Package ‘pks’

April 15, 2022

Version 0.5-0

Date 2022-04-14

Title Probabilistic Knowledge Structures

Depends R (>= 3.5.0), stats, sets

Imports graphics

Description Fitting and testing probabilistic knowledge structures, especially the basic local independence model (BLIM, Doignon & Flamagne, 1999), using the minimum discrepancy maximum likelihood (MDML) method (Heller & Wickelmaier, 2013 <doi:10.1016/j.endm.2013.05.145>).

License GPL (>= 2)

URL `http://www.mathpsy.uni-tuebingen.de/wickelmaier/`

NeedsCompilation no

Author Florian Wickelmaier [aut, cre], Juergen Heller [aut], Julian Mollenhauer [aut], Pasquale Anselmi [ctb]

Maintainer Florian Wickelmaier <wickelmaier@web.de>

Repository CRAN

Date/Publication 2022-04-14 22:30:02 UTC

R topics documented:

`blim` *Basic Local Independence Models (BLIMs)*

Description

Fits a basic local independence model (BLIM) for probabilistic knowledge structures by minimum discrepancy maximum likelihood estimation.

Usage

```
blim(K, N.R, method = c("MD", "ML", "MDML"), R = as.binmat(N.R),
     P.K = rep(1/nstates, nstates),
     beta = rep(0.1, nitems), eta = rep(0.1, nitems),
     betafix = rep(NA, nitems), etafix = rep(NA, nitems),
     betaequal = NULL, etaequal = NULL,
     randinit = FALSE, incradius = 0,
     tol = 1e-07, maxiter = 10000, zeropad = 16)

blimMD(K, N.R, R = as.binmat(N.R),
       betafix = rep(NA, nitems), etafix = rep(NA, nitems),
       incrule = c("minimum", "hypblc1", "hypblc2"), m = 1)

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

Arguments

`K`	a state-by-problem indicator matrix representing the knowledge structure. An element is one if the problem is contained in the state, and else zero.
`N.R`	a (named) vector of absolute frequencies of response patterns.
`method`	`MD` for minimum discrepancy estimation, `ML` for maximum likelihood estimation, `MDML` for minimum discrepancy maximum likelihood estimation.
`R`	a person-by-problem indicator matrix of unique response patterns. Per default inferred from the names of `N.R`.
`P.K`	the vector of initial parameter values for probabilities of knowledge states.
`beta, eta`	vectors of initial parameter values for probabilities of a careless error and a lucky guess, respectively.

`betafix, etafix`
: vectors of fixed error and guessing parameter values; `NA` indicates a free parameter.

`betaequal, etaequal`
: lists of vectors of problem indices; each vector represents an equivalence class: it contains the indices of problems for which the error or guessing parameters are constrained to be equal. (See Examples.)

`randinit`
: logical, if `TRUE` then initial parameter values are sampled uniformly with constraints. (See Details.)

`incradius`
: include knowledge states of distance from the minimum discrepant states less than or equal to `incradius`.

`tol`
: tolerance, stopping criterion for iteration.

`maxiter`
: the maximum number of iterations.

`zeropad`
: the maximum number of items for which an incomplete `N.R` vector is completed and padded with zeros.

`incrule`
: inclusion rule for knowledge states. (See Details.)

`m`
: exponent for hyperbolic inclusion rules.

`object`
: an object of class `blim`, typically the result of a call to `blim`.

`test`
: should the p-values of the chi-square distributions be reported?

`...`
: additional arguments passed to other methods.

Details

See Doignon and Falmagne (1999) for details on the basic local independence model (BLIM) for probabilistic knowledge structures.

Minimum discrepancy (MD) minimizes the number of expected response errors (careless errors or lucky guesses). Maximum likelihood maximizes the likelihood, possibly at the expense of inflating the error and guessing parameters. Minimum discrepancy maximum likelihood (MDML) maximizes the likelihood subject to the constraint of minimum response errors. See Heller and Wickelmaier (2013) for details on the parameter estimation methods.

If `randinit` is `TRUE`, initial parameter values are sampled uniformly with the constraint `beta + eta < 1` (Weisstein, 2013) for the error parameters, and with `sum(P.K) == 1` (Rubin, 1981) for the probabilities of knowledge states. Setting `randinit` to `TRUE` overrides any values given in the `P.K`, `beta`, and `eta` arguments.

The degrees of freedom in the goodness-of-fit test are calculated as number of possible response patterns minus one or number of respondents, whichever is smaller, minus number of parameters.

`blimMD` uses minimum discrepancy estimation only. Apart from the hyperbolic inclusion rules, all of its functionality is also provided by `blim`. It may be removed in the future.

Value

An object of class `blim` having the following components:

`discrepancy`
: the mean minimum discrepancy between response patterns and knowledge states.

`P.K`
: the vector of estimated parameter values for probabilities of knowledge states.

`beta`	the vector of estimated parameter values for probabilities of a careless error.
`eta`	the vector of estimated parameter values for probabilities of a lucky guess.
`disc.tab`	the minimum discrepancy distribution.
`K`	the knowledge structure.
`N.R`	the vector of frequencies of response patterns.
`nitems`	the number of items.
`nstates`	the number of knowledge states.
`npatterns`	the number of response patterns.
`ntotal`	the number of respondents.
`nerror`	the number of response errors.
`npar`	the number of parameters.
`method`	the parameter estimation method.
`iter`	the number of iterations needed.
`loglik`	the log-likelihood.
`fitted.values`	the fitted response frequencies.
`goodness.of.fit`	the goodness of fit statistic including the likelihood ratio fitted vs. saturated model (G2), the degrees of freedom, and the p-value of the corresponding chi-square distribution. (See Details.)

References

Doignon, J.-P., & Falmagne, J.-C. (1999). *Knowledge spaces*. Berlin: Springer.

Heller, J., & Wickelmaier, F. (2013). Minimum discrepancy estimation in probabilistic knowledge structures. *Electronic Notes in Discrete Mathematics*, **42**, 49–56. doi: 10.1016/j.endm.2013.05.145

Rubin, D.B. (1981). The Bayesian bootstrap. *The Annals of Statistics*, **9**(1), 130–134. doi: 10.1214/aos/1176345338

Weisstein, E.W. (2013, August 29). Triangle point picking. In *MathWorld – A Wolfram Web Resource*. Retrieved from `https://mathworld.wolfram.com/TrianglePointPicking.html`.

See Also

`simulate.blim`, `plot.blim`, `residuals.blim`, `logLik.blim`, `delineate`, `jacobian`, `endm`, `probability`, `chess`.

Examples

