Package ‘eba’

September 25, 2018

Version 1.8-0
Date 2018-09-25
Title Elimination-by-Aspects Models
Depends R (>= 2.15.0), stats, graphics
Imports nlme
Description Fitting and testing multi-attribute probabilistic choice
models, especially the Bradley-Terry-Luce (BTL) model (Bradley &
Terry, 1952 <doi:10.1093/biomet/39.3-4.324>; Luce, 1959),
elimination-by-aspects (EBA) models (Tversky, 1972 <doi:10.1037/h0032955>),
and preference tree (Pretree) models (Tversky & Sattath, 1979
License GPL (>= 2)
URL http://homepages.uni-tuebingen.de/florian.wickelmaier
NeedsCompilation no
Author Florian Wickelmaier [aut, cre]
Maintainer Florian Wickelmaier <wickelmaier@web.de>
Repository CRAN
Date/Publication 2018-09-25 13:20:06 UTC

R topics documented:

balanced.pcdesign ......................................................... 2
boot ................................................................. 3
celebrities ............................................................ 4
circular ............................................................ 5
cov.u ............................................................... 7
drugrisk ........................................................... 7
eba ............................................................... 9
eba.order ......................................................... 11
group.test ....................................................... 13
heaviness ......................................................... 15
balanced.pcdesign

Description

Creates a (completely) balanced paired-comparison design.

Usage

balanced.pcdesign(nstimuli)

Arguments

nstimuli number of stimuli in the paired-comparison design

Details

When nstimuli is odd, the presentation order is completely balanced, that is any given stimulus appears an equal number of times as the first and second member of a pair. When nstimuli is even, the presentation order is balanced as much as possible.

Subjects should be equally assigned to listA and listB for the purpose of balancing the within-pair presentation order across a sample of subjects.

Pairs should be re-randomized for each subject.
Value

pairs a character array holding the balanced pairs; see David (1988) for details on how it is constructed

listA the vector pairs in the original within-pair order

listB the vector of pairs in the inverted within-pair order

References


See Also

pcX, eba.

Examples

```r
## Create balanced design for 6 stimuli
bp <- balanced.pcdesign(6)

## Replicate each within-pair order 10 times and re-randomize
cbind(replicate(10, sample(bp$listA)), replicate(10, sample(bp$listB)))
```

Description

Performs a bootstrap by resampling the individual data matrices.

Usage

```r
boot(D, R = 100, A = 1:i, s = rep(1/J, J), constrained = TRUE)
```

Arguments

- `D`: a 3d array consisting of the individual paired comparison matrices
- `R`: the number of bootstrap samples
- `A`: a list of vectors consisting of the stimulus aspects; the default is `1:i`, where `i` is the number of stimuli
- `s`: the starting vector with default `1/J` for all parameters, where `J` is the number of parameters
- `constrained`: logical, if `TRUE` (default), parameters are constrained to be positive

Details

The bootstrap functions `eba.boot.constrained` and `eba.boot` are called automatically by `boot`. The code is experimental and may change in the future.
celebrities

Description

This data set provides the absolute choice frequencies of 234 subjects choosing between pairs of nine celebrities. L. B. Johnson (LBJ), Harold Wilson (HW), and Charles De Gaulle (CDG) are politicians; Johnny Unitas (JU), Carl Yastrzemski (CY), and A. J. Foyt (AJF) are athletes; Brigitte Bardot (BB), Elizabeth Taylor (ET), and Sophia Loren (SL) are female movie stars. Subjects were instructed to choose the person with whom they would rather spend an hour of discussion. Row stimuli are chosen over column stimuli.

Usage

data(celebrities)

Format

A square data frame with a diagonal of zeros.

Source


Examples

data(celebrities)
celebrities["LBJ", "HW"] # 159 subjects choose LBJ over HW
Circular Triads (Intransitive Cycles)

Description

Number of circular triads and coefficient of consistency.

Usage

circular(mat, alternative = c("two.sided", "less", "greater"),
  exact = NULL, correct = TRUE, simulate.p.value = FALSE,
  nsim = 2000)

Arguments

- **mat**: a square matrix or a data frame consisting of (individual) binary choice data; row stimuli are chosen over column stimuli
- **alternative**: a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "less" or "greater"
- **exact**: a logical indicating whether an exact p-value should be computed
- **correct**: a logical indicating whether to apply continuity correction in the chi-square approximation for the p-value
- **simulate.p.value**: a logical indicating whether to compute p-values by Monte Carlo simulation
- **nsim**: an integer specifying the number of replicates used in the Monte Carlo test

Details

Kendall’s coefficient of consistency,

\[
zeta = 1 - T/T_{\text{max}},
\]

lies between one (perfect consistency) and zero, where \(T\) is the observed number of circular triads, and the maximum possible number of circular triads is \(T_{\text{max}} = n(n^2 - 4)/24\), if \(n\) is even, and \(T_{\text{max}} = n(n^2 - 1)/24\) else, and \(n\) is the number of stimuli or objects being judged. For details, see Kendall and Babington Smith (1940) and David (1988).

Kendall (1962) discusses a test of the hypothesis that the number of circular triads \(T\) is different (smaller or greater) than expected when choosing randomly. For small \(n\), an exact p-value is computed, based on the exact distributions listed in Alway (1962) and in Kendall (1962). Otherwise, an approximate chi-square test is computed. In this test, the sampling distribution is measured from lower to higher values of \(T\), so that the probability that \(T\) will be exceeded is the complement of the probability for \(\chi^2\). The chi-square approximation may be incorrect if \(n < 8\) and is only available for \(n > 4\).
Value

- `T` number of circular triads
- `T.max` maximum possible number of circular triads
- `T.exp` expected number of circular triads \( E(T) \) when choices are totally random
- `zeta` Kendall’s coefficient of consistency
- `chi2, df, correct` the chi-square statistic and degrees of freedom for the approximate test, and whether continuity correction has been applied
- `p.value` the p-value for the test (see Details)
- `simulate.p.value, nsim` whether the p-value is based on simulations, number of simulation runs

References


See Also

eba, strans, kendall.u.

