <code>level = "ratio"</code> .
<code>boot.sample</code>	when applicable, the bootstrap sample.
<code>call</code>	the matched call.
<code>coefficients</code>	a named vector of parameter estimates.
<code>confint</code>	the value of argument <code>confint</code> .
<code>control</code>	the list of control parameters.
<code>convergence</code>	unless optimization failed, the value of convergence returned by <code>optim</code> .
<code>cov.hat</code>	if <code>confint = "asymptotic"</code> , the estimate of the covariance matrix of the parameter estimator.
<code>data</code>	the matrix of scores, perhaps altered to remove rows (units) containing fewer than two scores.
<code>iter</code>	if optimization converged, the value of <code>iter</code> returned by <code>optim</code> .
<code>level</code>	the level of measurement.
<code>message</code>	the value of message returned by <code>optim</code> .
<code>method</code>	the approach to inference, one of <code>"CML"</code> , <code>"DT"</code> , <code>"ML"</code> , or <code>"SMP"</code> (semiparametric).
<code>mpar</code>	the number of marginal parameters.
<code>npar</code>	the total number of parameters.
<code>R</code>	the initial value of the copula correlation matrix.
<code>R.hat</code>	the estimated value of the copula correlation matrix.
<code>residuals</code>	the residuals.
<code>root.R.hat</code>	a square root of the estimated copula correlation matrix. This is used for simulation and to compute the residuals.
<code>value</code>	the minimum of the log objective function.
<code>verbose</code>	the value of argument <code>verbose</code> .
<code>y</code>	the scores as a vector, perhaps altered to remove rows (units) containing fewer than two scores.

References

- Hughes, J. (2018). Sklar's Omega: A Gaussian copula-based framework for assessing agreement. *ArXiv e-prints*, March.
- Nissi, M. J., Mortazavi, S., Hughes, J., Morgan, P., and Ellermann, J. (2015). T2* relaxation time of acetabular and femoral cartilage with and without intra-articular Gd-DTPA2 in patients with femoroacetabular impingement. *American Journal of Roentgenology*, **204**(6), W695.

Examples

```

# The following data were presented in Krippendorff (2013).

data = matrix(c(1,2,3,3,2,1,4,1,2,NA,NA,NA,
               1,2,3,3,2,2,4,1,2,5,NA,3,
               NA,3,3,3,2,3,4,2,2,5,1,NA,
               1,2,3,3,2,4,4,1,2,5,1,NA), 12, 4)
colnames(data) = c("c.1.1", "c.2.1", "c.3.1", "c.4.1")
data

# Compute asymptotic confidence intervals. Since the distributional transform
# approximation is used, the asymptotic covariance matrix has a sandwich form.
# We use a bootstrap estimator of the "filling" (the variance of the score).
# Obtain a bootstrap sample of size 1,000. Do the bootstrap in parallel, using
# all but one of the available cores. Since we set 'verbose' equal to TRUE, a
# progress bar is displayed during the bootstrap.

set.seed(12)
fit = sklars.omega(data, level = "nominal", confint = "asymptotic", verbose = TRUE,
                  control = list(bootit = 1000, parallel = TRUE,
                                nodes = parallel::detectCores() - 1))
summary(fit)

# Fit a subset of the cartilage data, assuming a Laplace marginal distribution. Compute
# confidence intervals in the usual ML way (observed information matrix).

data(cartilage)
data = as.matrix(cartilage)[1:100, ]
colnames(data) = c("c.1.1", "c.2.1")
fit = sklars.omega(data, level = "interval", confint = "asymptotic",
                  control = list(dist = "laplace"))
summary(fit)
vcov(fit)

# Now assume a t marginal distribution.

fit = sklars.omega(data, level = "interval", confint = "asymptotic",
                  control = list(dist = "t"))
summary(fit)
vcov(fit)

```

sklars.omega.bayes *Do Bayesian inference for Sklar's Omega.*

Description

Do Bayesian inference for Sklar's Omega.