```
data(DoignonFalmagne7)
K   <- DoignonFalmagne7$K     # knowledge structure
N.R <- DoignonFalmagne7$N.R  # frequencies of response patterns

## Fit basic local independence model (BLIM) by different methods
blim(K, N.R, method = "MD")    # minimum discrepancy estimation
blim(K, N.R, method = "ML")    # maximum likelihood estimation by EM
```

```
blim(K, N.R, method = "MDML")  # MDML estimation

## Parameter restrictions: beta_a = beta_b = beta_d, beta_c = beta_e
##                          eta_a =  eta_b = 0.1
m1 <- blim(K, N.R, method = "ML",
           betaequal = list(c(1, 2, 4), c(3, 5)),
              etafix = c(0.1, 0.1, NA, NA, NA))
m2 <- blim(K, N.R, method = "ML")
anova(m1, m2)

## See ?endm, ?probability, and ?chess for further examples.
```

---

chess *Responses to Chess Problems and Knowledge Structures*

Description

Held, Schrepp and Fries (1995) derive several knowledge structures for the representation of 92 responses to 16 chess problems. See Schrepp, Held and Albert (1999) for a detailed description of these problems.

Usage

```
data(chess)
```

Format

A list consisting of five components:

`dst1` a state-by-problem indicator matrix representing the knowledge structure DST1.

`dst3` the knowledge structure DST3.

`dst4` the knowledge structure DST4.

`N.R` a named integer vector. The names denote response patterns, the values denote their frequencies.

`R` a state-by-problem indicator matrix representing the raw responses. Column names `hdbgXX` and `grazYY` identify responses collected in Heidelberg and Graz, respectively.

Note

The graphs of the precedence relations for DST1 and DST4 in Held et. al (1995) contain mistakes that have been corrected. See examples.

Source

Held, T., Schrepp, M., & Fries, S. (1995). Methoden zur Bestimmung von Wissensstrukturen – eine Vergleichsstudie. *Zeitschrift fuer Experimentelle Psychologie*, **42**(2), 205–236.

References

Schrepp, M., Held, T., & Albert, D. (1999). Component-based construction of surmise relations for chess problems. In D. Albert & J. Lukas (Eds.), *Knowledge spaces: Theories, empirical research, and applications* (pp. 41–66). Mahwah, NJ: Erlbaum.

Examples