Examples

```r
# A dog's preferences for six samples of food
# (Kendall and Babington Smith, 1940, p. 326)
dog <- matrix(c(0, 1, 1, 0, 1, 1,
               0, 0, 0, 1, 1, 0,
               0, 1, 0, 1, 1, 1,
               1, 0, 0, 0, 0, 0,
               0, 0, 0, 1, 0, 1,
               0, 1, 0, 1, 0, 0), 6, 6, byrow=TRUE)
dimnames(dog) <- setNames(rep(list(c("meat", "biscuit", "chocolate",
                                    "apple", "pear", "cheese")), 2), c(">", "<"))
circular(dog, alternative="less") # moderate consistency
subset(strans(dog)$violdf, lwst) # list circular triads
```
Covariance Matrix of the EBA Utility Scale

Description

Computes the (normalized) covariance matrix of the utility scale from the covariance matrix of elimination-by-aspects (EBA) model parameters.

Usage

cov.u(object, norm = "sum")

Arguments

object an object of class eba, typically the result of a call to eba
norm either sum (default), a number from 1 to number of stimuli, or NULL; see uscale for details

Details

The additivity rule for covariances \( \text{cov}(x + y, z) = \text{cov}(x, z) + \text{cov}(y, z) \) is used for the computations.

If \( \text{norm} \) is not NULL, the unnormalized covariance matrix is transformed using \( a^2 \text{cov}(u) \), where the constant \( a \) results from the type of normalization applied.

Value

The (normalized) covariance matrix of the utility scale.

See Also

uscale, eba, wald.test.

drugrisk Perceived Health Risk of Drugs

Description

In summer 2007, a survey was conducted at the Department of Psychology, University of Tuebingen. Hundred and ninety-two participants were presented with all 15 unordered pairs of the names of six drugs or substances and asked to choose the drug they judged as more dangerous for their health. The six drugs were alcohol (alc), tobacco (tob), cannabis (can), ecstasy (ecs), heroine (her), and cocaine (coc). Choice frequencies were aggregated in four groups defined by gender and age.
Usage

data(drugrisk)

Format

A 3d array consisting of four square matrices of choice frequencies (row drugs are judged over column drugs):

drugrisk[, , group = "female30"] holds the choices of the 48 female participants up to 30 years of age.
drugrisk[, , group = "female31"] holds the choices of the 48 female participants from 31 years of age.
drugrisk[, , group = "male30"] holds the choices of the 48 male participants up to 30 years of age.
drugrisk[, , group = "male31"] holds the choices of the 48 male participants from 31 years of age.

Source


Examples

data(drugrisk)

### Bradley-Terry-Luce (BTL) model
btl <- eba(drugrisk[, , group = "male30"])

### Elimination-by-aspects (EBA) model, 1 additional aspect
A1 <- list(c(1), c(2,7), c(3,7), c(4,7), c(5,7), c(6,7))
eba1 <- eba(drugrisk[, , group = "male30"], A1)

### EBA model, 2 additional aspects
A2 <- list(c(1), c(2,7), c(3,7), c(4,7,8), c(5,7,8), c(6,7,8))
eba2 <- eba(drugrisk[, , group = "male30"], A2)

### Model selection
anova(btl, eba1, eba2)

### Utility scale values (BTL for females, EBA for males)
dotchart(cbind(
  apply(drugrisk[,1:2], 3, function(x) uscale(eba(x), norm=1, log=TRUE)),
  apply(drugrisk[,3:4], 3, function(x) uscale(eba(x, A2), norm=1, log=TRUE))
), xlab="log[ Utility scale value (BTL and EBA models) ]",
  main="Perceived health risk of drugs")
abline(v=0, col="gray")
mtext("(Wickelmaier, 2008)", line=.5)
Description

Fits a (multi-attribute) probabilistic choice model by maximum likelihood.

Usage

`eba(M, A = 1:I, s = rep(1/J, J), constrained = TRUE)`

`OptiPt(M, A = 1:I, s = rep(1/J, J), constrained = TRUE)`

```r
## S3 method for class 'eba'
summary(object, ...)
```

```r
## S3 method for class 'eba'
anova(object, ..., test = c("Chisq", "none"))
```

Arguments

- `M` a square matrix or a data frame consisting of absolute choice frequencies; row stimuli are chosen over column stimuli
- `A` a list of vectors consisting of the stimulus aspects; the default is `1:I`, where `I` is the number of stimuli
- `s` the starting vector with default `1/J` for all parameters, where `J` is the number of parameters
- `constrained` logical, if `TRUE` (default), parameters are constrained to be positive
- `object` an object of class `eba`, typically the result of a call to `eba`
- `test` should the p-values of the chi-square distributions be reported?
- `...` additional arguments; none are used in the summary method; in the anova method they refer to additional objects of class `eba`.

Details

`eba` is a wrapper function for `OptiPt`. Both functions can be used interchangeably. See Wickelmaier and Schmid (2004) for further details.

The probabilistic choice models that can be fitted to paired-comparison data are the Bradley-Terry-Luce (BTL) model (Bradley, 1984; Luce, 1959), preference tree (Pretree) models (Tversky & Satath, 1979), and elimination-by-aspects (EBA) models (Tversky, 1972), the former being special cases of the latter.

`A` represents the family of aspect sets. It is usually a list of vectors, the first element of each being a number from 1 to `I`; additional elements specify the aspects shared by several stimuli. `A` must have as many elements as there are stimuli. When fitting a BTL model, `A` reduces to `1:I` (the default), i.e. there is only one aspect per stimulus.
The maximum likelihood estimation of the parameters is carried out by `nlm`. The Hessian matrix, however, is approximated by `nlme::fdHess`. The likelihood functions `lNconstrained` and `l` are called automatically.

