

Package ‘sklarsomega’

July 23, 2025

Type Package

Title Measuring Agreement Using Sklar's Omega Coefficient

Version 3.0-2

Date 2023-04-04

Author John Hughes <drjphughesjr@gmail.com>

Maintainer John Hughes <drjphughesjr@gmail.com>

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

Suggests lattice, parallel, pbapply

Description

Provides tools for applying Sklar's Omega (Hughes, 2018) <[doi:10.48550/arXiv.1803.02734](https://doi.org/10.48550/arXiv.1803.02734)> methodology to nominal scores, ordinal scores, percentages, counts, amounts (i.e., non-negative real numbers), and balances (i.e., any real number). 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 continuous scores only.

License GPL (>= 2)

RoxygenNote 7.2.1

NeedsCompilation yes

Encoding UTF-8

Repository CRAN

Date/Publication 2023-04-04 16:50:02 UTC

Contents

baplot	2
build.R	3
cartilage	4
check.colnames	5
confint.sklarsomega	6

influence.sklarsomega	7
logLik.sklarsomega	8
nobs.sklarsomega	9
pcat	10
qcat	10
residuals.sklarsomega	11
simulate.sklarsomega	12
sklars.omega	13
sklars.omega.bayes	16
summary.sklarsomega	18
vcov.sklarsomega	19

Index	21
--------------	-----------

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

```
# Produce a Bland-Altman plot for the cartilage dataset.

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

build.R	<i>Build a Sklar's Omega correlation matrix.</i>
---------	--

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.

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.

<code>R</code>	the correlation matrix.
<code>onames</code>	a character vector that contains names for the parameters of the correlation matrix.

See Also

`sklars.omega`, `check.colnames`

<code>cartilage</code>	<i>Data from an MRI study of hip cartilage in femoroacetabular impingement.</i>
------------------------	---

Description

This data frame has exactly two columns. The first column contains raw T2* values, the second column contrast-enhanced T2* values.

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	<i>Check the column names of a Sklar's Omega data matrix for correctness.</i>
----------------	---

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.nom = 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.nom) = c("c.1.1", "c.2.1", "c.3.1", "c.4.1")
data.nom
(check.colnames(data.nom))
```

```
# Introduce errors for columns 1 and 4.

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

# The following scenario passes the check but is illogical.

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

confint.sklarsomega	<i>Compute confidence/credible intervals for Sklar's Omega.</i>
---------------------	---

Description

Compute confidence/credible intervals for Sklar's Omega.

Usage

```
## S3 method for class 'sklarsomega'
confint(object, parm, level = 0.95, ...)
```

Arguments

object	an object of class "sklarsomega", the result of a call to sklars.omega .
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the desired confidence level for the interval. The default value is 0.95.
...	additional arguments. These are passed to quantile if confint is equal to "bootstrap".

Details

This function computes confidence/credible intervals for a Sklar's Omega fit.

Value

A vector with entries giving lower and upper confidence limits. These will be labelled as $(1 - \text{level}) / 2$ and $1 - (1 - \text{level}) / 2$.

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). Note that
# calling function sklars.omega with confint = bootstrap will lead to bootstrap sampling,
# in which case confidence intervals will be bootstrap intervals.
```

```
data(cartilage)
data.cart = as.matrix(cartilage)[1:100, ]
colnames(data.cart) = c("c.1.1", "c.2.1")
fit.lap = sklars.omega(data.cart, level = "balance", confint = "asymptotic",
                      control = list(dist = "laplace"))
summary(fit.lap)
confint(fit.lap, level = 0.99)
```

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). After
# analyzing the data and displaying a summary of the fit, compute
# and display DFBETAs for unit 11 and coders 2 and 3.

data.nom = 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.nom) = c("c.1.1", "c.2.1", "c.3.1", "c.4.1")
fit.nom = sklars.omega(data.nom, level = "nominal", confint = "none")
summary(fit.nom)
(inf = influence(fit.nom, units = 11, coders = c(2, 3)))
```

logLik.sklarsomega	<i>Return the maximum of the Sklar's Omega log objective function.</i>
--------------------	--

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	<i>Return the number of observations for a Sklar's Omega fit.</i>
------------------	---

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

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 residuals may not be 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.
# Produce a normal probability plot of the residuals, and overlay the line y = x.

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

simulate.sklarsomega	<i>Simulate a Sklar's Omega dataset(s).</i>
----------------------	---

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 set.seed 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 .Random.seed 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

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

data(cartilage)
data.cart = as.matrix(cartilage)[1:100, ]
colnames(data.cart) = c("c.1.1", "c.2.1")
fit.lap = sklars.omega(data.cart, level = "balance", confint = "none",
                      control = list(dist = "laplace"))
summary(fit.lap)

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

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

sklars.omega

Apply Sklar's Omega.

Description

Apply Sklar's Omega.

Usage

```
sklars.omega(
  data,
  level = c("nominal", "ordinal", "count", "percentage", "amount", "balance"),
  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", "count", "amount", "balance", or "percentage".
confint	the method for computing confidence intervals, one of "none", "bootstrap", or "asymptotic".
verbose	logical; if TRUE, various messages may be printed to the console.
control	a list of control parameters.

bootit the size of the (parametric) bootstrap sample. This applies when `confint = "bootstrap"` or when `confint = "asymptotic"` and `level = "count"`. Defaults to 1,000.

dist when `level = "balance"`, one of "gaussian", "laplace", "t", or "empirical"; when `level = "amount"`, one of "gamma" or "empirical"; when `level = "percentage"`, one of "beta" or "kumaraswamy"; when `level = "count"`, one of "poisson" or "negbinomial".

nodes the desired number of nodes in the cluster.

parallel logical; if TRUE (the default is FALSE), 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, count, amount, balance, or percentage 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. A composite marginal likelihood (CML) approach is used for categorical scores. Only parametric bootstrap intervals are supported for categorical outcomes since sandwich estimation tends to lead to inflated standard errors.