```
data(chess)
chess$dst1  # knowledge structure DST1

## Precedence relation (Held et al., 1995, p. 215) and knowledge space
P <- as.binmat(c("1111011101111001",   # s
               # "0100000000000000",   # gs   mistake in Abb. 3
                 "0111010100111000",   # gs   correction
                 "0011010000011000",   # egs
                 "0011010000011000",   # eegs
                 "0000110000000000",   # cs
                 "0000010000000000",   # gcs
                 "0011011100111000",   # ts
                 "0011010100011000",   # ges
                 "1111111111111111",   # f
                 "0111010101111000",   # gf
                 "0011010000111000",   # gff
                 "0000000000010000",   # ggff
                 "0000000000001000",   # ggf
                 "0111011101111101",   # ff
                 "0111011101111011",   # tf
                 "0011010100111001"),  # tff
               as.logical = TRUE)
dimnames(P) <- list("<" = colnames(chess$R), ">" = colnames(chess$R))
K <- rbind(0L, binary_closure(t(P)))
identical(sort(as.pattern(K)),
          sort(as.pattern(chess$dst1)))

blim(chess$dst1, chess$N.R)  # Tab. 1
```

`conversion` *Conversion between Representations of Responses or States*

Description

Converts between binary matrix and pattern representations of response patterns or knowledge states.

Usage

```
as.pattern(R, freq = FALSE, as.letters = FALSE, as.set = FALSE)

as.binmat(N.R, uniq = TRUE, col.names = NULL, as.logical = FALSE)
```

Arguments

`R`	an indicator matrix of response patterns or knowledge states.
`N.R`	either a (named) vector of absolute frequencies of response patterns; or a character vector of response patterns or knowledge states; or a `set` of sets representing the knowledge structure.
`freq`	logical, should the frequencies of response patterns be reported?
`uniq`	logical, if `TRUE`, only the unique response patterns are returned.
`as.letters`	logical, return response patterns as combinations of letters.
`as.set`	logical, return response patterns as set of sets.
`col.names`	column names for the state or response matrix.
`as.logical`	logical, return logical matrix of states.

Value

`as.pattern` returns a vector of integers named by the response patterns if `freq` is `TRUE`, else a character vector. If `as.set` is `TRUE`, the return value is of class `set`.

`as.binmat` returns an indicator matrix. If `as.logical` is `TRUE`, it returns a logical matrix.

See Also

`blim`, `set` in package `sets`.

Examples

```
data(DoignonFalmagne7)
as.pattern(DoignonFalmagne7$K)
as.pattern(DoignonFalmagne7$K, freq = TRUE)
as.pattern(DoignonFalmagne7$K, as.letters = TRUE)
as.pattern(DoignonFalmagne7$K, as.set = TRUE)

dim(as.binmat(DoignonFalmagne7$N.R))
dim(as.binmat(DoignonFalmagne7$N.R, uniq = FALSE))

## Knowledge structure as binary matrix
as.binmat(c("000", "100", "101", "111"))
as.binmat(set(set(), set("a"), set("a", "c"), set("a", "b", "c")))
as.binmat(c("000", "100", "101", "111"), as.logical = TRUE)
```

`delineate` *Delineate a Knowledge Structure by a Skill Function*

Description

Computes the knowledge structure delineated by a skill function.

Usage

```
delineate(skillfun, itemID = 1)
```

Arguments

`skillfun`	a data frame or a matrix representing the skill function. It consists of an item indicator and a problem-by-skill indicator matrix.
`itemID`	index of the column in `skillfun` that holds the item indicator.

Details

The skill function (Q, S, μ) indicates for each item in Q which subsets of skills in S are required to solve the item. Thus, $\mu(q)$ is a set containing sets of skills. An item may have multiple entries in `skillfun`, each in a separate row identified by the same `itemID`.

See Doignon and Falmagne (1999, Chap. 4).

Value

A list of two components:

`K`	the knowledge structure delineated by the skill function.
`classes`	a list of equivalence classes of competence states; the members of these classes are mapped onto the same knowledge state by the problem function induced by the skill function μ.

References

Doignon, J.-P., & Falmagne, J.-C. (1999). *Knowledge spaces*. Berlin: Springer.

See Also

blim.

Examples