See `group.test` for details on the likelihood ratio tests reported by `summary.eba`.

**Value**

- **coefficients**: a vector of parameter estimates
- **estimate**: same as `coefficients`
- **logL.eba**: the log-likelihood of the fitted model
- **logL.sat**: the log-likelihood of the saturated (binomial) model
- **goodness.of.fit**: the goodness of fit statistic including the likelihood ratio fitted vs. saturated model (-2logL), the degrees of freedom, and the p-value of the corresponding chi-square distribution
- **u.scale**: the unnormalized utility scale of the stimuli; each utility scale value is defined as the sum of aspect values (parameters) that characterize a given stimulus
- **hessian**: the Hessian matrix of the likelihood function
- **cov.p**: the covariance matrix of the model parameters
- **chi.alt**: the Pearson chi-square goodness of fit statistic
- **fitted**: the fitted paired-comparison matrix
- **y1**: the data vector of the upper triangle matrix
- **y0**: the data vector of the lower triangle matrix
- **n**: the number of observations per pair (y1 + y0)
- **mu**: the predicted choice probabilities for the upper triangle
- **nobs**: the number of pairs

**Author(s)**

Florian Wickelmaier

**References**


See Also
strans, uscale, cov.u, group.test, wald.test, plot.eba, residuals.eba, logLik.eba, simulate.eba, kendall.u, circular, trineq, thurstone, nlm.

Examples
data(celebrities)  # absolute choice frequencies
btl1 <- eba(celebrities)  # fit Bradley-Terry-Luce model
A <- list(c(1,10), c(2,10), c(3,10),
  c(4,11), c(5,11), c(6,11),
  c(7,12), c(8,12), c(9,12))  # the structure of aspects
eba1 <- eba(celebrities, A)  # fit elimination-by-aspects model
summary(eba1)  # goodness of fit
plot(eba1)  # residuals versus predicted values
anova(btl1, eba1)  # model comparison based on likelihoods
confint(eba1)  # confidence intervals for parameters
uscale(eba1)  # utility scale
ci <- 1.96 * sqrt(diag(cov.useba(eba1)))  # 95% CI for utility scale values
dotchart(uscale(eba1), xlab="Utility scale value (EBA model)", pch=16)  # plot the scale
arrows(uscale(eba1)-ci, 1:9, uscale(eba1)+ci, 1:9, .05, 90, 3)  # error bars
dotchart(uscale(eba1), xlab="Utility scale value (EBA model)", pch=16)
abline(v=1/9, lty=1)  # indifference line
mtext("(Rumelhart and Greeno, 1971)", line=1.5)

Description
Fits a (multi-attribute) probabilistic choice model that accounts for the effect of the presentation order within a pair.

Usage
eba.order(M1, M2 = NULL, A = I:1, s = c(rep(1/J, J), 1),
  constrained = TRUE)

## S3 method for class 'eba.order'
summary(object, ...)
s

the starting vector with default 1/J for all J aspect parameters, and 1 for the order effect

constrained

see eba

object

an object of class eba.order, typically the result of a call to eba.order

... additional arguments

Details

The choice models include a single multiplicative order effect, order, that is constant for all pairs (see Davidson and Beaver, 1977). An order effect < 1 (> 1) indicates a bias in favor of the first (second) interval.

See eba for choice models without order effect.

Several likelihood ratio tests are performed (see also summary.eba).

EBA.order tests an order-effect EBA model against a saturated binomial model; this corresponds to a goodness of fit test of the former model.

Order tests an EBA model with an order effect constrained to 1 against an unconstrained order-effect EBA model; this corresponds to a test of the order effect.

Effect tests an order-effect indifference model (where all scale values are equal, but the order effect is free) against the order-effect EBA model; this corresponds to testing for a stimulus effect; order0 is the estimate of the former model.

Wickelmaier and Choisel (2006) describe a model that generalizes the Davidson-Beaver model and allows for an order effect in Pretree and EBA models.

Value

coefficients a vector of parameter estimates, the last component holds the order-effect estimate

estimate same as coefficients

logL.eba the log-likelihood of the fitted model

logL.sat the log-likelihood of the saturated (binomial) model

goodness.of.fit the goodness of fit statistic including the likelihood ratio fitted vs. saturated model (-2logL), the degrees of freedom, and the p-value of the corresponding chi-square distribution

u.scale the unnormalized utility scale of the stimuli; each utility scale value is defined as the sum of aspect values (parameters) that characterize a given stimulus

hessian the Hessian matrix of the likelihood function
cov.p the covariance matrix of the model parameters

chi.alt the Pearson chi-square goodness of fit statistic

fitted 3d array of the fitted paired-comparison matrices

y1 the data vector of the upper triangle matrices

y0 the data vector of the lower triangle matrices

n the number of observations per pair (y1 + y0)

mu the predicted choice probabilities for the upper triangles

M1, M2 the data matrices
Author(s)

Florian Wickelmaier

References


See Also

eba, group.test, plot.eba, residuals.eba, logLik.eba.

Examples

data(heaviness) # weights judging data
ebao1 <- eba.order(heaviness) # Davidson-Beaver model
summary(ebao1) # goodness of fit
plot(ebao1) # residuals versus predicted values
confint(ebao1) # confidence intervals for parameters

Description

Tests for group effects in elimination-by-aspects (EBA) models.

Usage

group.test(groups, A = 1:I, s = rep(1/J, J), constrained = TRUE)

Arguments

groups a 3d array containing one aggregate choice matrix per group
A a list of vectors consisting of the stimulus aspects; the default is 1:I, where I is the number of stimuli
s the starting vector with default 1/J for all parameters, where J is the number of parameters
constrained logical, if TRUE (default), EBA parameters are constrained to be positive
Details

The five tests are all based on likelihood ratios.