Usage

```
sklars.omega.bayes(data, level = c("interval", "ratio"), verbose = FALSE,
  control = list())
```

Arguments

<code>data</code>	a matrix of scores. Each row corresponds to a unit, each column a coder. The columns must be named appropriately so that the correct copula correlation matrix can be constructed. See build.R for details regarding column naming.
<code>level</code>	the level of measurement, either "interval" or "ratio".
<code>verbose</code>	logical; if TRUE, various messages are printed to the console.
<code>control</code>	a list of control parameters.

dist when `level = "interval"`, one of "gaussian", "laplace", "t", or "gamma"; when `level = "ratio"`, one of "beta" or "kumaraswamy".

minit the minimum sample size. This should be large enough to permit accurate estimation of Monte Carlo standard errors. The default is 1,000.

maxit the maximum sample size. Sampling from the posterior terminates when all estimated coefficients of variation are smaller than `tol` or when `maxit` samples have been drawn, whichever happens first. The default is 10,000.

sigma.1 the proposal standard deviation for the first marginal parameter. Defaults to 0.1.

sigma.2 the proposal standard deviation for the second marginal parameter. Defaults to 0.1.

sigma.omega a vector of proposal standard deviations for the parameters of the copula correlation matrix. These default to 0.1.

tol a tolerance. If all estimated coefficients of variation are smaller than `tol`, no more samples are drawn from the posterior. The default value is 0.1.

Details

This function does MCMC for Bayesian inference for interval or ratio scores.

Control parameter `dist` must be used to select a marginal distribution from among "gaussian", "laplace", "t", and "gamma" (for interval scores), or from among "beta" or "kumaraswamy" (for ratio scores).

Details regarding prior distributions and sampling are provided in the second package vignette.

Value

Function `sklars.omega.bayes` returns an object of class "sklarsomega", which is a list comprising the following elements.

<code>accept</code>	a vector of acceptance rates.
<code>DIC</code>	the value of DIC for the fit.
<code>call</code>	the matched call.
<code>coefficients</code>	a named vector of parameter estimates.

control	the list of control parameters.
data	the matrix of scores, perhaps altered to remove rows (units) containing fewer than two scores.
iter	the number of posterior samples that were drawn.
level	the level of measurement.
mcse	a vector of Monte Carlo standard errors.
method	always equal to "Bayesian" for this function.
mpar	the number of marginal parameters.
npar	the total number of parameters.
R	the initial value of the copula correlation matrix.
R.hat	the estimated value of the copula correlation matrix.
residuals	the residuals.
root.R.hat	a square root of the estimated copula correlation matrix. This is used for simulation and to compute the residuals.
samples	the posterior samples.
verbose	the value of argument verbose.
y	the scores as a vector, perhaps altered to remove rows (units) containing fewer than two scores.

References

- Hughes, J. (2018). Sklar's Omega: A Gaussian copula-based framework for assessing agreement. *ArXiv e-prints*, March.
- Nissi, M. J., Mortazavi, S., Hughes, J., Morgan, P., and Ellermann, J. (2015). T2* relaxation time of acetabular and femoral cartilage with and without intra-articular Gd-DTPA2 in patients with femoroacetabular impingement. *American Journal of Roentgenology*, **204**(6), W695.

Examples

```
# Fit a subset of the cartilage data, assuming a Laplace marginal distribution. Compute
# 95% HPD intervals. Show the acceptance rates for the three parameters.

data(cartilage)
data = as.matrix(cartilage)[1:100, ]
colnames(data) = c("c.1.1", "c.2.1")
set.seed(111111)
fit1 = sklars.omega.bayes(data, verbose = FALSE,
                          control = list(dist = "laplace", minit = 1000, maxit = 5000, tol = 0.01,
                                         sigma.1 = 1, sigma.2 = 0.1, sigma.omega = 0.2))

summary(fit1)
fit1$accept

# Now assume a t marginal distribution.

set.seed(4565)
```



```
fit2 = sklars.omega.bayes(data, verbose = FALSE,
                          control = list(dist = "t", minit = 1000, maxit = 5000, tol = 0.01,
                                          sigma.1 = 0.2, sigma.2 = 2, sigma.omega = 0.2))
summary(fit2)
fit2$accept
```

summary.sklarsomega *Print a summary of a Sklar's Omega fit.*

Description

Print a summary of a Sklar's Omega fit.