```
# Skill function
# mu(e) = {{s, t}, {s, u}},  mu(f) = {{u}}
# mu(g) = {{s}, {t}},        mu(h) = {{t}}
sf <- read.table(header = TRUE, text = "
  item s t u
     e 1 1 0
     e 1 0 1
     f 0 0 1
     g 1 0 0
     g 0 1 0
     h 0 1 0
")
delineate(sf)

## See ?probability for further examples.
```

DoignonFalmagne7 *Artificial Responses from Doignon and Falmagne (1999)*

Description

Fictitious data set from Doignon and Falmagne (1999, chap. 7). Response patterns of 1000 respondents to five problems. Each respondent is assumed to be in one of nine possible states of the knowledge structure `K`.

Usage

```
data(DoignonFalmagne7)
```

Format

A list consisting of two components:

`K` a state-by-problem indicator matrix representing the hypothetical knowledge structure. An element is one if the problem is contained in the state, and else zero.

`N.R` a named numeric vector. The names denote response patterns, the values denote their frequencies.

Source

Doignon, J.-P., & Falmagne, J.-C. (1999). *Knowledge spaces*. Berlin: Springer.

Examples

```
data(DoignonFalmagne7)
DoignonFalmagne7$K    # knowledge structure
DoignonFalmagne7$N.R  # response patterns
```

endm *Responses and Knowledge Structures from Heller and Wickelmaier (2013)*

Description

Knowledge structures and 200 artificial responses to four problems are used to illustrate parameter estimation in Heller and Wickelmaier (2013).

Usage

```
data(endm)
```

Format

A list consisting of three components:

`K` a state-by-problem indicator matrix representing the true knowledge structure that underlies the model that generated the data.

`K2` a slightly misspecified knowledge structure.

`N.R` a named numeric vector. The names denote response patterns, the values denote their frequencies.

Source

Heller, J., & Wickelmaier, F. (2013). Minimum discrepancy estimation in probabilistic knowledge structures. *Electronic Notes in Discrete Mathematics*, **42**, 49–56. doi: 10.1016/j.endm.2013.05.145

Examples

```
data(endm)
endm$K    # true knowledge structure
endm$K2   # misspecified knowledge structure
endm$N.R  # response patterns

## Generate data from BLIM based on K
blim0 <- list(
     P.K = setNames(c(.1, .15, .15, .2, .2, .1, .1), as.pattern(endm$K)),
    beta = rep(.1, 4),
     eta = rep(.1, 4),
       K = endm$K,
  ntotal = 200)
class(blim0) <- "blim"
simulate(blim0)

## Fit BLIM based on K2
blim1 <- blim(endm$K2, endm$N.R, "MD")
```

`gradedness` *Forward- or Backward-Gradedness of a Knowledge Structure*

Description

Checks if a knowledge structure is forward- or backward-graded in any item.

Usage

```
is.forward.graded(K)

is.backward.graded(K)
```

Arguments

`K`	a state-by-problem indicator matrix representing the knowledge structure. An element is one if the problem is contained in the state, and else zero. `K` should have non-empty `colnames`.

Details

A knowledge structure K is forward-graded in item q, if $S \cup \{q\}$ is in K for every state $S \in K$.

A knowledge structure K is backward-graded in item q, if $S - \{q\}$ is in K for every state $S \in K$.

See Spoto, Stefanutti, and Vidotto (2012).

Value

A named logical vector with as many elements as columns in `K`.

References

Spoto, A., Stefanutti, L., & Vidotto, G. (2012). On the unidentifiability of a certain class of skill multi map based probabilistic knowledge structures. *Journal of Mathematical Psychology*, **56**(4), 248–255. doi: 10.1016/j.jmp.2012.05.001

See Also

`blim`, `jacobian`.

Examples

```
K <- as.binmat(c("0000", "1000", "1100", "1010", "0110", "1110", "1111"))
is.forward.graded(K)   # forward-graded in a
is.backward.graded(K)  # not backward-graded in a
```

`ita` *Item Tree Analysis (ITA)*

Description

Performs an item tree analysis (ITA) on a set of binary responses.

Usage