Overall compares a 1-parameter Poisson model to a saturated Poisson model, thereby testing the equality of the frequencies in each cell of the array. This test corresponds to simultaneously testing for a null effect of (1) the context induced by a given pair, (2) the grouping factor, (3) the stimuli, and (4) the imbalance between pairs. The deviances of the remaining tests sum to the total deviance associated with the overall test.

EBA. g tests an EBA group model against a saturated binomial group model, which corresponds to a goodness of fit test of the EBA group model.

Group tests an EBA model having its parameters restricted to be equal across groups (single set of parameters) against the EBA group model allowing its parameters to vary freely across groups (one set of parameters per group); this corresponds to testing for group differences.

Effect tests an indifference model (where all choice probabilities are equal to 0.5) against the restricted EBA model, which corresponds to testing for a stimulus effect.

Imbalance tests for differences in the number of observations per pair by comparing the average sample size (1-parameter Poisson model) to the actual sample sizes (saturated Poisson model).


Value

tests a table displaying the likelihood ratio test statistics

References


See Also

eba, wald.test.

Examples

## Bradley-Terry-Luce model
data(pork)  # Is there a difference between Judge 1 and Judge 2?
groups <- array(c(apply(pork[, 1:5], 1:2, sum),
                 apply(pork[, 6:10], 1:2, sum)), c(3,3,2))
group.test(groups)  # Yes, there is.

## Elimination-by-aspects model
data(drugrisk)  # Do younger and older males judge risk of drugs differently?
A2 <- list(c(1), c(2,7), c(3,7), c(4,7,8), c(5,7,8), c(6,7,8))
group.test(drugrisk[, , 3:4], A2)  # Yes.
**heaviness**

### Weights Judging Data

**Description**

Fifty subjects were presented with all 20 ordered pairs of bottles filled with lead shot and asked to choose the bottle that felt heavier. The mass of the bottles was 90, 95, 100, 105, and 110 grams, respectively. Choice frequencies were aggregated across subjects for the two within-pair presentation orders.

**Usage**

```r
data(heaviness)
```

**Format**

A 3d array consisting of two square matrices:

- `heaviness[, , order = 1]` holds the choices where the row stimulus was presented first for each pair (in the upper triangle, and vice versa in the lower triangle).
- `heaviness[, , order = 2]` holds the choices where the column stimulus was presented first for each pair (in the upper triangle, and vice versa in the lower triangle).

**Source**


**Examples**

```r
data(heaviness)
## 6 subjects chose 90g over 100g, when 90g was presented first.
heaviness["90g", "100g", order=1]

## 44 subjects chose 100g over 90g, when 90g was presented first.
heaviness["100g", "90g", order=1]

## 14 subjects chose 90g over 100g, when 90g was presented second.
heaviness["90g", "100g", order=2]

## 36 subjects chose 100g over 90g, when 90g was presented second.
heaviness["100g", "90g", order=2]

## Bradley-Terry-Luce (BTL) model for each within-pair order
btl1 <- eba(heaviness[, , 1])
btl2 <- eba(heaviness[, , 2])
xval <- seq(90, 110, 5)
plot(uscale(btl1) ~ xval, type="n", log="y",
```
inclusion.rule

Description
Checks if a family of sets fulfills the inclusion rule.

Usage
inclusion.rule(A)

Arguments
A
a list of vectors consisting of the stimulus aspects of an elimination-by-aspects model

Details
The inclusion rule is necessary and sufficient for a tree structure on the aspect sets:
Structure theorem. A family \( \{ x' | x \in T \} \) of aspect sets is representable by a tree iff either \( x' \cap y' \supset x' \cap z' \) or \( x' \cap z' \supset x' \cap y' \) for all \( x, y, z \in T \). (Tversky & Sattath, 1979, p. 546)

Value
Either TRUE if the inclusion rule holds for A, or FALSE otherwise.

References

See Also
eba, trineq, strans.
Examples
A <- list(c(1, 5), c(2, 5), c(3, 6), c(4, 6))  # tree
inclusion.rule(A)

B <- list(c(1, 5), c(2, 5, 6), c(3, 6), c(4, 6))  # lattice
inclusion.rule(B)

**Description**
Kendall’s u coefficient of agreement between judges.

**Usage**
kendall.u(M, correct = TRUE)

**Arguments**
- **M**
a square matrix or a data frame consisting of absolute choice frequencies; row stimuli are chosen over column stimuli
- **correct**
logical, if TRUE (default) a continuity correction is applied when computing the test statistic (by subtracting one from the sum of agreeing pairs)

**Details**
Kendall’s u (Kendall and Babington Smith, 1940) takes on values between \( \min.u \) (minimum agreement) and 1 (maximum agreement). The minimum \( \min.u \) equals \(-1/(m - 1)\), if \( m \) is even, and \(-1/m\), if \( m \) is odd, where \( m \) is the number of subjects (judges).

The null hypothesis in the chi-square test is that the agreement between judges is by chance.
It is assumed that there is an equal number of observations per pair and that each subject judges each pair only once.

**Value**
- **u**
  Kendall’s u coefficient of agreement
- **min.u**
  the minimum value for u
- **chi2**
  the chi-square statistic for a test that the agreement is by chance
- **df**
  the degrees of freedom
- **p.value**
  the p-value of the test

**References**
See Also

`schoolsubjects, eba, strans, circular`.

Examples

data(schoolsubjects)
lapply(schoolsubjects, kendall.u)  # better-than-chance agreement

linear2btl  

*Linear Coefficients to Bradley-Terry-Luce (BTL) Estimates*

Description

Transforms linear model coefficients to Bradley-Terry-Luce (BTL) model parameter estimates.