Usage

```
## S3 method for class 'sklarsomega'
summary(object, alpha = 0.05, digits = 4, ...)
```

Arguments

object	an object of class sklarsomega, the result of a call to sklars.omega .
alpha	the significance level for the confidence intervals. The default is 0.05.
digits	the number of significant digits to display. The default is 4.
...	additional arguments.

Details

Unless optimization of the objective function failed, this function prints a summary of the fit. First, the value of the objective function at its maximum is displayed, along with the number of iterations required to find the maximum. Then the values of the control parameters (defaults and/or values supplied in the call) are printed. Then a table of estimates is shown. If applicable, the table includes confidence intervals. Finally, the values of [AIC](#) and [BIC](#) are displayed (if the scores were interval and inference was parametric).

References

Nissi, M. J., Mortazavi, S., Hughes, J., Morgan, P., and Ellermann, J. (2015). T2* relaxation time of acetabular and femoral cartilage with and without intra-articular Gd-DTPA2 in patients with femoroacetabular impingement. *American Journal of Roentgenology*, **204**(6), W695.

See Also

[sklars.omega](#)

Examples

```
# Fit a subset of the cartilage data, assuming a Laplace marginal distribution. Compute
# confidence intervals in the usual ML way (observed information matrix).
```

```
data(cartilage)
data = as.matrix(cartilage)[1:100, ]
colnames(data) = c("c.1.1", "c.2.1")
fit = sklars.omega(data, level = "interval", confint = "asymptotic",
                  control = list(dist = "laplace"))
summary(fit)
vcov(fit)
```

vcov.sklarsomega

Compute an estimated covariance matrix for a Sklar's Omega fit.

Description

Compute an estimated covariance matrix for a Sklar's Omega fit.

Usage

```
## S3 method for class 'sklarsomega'
vcov(object, ...)
```

Arguments

```
object      a fitted model object.
...         additional arguments.
```

Details

See the package vignette for detailed information regarding covariance estimation for Sklar's Omega.

Value

A matrix of estimated variances and covariances for the parameter estimator. This should have row and column names corresponding to the parameter names given by the `coef` method. Note that a call to this function will result in an error if `sklars.omega` was called with argument `confint` equal to "none", or if optimization failed.

References

Nissi, M. J., Mortazavi, S., Hughes, J., Morgan, P., and Ellermann, J. (2015). T2* relaxation time of acetabular and femoral cartilage with and without intra-articular Gd-DTPA2 in patients with femoroacetabular impingement. *American Journal of Roentgenology*, **204**(6), W695.

Examples

```
# Fit a subset of the cartilage data, assuming a Laplace marginal distribution. Compute  
# confidence intervals in the usual ML way (observed information matrix).
```

```
data(cartilage)  
data = as.matrix(cartilage)[1:100, ]  
colnames(data) = c("c.1.1", "c.2.1")  
fit = sklars.omega(data, level = "interval", confint = "asymptotic",  
                  control = list(dist = "laplace"))  
summary(fit)  
vcov(fit)
```

Index

*Topic **datasets**

cartilage, [4](#)
.Random.seed, [11](#)

abline, [2](#)
AIC, [7](#), [8](#), [17](#)

baplot, [2](#)
build.R, [3](#), [12](#), [15](#)

cartilage, [4](#)
check.colnames, [4](#), [4](#)
coef, [18](#)

dcat, [8](#)

influence.sklarsomega, [5](#)

logLik.sklarsomega, [7](#)

makeCluster, [12](#)

nobs.sklarsomega, [7](#)

optim, [13](#)

pcat, [8](#)
plot, [2](#)

qcat, [9](#)

residuals.sklarsomega, [9](#)

set.seed, [11](#)
simulate.sklarsomega, [10](#)
sklars.omega, [3](#), [4](#), [9](#), [10](#), [11](#), [17](#), [18](#)
sklars.omega.bayes, [14](#)
summary.sklarsomega, [17](#)

vcov.sklarsomega, [18](#)