```
ita(R, L = NULL, makeK = FALSE)
```

Arguments

`R`	a subject-by-problem indicator matrix representing the responses.
`L`	the threshold of violations acceptable for the precedence relation. If `NULL` (default), an optimal threshold is searched for.
`makeK`	should the corresponding knowledge structure be returned?

Details

ITA seeks to establish a precedence relation among a set of binary items. For each pair of items (p, q), it counts how often p is not solved if q is solved, which constitutes a violation of the relation. ITA searches for a threshold `L` for the maximum number of violations consistent with a (transitive) precedence relation.

See van Leeuwe (1974) and Schrepp (1999) for details.

Value

An object of class `ita` having the following components:

`K`	the knowledge structure corresponding to the precedence relation.
`discrepancy`	the discrepancy between `R` and `K` (fit), between `K` and `R` (complexity), and their sum (total).
`transitiveL`	the vector of transitive thresholds.
`searchL`	has the threshold value been optimized?
`L`	the selected or requested threshold.
`P`	the precedence matrix containing the number of violations.

References

Schrepp, M. (1999). On the empirical construction of implications between bi-valued test items. *Mathematical Social Sciences*, **38**(3), 361–375. doi: 10.1016/S01654896(99)000256

Van Leeuwe, J.F. (1974). Item tree analysis. *Nederlands Tijdschrift voor de Psychologie en haar Grensgebieden*, **29**(6), 475–483.

See Also

`blim`.

Examples

```
data(chess)

ita(chess$R)  # find optimal threshold L

i <- ita(chess$R, L = 6, makeK = TRUE)
identical(sort(as.pattern(i$K)),
          sort(as.pattern(chess$dst1)))
```

jacobian *Jacobian Matrix for Basic Local Independence Model*

Description

Computes the Jacobian matrix for a basic local independence model (BLIM).

Usage

```
jacobian(object, P.K = rep(1/nstates, nstates),
         beta = rep(0.1, nitems), eta = rep(0.1, nitems),
         betafix = rep(NA, nitems), etafix = rep(NA, nitems))
```

Arguments

- `object` an object of class `blim`, typically the result of a call to `blim`.
- `P.K` the vector of parameter values for probabilities of knowledge states.
- `beta` the vector of parameter values for probabilities of a careless error.
- `eta` the vector of parameter values for probabilities of a lucky guess.
- `betafix, etafix` vectors of fixed error and guessing parameter values; `NA` indicates a free parameter.

Details

This is a draft version. It may change in future releases.

Value

The Jacobian matrix. The number of rows equals 2^(number of items) - 1, the number of columns equals the number of independent parameters in the model.

References

Heller, J. (2017). Identifiability in probabilistic knowledge structures. *Journal of Mathematical Psychology*, **77**, 46–57. doi: 10.1016/j.jmp.2016.07.008

Stefanutti, L., Heller, J., Anselmi, P., & Robusto, E. (2012). Assessing the local identifiability of probabilistic knowledge structures. *Behavior Research Methods*, **44**(4), 1197–1211. doi: 10.3758/s134280120187z

See Also

`blim`, `simulate.blim`, `gradedness`.

Examples

```
data(endm)
m <- blim(endm$K2, endm$N.R)

## Test of identifiability
J <- jacobian(m)
dim(J)
qr(J)$rank
```

---

plot.blim *Diagnostic Plot for Basic Local Independence Models*

Description

Plots BLIM residuals against fitted values.

Usage

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

Arguments

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

Details

The deviance residuals are plotted against the predicted response probabilities for each response pattern.

See Also

`blim`, `residuals.blim`.

Examples

```
## Compare MD and MDML estimation

data(DoignonFalmagne7)
blim1 <- blim(DoignonFalmagne7$K, DoignonFalmagne7$N.R, method="MD")
blim2 <- blim(DoignonFalmagne7$K, DoignonFalmagne7$N.R, method="MDML")

par(mfrow = 1:2)      # residuals versus fitted values
plot(blim1, main = "MD estimation",   ylim = c(-4, 4))
plot(blim2, main = "MDML estimation", ylim = c(-4, 4))
```

`print.blim` *Print a blim Object*

Description

Prints the output of a `blim` model object.

Usage