Usage

`linear2btl(object, order = FALSE)`

Arguments

- `object`: an object of class `glm` or `lm` specifying a BTL model
- `order`: logical, does the model include an order effect? Defaults to FALSE

Details

The design matrix used by `glm` or `lm` usually results from a call to `pcX`. It is assumed that the reference category is the first level.

The covariance matrix is estimated by employing the delta method.


Value

- `btl.parameters`: a matrix; the first column holds the BTL parameter estimates, the second column the approximate standard errors
- `cova`: the approximate covariance matrix of the BTL parameter estimates
- `linear.coefs`: a vector of the original linear coefficients as returned by `glm` or `lm`

References


See Also

`eba, eba.order, glm, pcX`. 
Examples

data(drugrisk)
y1 <- t(drugrisk[, , 1])[lower.tri(drugrisk[, , 1])]
y0 <- drugrisk[, , 1][lower.tri(drugrisk[, , 1])]

## Fit BTL model using glm (maximum likelihood)
btl.glm <- glm(cbind(y1, y0) ~ 0 + pcX(6), binomial)
linear2btl(btl.glm)

## Fit BTL model using lm (weighted least squares)
btl.lm <- lm(log(y1/y0) ~ 0 + pcX(6), weights=y1*y0/(y1 + y0))
linear2btl(btl.lm)

---

logLik.eba

*Log-Likelihood of an eba Object*

Description

Returns the log-likelihood value of the (multi-attribute) probabilistic choice model represented by object evaluated at the estimated parameters.

Usage

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

Arguments

- `object`: an object inheriting from class eba, representing a fitted elimination-by-aspects model.
- `...`: some methods for this generic require additional arguments. None are used in this method.

Value

The log-likelihood of the model represented by object evaluated at the estimated parameters.

See Also

eba, logLik.lm, AIC.

Examples

data(heaviness)
bt11 <- eba(heaviness[, , order=1])
logLik(btl11)
AIC(btl11)
BIC(btl11)
**mbt**

*Mallows-Bradley-Terry Model*

**Description**

Fits a Mallows-Bradley-Terry (MBT) model by maximum likelihood.

**Usage**

```r
mbt(data, bootstrap = FALSE, nsim = 1000, ...)
```

**Arguments**

- `data`: a data frame, the first `t` columns containing the ranks, the `(t + 1)`th column containing the frequencies.
- `bootstrap`: logical. Return a parametric bootstrap p-value?
- `nsim`: number of bootstrap replicates.
- `...`: further arguments passed to `simulate`.

**Details**

`mbt` provides a front end for `glm`.

See Critchlow and Fligner (1991) for more details.

**Value**

- `coefficients`: a vector of parameter estimates (scale values) constrained to sum to unity
- `goodness.of.fit`: the goodness of fit statistic including the likelihood ratio fitted vs. saturated model (-2logL), the degrees of freedom, the p-value of the corresponding chi-square distribution, and if `bootstrap` is `TRUE` the bootstrap p-value
- `perm.idx`: the names of the non-zero frequency ranks
- `y`: the vector of rank frequencies including zeros
- `mbt.glm`: the output from a call to `glm`

**Author(s)**

Florian Wickelmaier

**References**


See Also

tartness, glm.

Examples

data(tartness) # tartness rankings of salad dressings (Vargo, 1989)
mbt(tartness, bootstrap=TRUE, nsim=500) # fit Mallows-Bradley-Terry model

Description

Computes a paired-comparison design matrix.

Usage

pcX(nstimuli, omitRef = TRUE)

Arguments

nstimuli number of stimuli in the paired-comparison design
omitRef logical, if TRUE (default), the first column corresponding to the reference category is omitted

Details

The design matrix can be used when fitting a Bradley-Terry-Luce (BTL) model or a Thurstone-Mosteller (TM) model by means of glm or lm.

See Critchlow and Fligner (1991) for more details.

Value

A matrix having (nstimuli - 1)*nstimuli/2 rows and nstimuli - 1 columns (if the reference category is omitted).

References


See Also

eba, thurstone, glm, balanced.pcdesign, linear2btl.
Examples

```r
data(drugrisk)  # absolute choice frequencies
btl <- eba(drugrisk[, , 1])  # fit Bradley-Terry-Luce model using eba
summary(btl)

y1 <- t(drugrisk[, , 1][lower.tri(drugrisk[, , 1])])
y0 <- drugrisk[, , 1][lower.tri(drugrisk[, , 1])]

## Fit Bradley-Terry-Luce model using glm
btl.glm <- glm(cbind(y1, y0) ~ 0 + pcX(6), binomial)
summary(btl.glm)

## Fit Thurstone Case V model using glm
tm.glm <- glm(cbind(y1, y0) ~ 0 + pcX(6), binomial(probit))
summary(tm.glm)
```

plot.eba  

## Diagnostic Plot for EBA Models

### Description

Plots elimination-by-aspects (EBA) model residuals against fitted values.

### Usage

```
## S3 method for class 'eba'
plot(x, xlab = "Predicted choice probabilities",
ylab = "Deviance residuals", ...)
```

### Arguments

- `x`  
an object of class `eba`, typically the result of a call to `eba`
- `xlab`, `ylab`, ...
  
  graphical parameters passed to `plot`.

### Details

The deviance residuals are plotted against the predicted choice probabilities for the upper triangle of the paired-comparison matrix.