If the scores are counts, control parameter `dist` must be used to select a marginal distribution of "poisson" or "negbinomial". For counts a distributional transform (DT) approximation of the likelihood is employed. Either bootstrap or sandwich intervals are available: `confint = "bootstrap"` or `confint = "asymptotic"`.

If the level of measurement is balance or amount or percentage, control parameter `dist` must be used to select a marginal distribution from among "gaussian", "laplace", "t", and "empirical"; or from among "gamma" and "empirical"; or from among "beta" or "kumaraswamy", respectively. The method of maximum likelihood (ML) 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.

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 = "amount"</code> or <code>level = "balance"</code> and <code>dist != "empirical"</code> , or if <code>level = "percentage"</code> .
BIC	the value of BIC for the fit, if <code>level = "amount"</code> or <code>level = "balance"</code> and <code>dist != "empirical"</code> , or if <code>level = "percentage"</code> .
<code>boot.sample</code>	when applicable, the bootstrap sample.

call	the matched call.
coefficients	a named vector of parameter estimates.
confint	the value of argument confint.
control	the list of control parameters.
convergence	unless optimization failed, the value of convergence returned by <code>optim</code> or <code>hjkb</code> .
cov.hat	if confint = "asymptotic", the estimate of the covariance matrix of the parameter estimator.
data	the matrix of scores, perhaps altered to remove rows (units) containing fewer than two scores.
iter	if optimization converged, the number of iterations required to optimize the objective function.
level	the level of measurement.
message	if applicable, the value of message returned by <code>optim</code> .
method	the approach to inference, one of "CML", "DT", "ML", or "SMP" (semiparametric).
mpar	the number of marginal parameters.
npar	the total number of parameters.
optim.method	the method used to optimize the objective function. The L-BFGS-B method is attempted first. If L-BFGS-B fails, a second attempt is made using the bounded Hooke-Jeeves algorithm.
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.
value	the minimum of the log objective function.
verbose	the value of argument verbose.
y	the scores as a vector, perhaps altered to remove rows (units) containing only one score.

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
# confidence intervals in the usual ML way (observed information matrix).

data(cartilage)
```

```

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

# Now assume a noncentral t marginal distribution.

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

```

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("amount", "balance", "percentage"),
  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, either "amount" or "balance" or "percentage".
verbose	logical; if TRUE, various messages are printed to the console.
control	a list of control parameters.

dist when level = "balance", one of "gaussian", "laplace", "t"; when level = "amount", "gamma"; when level = "percentage", 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 continuous scores.

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

Details regarding prior distributions and sampling are provided in the 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.
<code>control</code>	the list of control parameters.
<code>data</code>	the matrix of scores, perhaps altered to remove rows (units) containing fewer than two scores.
<code>iter</code>	the number of posterior samples that were drawn.
<code>level</code>	the level of measurement.
<code>mcse</code>	a vector of Monte Carlo standard errors.
<code>method</code>	always equal to "Bayesian" for this function.
<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>samples</code>	the posterior samples.
<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

```
# 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, level = "balance", 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 noncentral t marginal distribution.

set.seed(4565)
fit2 = sklars.omega.bayes(data, level = "balance", 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 are continuous and inference is 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). Note that
# using confint = bootstrap leads to bootstrap sampling and bootstrap intervals.

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

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). Also display
# the observed information matrix. Note that using confint = bootstrap leads to bootstrap
# sampling, in which case vcov returns the sample covariance matrix for the bootstrap
# sample.

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

summary(fit.lap)
vcov(fit.lap)
```

Index

* datasets

- cartilage, [4](#)
- .Random.seed, [12](#)
- abline, [3](#)
- AIC, [9](#), [19](#)
- baplot, [2](#)
- build.R, [3](#), [13](#), [16](#)
- cartilage, [4](#)
- check.colnames, [4](#), [5](#)
- coef, [20](#)
- confint.skarsomega, [6](#)
- dcat, [10](#)
- hjkb, [15](#)
- influence.skarsomega, [7](#)
- logLik.skarsomega, [8](#)
- makeCluster, [14](#)
- nobs.skarsomega, [9](#)
- optim, [15](#)
- pcat, [10](#)
- plot, [3](#)
- qcat, [10](#)
- quantile, [6](#)
- residuals.skarsomega, [11](#)
- set.seed, [12](#)
- simulate.skarsomega, [12](#)
- sklars.omega, [4](#), [6](#), [7](#), [11](#), [13](#), [18–20](#)
- sklars.omega.bayes, [16](#)
- summary.skarsomega, [18](#)
- vcov.skarsomega, [19](#)