```
## S3 method for class 'blim'
print(x, P.Kshow = FALSE, errshow = TRUE,
      digits=max(3, getOption("digits") - 2), ...)
```

Arguments

`x`	an object of class `blim`, typically the result of a call to `blim`.
`P.Kshow`	logical, should the estimated distribution of knowledge states be printed?
`errshow`	logical, should the estimates of careless error and lucky guess parameters be printed?
`digits`	a non-null value for `digits` specifies the minimum number of significant digits to be printed in values.
`...`	further arguments passed to or from other methods. None are used in this method.

Value

Returns the `blim` object invisibly.

See Also

`blim`.

Examples

```
data(DoignonFalmagne7)

blim1 <- blim(DoignonFalmagne7$K, DoignonFalmagne7$N.R)
print(blim1, showP.K = TRUE)
```

`probability` *Problems in Elementary Probability Theory*

Description

This data set contains responses to problems in elementary probability theory observed before and after some instructions (the so-called learning object) were given. Data were collected both in the lab and via an online questionnaire. Of the 1127 participants eligible in the online study, 649 were excluded because they did not complete the first set of problems (p101, ..., p112) or they responded too quickly or too slowly. Based on similar criteria, further participants were excluded for the second set of problems, indicated by missing values in the variables b201, ..., b212. Problems were presented in random order.

Participants were randomized to two conditions: an enhanced learning object including instructions with examples and a basic learning object without examples. Instructions were given on four concepts: how to calculate the classic probability of an event (pb), the probability of the complement of an event (cp), of the union of two disjoint events (un), and of two independent events (id).

The questionnaire was organized as follows:

Page 1 Welcome page.

Page 2 Demographic data.

Page 3 First set of problems.

Page 4 to 8 Instructions (learning object).

Page 9 Second set of problems.

Page 10 Feedback about number of correctly solved problems.

Usage