### See Also

`eba`, `residuals.eba`. 
Examples

```r
## Compare two choice models

data(celebrities)                      # absolute choice frequencies
btl1 <- eba(celebrities)               # fit Bradley-Terry-Luce model
A <- list(c(1,10), c(2,10), c(3,10),
          c(4,11), c(5,11), c(6,11),
          c(7,12), c(8,12), c(9,12))  # the structure of aspects
eba1 <- eba(celebrities, A)           # fit elimination-by-aspects model
anova(btl1, eba1)                     # model comparison based on likelihoods
par(mfrow = 1:2)                       # residuals versus fitted values
plot(btl1, main = "BTL", ylim = c(-4, 4.5))  # BTL doesn't fit well
plot(eba1, main = "EBA", ylim = c(-4, 4.5))  # EBA fits better
```

Description

Bradley and Terry (1952) provide the individual choice matrices of two judges choosing between pairs of three samples of pork meet. The pigs had been fed on either corn (C), corn plus peanut supplement (Cp), or corn plus a large peanut supplement (CP). Each judge does five repetitions. The data are stored in a 3d array, the first five matrices of which correspond to the five repetitions of the first judge, the last five to the repetitions of the second judge. Row stimuli are chosen (preferred) over column stimuli.

Usage

data(pork)

Format

A 3d array consisting of ten square matrices.

Source


Examples

```r
data(pork)
apply(pork, 1:2, sum)  # aggregate choice frequencies
```
Description

Computes deviance and Pearson residuals for eba objects.

Usage

```r
## S3 method for class 'eba'
residuals(object, type = c("deviance", "pearson"), ...)
```

Arguments

- `object`: an object of class eba, typically the result of a call to `eba`
- `type`: the type of residuals which should be returned; the alternatives are: "deviance" (default) and "pearson"
- `...`: further arguments passed to or from other methods. None are used in this method.

Details

Residuals are computed from the upper triangle of the paired-comparison matrix.
See `residuals.glm` for details.

Value

A vector of residuals having as many elements as pairs of stimuli.

See Also

eba, residuals.glm, plot.eba.

Examples

```r
data(celebrities) # absolute choice frequencies
btl1 <- eba(celebrities) # fit Bradley-Terry-Luce model
sum( resid(btl1)^2 ) # deviance G2
deviance(btl1)
sum( resid(btl1, "pearson")^2 ) # Pearson X2
```
Description

Two classes of children (ages 11 to 13) were asked to state their preferences with respect to certain school subjects. Each child was given a sheet on which were written the possible pairs of subjects and asked to underline the one preferred in each case.

Usage

data(schoolsubjects)

Format

A list containing two square matrices of aggregate choice frequencies (row entries are preferred over column entries):

schoolsubjects["boys"] holds the frequencies of 21 boys choosing among 13 school subjects: woodwork, gymnastics, art, science, history, geography, arithmetic, religion, English literature, commercial subjects, algebra, English grammar, geometry.

schoolsubjects["girls"] holds the frequencies of 25 girls choosing among 11 school subjects: gymnastics, science, art, domestic science, history, arithmetic, geography, English literature, religion, algebra, English grammar.

Source


Examples

data(schoolsubjects)

m <- lapply(schoolsubjects, eba)  # Bradley-Terry-Luce (BTL) model
par(mfrow = 1:2)
dotchart(uscale(m$boys), main = "Boys' preferences",)
dotchart(uscale(m$girls), main = "Girls' preferences")
mtext("Utility scale value (BTL model)", outer = TRUE, side = 1, line = -2)
mtext("(Kendall and Babington Smith, 1940)", outer = TRUE, line = -4)
Simulate Responses from EBA Models

Description

Simulates responses from the distribution corresponding to a fitted eba model object.

Usage

```r
## S3 method for class 'eba'
simulate(object, nsim, seed, pool = TRUE, ...)
```

Arguments

- `object`: an object of class `eba`, typically the result of a call to `eba`.
- `nsim`: currently not used.
- `seed`: currently not used.
- `pool`: logical, if TRUE (default), pooled responses (summed across respondents) are returned.
- `...`: further arguments passed to or from other methods. None are used in this method.

Details

Responses are simulated by `rbinom` with sizes taken from the original sample and probabilities computed from the model object.

Value

A paired-comparison matrix of (pooled) responses.

See Also

eba, boot.

Examples

```r
data(celebrities)  # absolute choice frequencies
A <- list(c(1,10), c(2,10), c(3,10),
          c(4,11), c(5,11), c(6,11),
          c(7,12), c(8,12), c(9,12))  # the structure of aspects
eba1 <- eba(celebrities, A)  # fit elimination-by-aspects model

## Perform parametric bootstrap
LR.stat <- replicate(200, deviance(eba(simulate(eba1), A)))

hist(LR.stat, col="lightgray", border="white", freq=FALSE, breaks=20,
```
Description

Checks the weak, moderate, and strong stochastic transitivity.

Usage

strans(M)

Arguments

M a square matrix or a data frame consisting of absolute choice frequencies; row stimuli are chosen over column stimuli

Details

The weak (WST), moderate (MST), and strong (SST) stochastic transitivity hold for a set of choice probabilities \( P \), whenever if \( P_{ij} \geq 0.5 \) and \( P_{jk} \geq 0.5 \), then

\[
P_{ik} \geq 0.5 \quad \text{(WST)},
\]

\[
P_{ik} \geq \min(P_{ij}, P_{jk}) \quad \text{(MST)},
\]

\[
P_{ik} \geq \max(P_{ij}, P_{jk}) \quad \text{(SST)}.
\]

See Suppes, Krantz, Luce, & Tversky (1989/2007, chap. 17) for an introduction to the representation of choice probabilities.

If WST holds, a permutation of the indices of the matrix exists such that the proportions in the upper triangular matrix are \( \geq 0.5 \). This re-arranged matrix is stored in \( \text{pcm} \). If WST does not hold, cells in the upper triangular matrix that are smaller than 0.5 are replaced by 0.5. The deviance resulting from this restriction is reported in \( \text{wst.fit} \).

The approximate likelihood ratio test for significance of the WST violations is according to Tversky (1969); for a more exact test of WST see Iverson and Falmagne (1985).

Value

A table displaying the number of violations of the weak, moderate, and strong stochastic transitivity, the number of tests, the error ratio (violations/tests), and the mean and maximum deviation from the minimum probability for which the corresponding transitivity would hold.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>weak</td>
<td>number of violations of WST</td>
</tr>
<tr>
<td>moderate</td>
<td>number of violations of MST</td>
</tr>
<tr>
<td>strong</td>
<td>number of violations of SST</td>
</tr>
</tbody>
</table>
n.tests number of transitivity tests performed
wst.violations a vector containing $0.5 - P_{ik}$ for all triples that violate WST
mst.violations a vector containing $\min(P_{ij}, P_{jk}) - P_{ik}$ for all triples that violate MST
sst.violations a vector containing $\max(P_{ij}, P_{jk}) - P_{ik}$ for all triples that violate SST
pcm the permuted square matrix of relative choice frequencies
ranking the ranking of the objects, which corresponds to the colnames of pcm
chkdf data frame reporting the choice proportions for each triple in each permutation
violdf data frame reporting for each triple which type of transitivity holds or does not hold
wst.fit likelihood ratio test of WST (see details)
wst.mat restricted matrix that satisfies WST

References


See Also

eba, circular, kendall.u, trineq.

Examples

data(celebrities) # absolute choice frequencies
strans(celebrities) # WST and MST hold, but not SST
strans(celebrities)$pcm # re-ordered relative frequencies
strans(celebrities)$violdf # transitivity violations

tartness Ttartness Rankings of Salad Dressings

Description

The data were collected by Vargo (1989; as cited in Critchlow and Fligner, 1991). Each of 32 judges is asked to rank four salad dressing preparations according to tartness, with a rank of 1 being assigned to the formulation judged to be the most tart.

Usage

data(tartness)
**thurstone**

**Format**

a data frame consisting the rankings and their frequencies.

**Source**


**References**

Vargo, M.D. (1989). Microbiological spoilage of a moderate acid food system using a dairy-based salad dressing model. Unpublished masters thesis, Ohio State University, Department of Food Science and Nutrition, Columbus, OH.

**Examples**

```r
data(tartness)
```

**thurstone**

**Thurstone-Mosteller Model (Case V)**

**Description**

Fits a Thurstone-Mosteller model (Case V) by maximum likelihood.

**Usage**

```r
thurstone(M)
```

**Arguments**

- `M` a square matrix or a data frame consisting of absolute choice frequencies; row stimuli are chosen over column stimuli

**Details**

thurstone provides a front end for glm.

See Critchlow and Fligner (1991) for more details.

**Value**

- `estimate` a vector of parameter estimates (scale values), first element is set to zero
- `goodness.of.fit` the goodness of fit statistic including the likelihood ratio fitted vs. saturated model (-2logL), the degrees of freedom, and the p-value of the corresponding chi-square distribution
- `tm.glm` the output from a call to glm
trineq

**Author(s)**

Florian Wickelmaier

**References**


**See Also**

eba, strans, pcx, kendall.u, circular, glm.

**Examples**

```r
# Taste data (David, 1988, p. 116)
taste <- matrix(c(0, 3, 2, 2,
                 12, 0, 11, 3,
                 13, 4, 0, 5,
                 13, 12, 10, 0), 4, 4, byrow=TRUE)
dimnames(taste) <- setNames(rep(list(c("A1", "A2", "A3", "A4")), 2), c(">", ":", "<"))
thurstone(taste) # Thurstone-Mosteller model fits OK
```

---

**trineq**  
*Trinary Inequality*

**Description**

Checks if binary choice probabilities fulfill the trinary inequality.

**Usage**

`trineq(M, A = 1:I)`

**Arguments**

- **M**
  - a square matrix or a data frame consisting of absolute choice frequencies; row stimuli are chosen over column stimuli
- **A**
  - a list of vectors consisting of the stimulus aspects; the default is `1:I`, where `I` is the number of stimuli
Details
For any triple of stimuli \(x, y, z\), the trinary inequality states that, if \(P(x, y) > 1/2\) and \((xy)z\), then
\[
R(x, y, z) > 1,
\]
where \(R(x, y, z) = R(x, y)R(y, z)R(z, x)\), \(R(x, y) = P(x, y)/P(y, x)\), and \((xy)z\) denotes that \(x\) and \(y\) share at least one aspect that \(z\) does not have (Tversky and Sattath, 1979, p. 554).

inclusion.rule checks if a family of aspect sets is representable by a tree.

Value
Results checking the trinary inequality.

- \(n\): number of tests of the trinary inequality
- \(prop\): proportion of triples confirming the trinary inequality
- \(quant\): quantiles of \(R(x, y, z)\)
- \(n\text{.tests}\): number of transitivity tests performed
- \(chkdf\): data frame reporting \(R(x, y, z)\) for each triple where \(P(x, y) > 1/2\) and \((xy)z\)

References

See Also
eba, inclusion.rule, strans.

Examples
```r
data(celebrities) # absolute choice frequencies
A <- list(c(1,10), c(2,10), c(3,10),
          c(4,11), c(5,11), c(6,11),
          c(7,12), c(8,12), c(9,12)) # the structure of aspects
trineq(celebrities, A) # check trinary inequality for tree A
trineq(celebrities, A)$chkdf # trinary inequality for each triple
```

uscale

*Utility Scale of an EBA Choice Model*

Description
Extract the (normalized) utility scale for an elimination-by-aspects (EBA) model.