```
data(probability)
```

Format

A data frame with 504 cases and 68 variables:

- `case` a factor giving the case id, a five-digits code the fist digit denoting lab or online branch of the study, the last four digits being the case number.
- `lastpage` Which page of the questionnaire was reached before quitting? The questionnaire consisted of ten pages.
- `mode` a factor; `lab` or `online` branch of study.
- `started` a timestamp of class POSIXlt. When did participant start working on the questionnaire?
- `sex` a factor coding sex of participant.
- `age` age of participant.
- `educat` education as a factor with three levels: `1` secondary school or below; `2` higher education entrance qualification; `3` university degree.

- `fos` field of study. Factor with eight levels: `ecla` economics, business, law; `else` miscellaneous; `hipo` history, politics; `lang` languages; `mabi` mathematics, physics, biology; `medi` medical science; `phth` philosophy, theology; `psco` psychology, computer science, cognitive science.
- `semester` ordered factor. What semester are you in?
- `learnobj` a factor with two levels: `enhan` learning object enhanced with examples; `basic` learning object without examples.

The twelve problems of the first part (before the learning object):

- `p101` A box contains 30 marbles in the following colors: 8 red, 10 black, 12 yellow. What is the probability that a randomly drawn marble is yellow? (Correct: 0.40)
- `p102` A bag contains 5-cent, 10-cent, and 20-cent coins. The probability of drawing a 5-cent coin is 0.35, that of drawing a 10-cent coin is 0.25, and that of drawing a 20-cent coin is 0.40. What is the probability that the coin randomly drawn is not a 5-cent coin? (0.65)
- `p103` A bag contains 5-cent, 10-cent, and 20-cent coins. The probability of drawing a 5-cent coin is 0.20, that of drawing a 10-cent coin is 0.45, and that of drawing a 20-cent coin is 0.35. What is the probability that the coin randomly drawn is a 5-cent coin or a 20-cent coin? (0.55)
- `p104` In a school, 40% of the pupils are boys and 80% of the pupils are right-handed. Suppose that gender and handedness are independent. What is the probability of randomly selecting a right-handed boy? (0.32)
- `p105` Given a standard deck containing 32 different cards, what is the probability of not drawing a heart? (0.75)
- `p106` A box contains 20 marbles in the following colors: 4 white, 14 green, 2 red. What is the probability that a randomly drawn marble is not white? (0.80)
- `p107` A box contains 10 marbles in the following colors: 2 yellow, 5 blue, 3 red. What is the probability that a randomly drawn marble is yellow or blue? (0.70)
- `p108` What is the probability of obtaining an even number by throwing a dice? (0.50)
- `p109` Given a standard deck containing 32 different cards, what is the probability of drawing a 4 in a black suit? (Responses that round to 0.06 were considered correct.)
- `p110` A box contains marbles that are red or yellow, small or large. The probability of drawing a red marble is 0.70 (lab: 0.30), the probability of drawing a small marble is 0.40. Suppose that the color of the marbles is independent of their size. What is the probability of randomly drawing a small marble that is not red? (0.12, lab: 0.28)
- `p111` In a garage there are 50 cars. 20 are black and 10 are diesel powered. Suppose that the color of the cars is independent of the kind of fuel. What is the probability that a randomly selected car is not black and it is diesel powered? (0.12)
- `p112` A box contains 20 marbles. 10 marbles are red, 6 are yellow and 4 are black. 12 marbles are small and 8 are large. Suppose that the color of the marbles is independent of their size. What is the probability of randomly drawing a small marble that is yellow or red? (0.48)

The twelve problems of the second part (after the learning object):

- `p201` A box contains 30 marbles in the following colors: 10 red, 14 yellow, 6 green. What is the probability that a randomly drawn marble is green? (0.20)

- `p202` A bag contains 5-cent, 10-cent, and 20-cent coins. The probability of drawing a 5-cent coin is 0.25, that of drawing a 10-cent coin is 0.60, and that of drawing a 20-cent coin is 0.15. What is the probability that the coin randomly drawn is not a 5-cent coin? (0.75)
- `p203` A bag contains 5-cent, 10-cent, and 20-cent coins. The probability of drawing a 5-cent coin is 0.35, that of drawing a 10-cent coin is 0.20, and that of drawing a 20-cent coin is 0.45. What is the probability that the coin randomly drawn is a 5-cent coin or a 20-cent coin? (0.80)
- `p204` In a school, 70% of the pupils are girls and 10% of the pupils are left-handed. Suppose that gender and handedness are independent. What is the probability of randomly selecting a left-handed girl? (0.07)
- `p205` Given a standard deck containing 32 different cards, what is the probability of not drawing a club? (0.75)
- `p206` A box contains 20 marbles in the following colors: 6 yellow, 10 red, 4 green. What is the probability that a randomly drawn marble is not yellow? (0.70)
- `p207` A box contains 10 marbles in the following colors: 5 blue, 3 red, 2 green. What is the probability that a randomly drawn marble is blue or red? (0.80)
- `p208` What is the probability of obtaining an odd number by throwing a dice? (0.50)
- `p209` Given a standard deck containing 32 different cards, what is the probability of drawing a 10 in a red suit? (Responses that round to 0.06 were considered correct.)
- `p210` A box contains marbles that are green or red, large or small The probability of drawing a green marble is 0.40, the probability of drawing a large marble is 0.20. Suppose that the color of the marbles is independent of their size. What is the probability of randomly drawing a large marble that is not green? (0.12)
- `p211` In a garage there are 50 cars. 15 are white and 20 are diesel powered. Suppose that the color of the cars is independent of the kind of fuel. What is the probability that a randomly selected car is not white and it is diesel powered? (0.28)
- `p212` A box contains 20 marbles. 8 marbles are white, 4 are green and 8 are red. 15 marbles are small and 5 are large. Suppose that the color of the marbles is independent of their size. What is the probability of randomly drawing a large marble that is white or green? (0.15)

Further variables:

- `time01`, ..., `time10` the time (in s) spent on each page of the questionnaire. In the lab branch of the study, participants started directly on Page 2.
- `b101`, ..., `b112` the twelve problems of the first part coded as correct (1) or error (0).
- `b201`, ..., `b212` the twelve problems of the second part coded as correct (1) or error (0).

Source

Data were collected by Pasquale Anselmi and Florian Wickelmaier at the Department of Psychology, University of Tuebingen, in February and March 2010.

Examples

```
data(probability)

## "Completer" sample
```

```
pb <- probability[!is.na(probability$b201), ]