Usage
```r
uscale(object, norm = "sum", log = FALSE)
```
Arguments

**object**
- an object of class `eba`, typically the result of a call to `eba`

**norm**
- either `sum`, so the scale values sum to unity (default); or a number from 1 to number of stimuli, so this scale value becomes one; or `NULL` (no normalization)

**log**
- should the log of the utility scale values be returned? Defaults to `FALSE`.

Details

Each utility scale value is defined as the sum of aspect values (EBA model parameters) that characterize a given stimulus. First these sums are computed for all stimuli, then normalization (if any) is applied. As each type of normalization corresponds to a multiplication by a positive real, the ratio between scale values remains constant.

Value

The (normalized) utility scale of the stimuli.

See Also

`eba`, `covNu`, `wald.test`.

Examples

```r
data(drugrisk)
A <- list(c(1), c(2,7), c(3,7), c(4,7,8), c(5,7,8), c(6,7,8))
eba1 <- eba(drugrisk[,group = "male30"], A)  # EBA model

uscale(eba1)                          # sum-to-unity normalization
uscale(eba1, norm=1)                  # u(alcohol) := 1
uscale(eba1, norm=5)                  # u(heroine) := 1
uscale(eba1, norm=NULL)               # no normalization
uscale(eba1, norm=1, log=TRUE)        # log utility scale, log u(alcohol) := 0
```

wald.test  Testing Linear Hypotheses in Elimination-by-Aspects (EBA) Models

Description

Tests linear hypotheses of the form $Cp = 0$ in elimination-by-aspects (EBA) models using the Wald test.

Usage

```r
wald.test(object, C, u.scale = TRUE)
```
Arguments

object an object of class eba, typically the result of a call to eba
C a matrix of contrasts, specifying the linear hypotheses
u.scale logical, if TRUE the test is performed on the utility scale, if FALSE the test is performed on the EBA parameters directly

Details

The Wald test statistic,

$$ W = (C p) [C \text{cov}(p) C']^{-1} (C p), $$

is approximately chi-square distributed with $rk(C)$ degrees of freedom.

C is usually of full rank and must have as many columns as there are parameters in $p$.

Value

C the matrix of contrasts, specifying the linear hypotheses
w the Wald test statistic
df the degrees of freedom ($rk(C)$)
pval the p-value of the test

See Also

eba, group.test, uscale, cov.u.

Examples

data(celebrities) # absolute choice frequencies
A <- list(c(1,10), c(2,10), c(3,10),
         c(4,11), c(5,11), c(6,11),
         c(7,12), c(8,12), c(9,12)) # the structure of aspects
eba1 <- eba(celebrities, A) # fit elimination-by-aspects model

# Test whether JU, CY, and AJF have equal utility scale values
C1 <- rbind(c(0,0,0,1,-1, 0,0,0,0),
            c(0,0,0,1, 0,-1,0,0,0))
wald.test(eba1, C1)

# Test whether the three branch parameters are different
C2 <- rbind(c(0,0,0,0,0,0,0,0,1,-1, 0),
            c(0,0,0,0,0,0,0,0,1, 0,-1))
wald.test(eba1, C2, u.scale = FALSE)
Index

*Topic datasets
  celebrities, 4
  drugrisk, 7
  heaviness, 15
  pork, 23
  schoolsubjects, 25
  tartness, 28

*Topic models
  balanced.pcdesign, 2
  boot, 3
  circular, 5
  cov.u, 7
  eba, 9
  eba.order, 11
  group.test, 13
  inclusion.rule, 16
  kendall.u, 17
  linear2btl, 18
  logLik.eba, 19
  mbt, 20
  pcX, 21
  plot.eba, 22
  residuals.eba, 24
  simulate.eba, 26
  strans, 27
  thurstone, 29
  trineq, 30
  uscale, 31
  wald.test, 32

AIC, 19
anova.eba (eba), 9
anova.eba.order (eba.order), 11

balanced.pcdesign, 2, 21
boot, 3, 26

celebrities, 4
circular, 5, 11, 18, 28, 30
cov.u, 7, 11, 32, 33
deviance.eba (eba), 9
deviance.eba.order (eba.order), 11
drugrisk, 7
eba, 3, 4, 6, 7, 9, 11–14, 16, 18, 19, 21, 22,
  24, 26, 28, 30–33
eba.boot (boot), 3
eba.order, 11, 12, 18
glm, 18, 21, 30
group.test, 10, 11, 13, 13, 33
heaviness, 15
inclusion.rule, 16, 31
kendall.u, 6, 11, 17, 28, 30
L (eba), 9
L.constrained.order (eba.order), 11
L.order (eba.order), 11
linear2btl, 18, 21
lm, 21
logLik.eba, 11, 13, 19
logLik.eba.order (eba.order), 11
logLik.lm, 19
mbt, 20

nlm, 11
nobs.eba (logLik.eba), 19

OptiPt (eba), 9
pcX, 3, 18, 21, 30
plot.eba, 11, 13, 22, 24
plot.eba.order (eba.order), 11
pork, 23
print.circular (circular), 5
print.eba (eba), 9
print.eba.order (eba.order), 11
INDEX

print.group.test (group.test), 13
print.kendall.u (kendall.u), 17
print.mbt (mbt), 20
print.strans (strans), 27
print.summary.eba (eba), 9
print.summary.eba.order (eba.order), 11
print.thurstone (thurstone), 29
print.trineq (trineq), 30
print.wald.test (wald.test), 32

rbinom, 26
residuals.eba, 11, 13, 22, 24
residuals.eba.order (eba.order), 11
residuals.glm, 24

schoolsubjects, 18, 25
simulate.eba, 4, 11, 26
strans, 6, 11, 16, 18, 27, 30, 31
summary.eba, 12
summary.eba (eba), 9
summary.eba.order (eba.order), 11

tartness, 21, 28
thurstone, 11, 21, 29
trineq, 11, 16, 28, 30

uscale, 7, 11, 31, 33

vcov.eba (eba), 9
vcov.eba.order (eba.order), 11

wald.test, 7, 11, 14, 32, 32