## Response frequencies for first and second part
N.R1 <- as.pattern(pb[, sprintf("b1%.2i", 1:12)], freq = TRUE)
N.R2 <- as.pattern(pb[, sprintf("b2%.2i", 1:12)], freq = TRUE)

## Conjunctive skill function, one-to-one problem function
sf1 <- read.table(header = TRUE, text = "
  item cp id pb un
     1  0  0  1  0
     2  1  0  0  0
     3  0  0  0  1
     4  0  1  0  0
     5  1  0  1  0
     6  1  0  1  0
     7  0  0  1  1
     8  0  0  1  1
     9  0  1  1  0
    10  1  1  0  0
    11  1  1  1  0
    12  0  1  1  1
")

## Extended skill function
sf2 <- rbind(sf1, read.table(header = TRUE, text = "
  item cp id pb un
     2  0  0  0  1
     3  1  0  0  0
     6  0  0  1  1
     7  1  0  1  0
    12  1  1  1  0
"))

## Delineated knowledge structures
K1 <- delineate(sf1)$K
K2 <- delineate(sf2)$K

## After instructions, fit of knowledge structures improves
sapply(list(N.R1, N.R2), function(n) blim(K1, n)$discrepancy)
sapply(list(N.R1, N.R2), function(n) blim(K2, n)$discrepancy)
```

`residuals.blim` *Residuals for Basic Local Independence Models*

Description

Computes deviance and Pearson residuals for `blim` objects.

Usage

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

Arguments

`object`	an object of class `blim`, typically the result of a call to `blim`.
`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

See `residuals.glm` for details.

Value

A named vector of residuals having as many elements as response patterns.

See Also

`blim`, `residuals.glm`, `plot.blim`.

Examples

```
data(DoignonFalmagne7)
blim1 <- blim(DoignonFalmagne7$K, DoignonFalmagne7$N.R)

sum( resid(blim1)^2 )              # likelihood ratio G2
sum( resid(blim1, "pearson")^2 )   # Pearson X2
```

`simulate.blim` *Simulate Responses from Basic Local Independence Models (BLIMs)*

Description

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

Usage

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

Arguments

`object`	an object of class `blim`, typically the result of a call to `blim`.
`nsim`	currently not used.
`seed`	currently not used.
`...`	further arguments passed to or from other methods. None are used in this method.

Details

Responses are simulated in two steps: First, a knowledge state is drawn with probability `P.K`. Second, responses are generated by applying `rbinom` with probabilities computed from the model object's `beta` and `eta` components.

Value

A named vector of frequencies of response patterns.

See Also

`blim`, `endm`.

Examples

```
data(DoignonFalmagne7)

blim1 <- blim(DoignonFalmagne7$K, DoignonFalmagne7$N.R)

simulate(blim1)

## Parametric bootstrap
disc <- replicate(200, blim(blim1$K, simulate(blim1))$discrepancy)

hist(disc, col = "lightgray", border = "white", freq = FALSE, breaks = 20,
     main = "Parametric bootstrap", xlim = c(.05, .3))
abline(v = blim1$discrepancy, lty = 2)

## See ?endm for further examples.
```

`Taagepera` *Responses and Knowledge Structures from Taagepera et al. (1997)*

Description

Taagepera et al. (1997) applied knowledge space theory to specific science problems. The density test was administered to 2060 students, the conservation of matter test to 1620 students. A subtest of five items each is included here. The response frequencies were reconstructed from histograms in the paper.

Usage

```
data(Taagepera)
```

Format

Two lists, each consisting of two components:

`density97` a list with components `K` and `N.R` for the density test.

`matter97` a list with components `K` and `N.R` for the conservation of matter test.

`K` a state-by-problem indicator matrix representing the hypothetical knowledge structure. An element is one if the problem is contained in the state, and else zero.

`N.R` a named numeric vector. The names denote response patterns, the values denote their frequencies.

Source

Taagepera, M., Potter, F., Miller, G.E., & Lakshminarayan, K. (1997). Mapping students' thinking patterns by the use of knowledge space theory. *International Journal of Science Education*, **19**(3), 283–302. doi: 10.1080/0950069970190303

Examples

```
data(Taagepera)
density97$K     # density test knowledge structure
density97$N.R   # density test response patterns
matter97$K      # conservation of matter knowledge structure
matter97$N.R    # conservation of matter response patterns
```

Index