Package 'irtProb'

April 17, 2009

Type Package

Title Utilities and Probability Distributions Related to Multidimensional Person Item Response Models (IRT)

Version 1.0

Date 2009-02-08

Encoding latin1

Author Gilles Raiche

Maintainer Gilles Raiche <Raiche.Gillest@uqam.ca>

Description The irtProb package was mainly developped to compute IRT probability distributions.

Depends R (>= 2.0.0), lattice, methods, stats, moments

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2009-02-09 08:07:00

R topics documented:

`irtProb-package` *Utilities and Probability Distributions Related to Multidimensional Person Item Response Models (IRT)*

Description

The `irtProb` package was mainly developped to compute probability distributions in the context of Item Response Theory (IRT). Actually two families of models are taken into account. The first is the family of 1, 2, 3 and 4 parameters logistic fonctions. The second is a new logistic family adding of 1, 2, 3 and 4 person parameters. With `irtProb` some utilitarian functions are also available. So it is possible to generate response patterns with each family of item response models and other functions are also available to do conversion of item parameters between classical test theory and item response theory (2PL). Maximum likelihood and Maximum a posteriori estimation function of the multidimensional person paramaters are also available.

Details

Package:	irtProb
Type:	Package
Version:	1.0
Date:	2007-08-28
Depends:	R (>= 2.0.0), lattice, methods, moments
License:	GPL
LazyLoad:	yes

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Bartholomew, D. J., Steele, F., Moustaki, I. and Galbraith, J. I. (2000). *The analysis and interpretation of multivariate data for social scientists*. Boca Raton, California: Chapman and Hall.

Carroll, J. B. (1945). The effect of difficulty and chance success on correlations between items or between tests. *Psychometrika, 10*(1), 1-19.

Freund, J. E. and Wallpole, R. E. (1980). *Mathematical statistics, 3rd edition*. Englewood Cliffs, New Jersey: Prentice-Hall.

Guilford, J. P. (1936). The determination of item difficulty when chance success is a factor. *Psychometrika, 1*(4), 259-264.

Guilford, J. P. (1954). *Psychometric methods, 2d edition*. New York, New jersey: McGraw-Hill.

Hambleton, R. K. and Swaminathan, H. (1985). *Item response theory - Principles and applications*. Boston, Massachuset: Kluwer.

Lord, F. M. (1944). Reliability of multiple-choice tests as a function of number of choice per item. *Journal of educational psychology, 35*(3), 175-180.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Lord, F. M. and Novick, M. R. (1968). *Statistical theories of mental test scores, 2nd edition*. Reading, Massacusett: Addison-Wesley.

Plumlee, L B. (1952). The effect of difficulty and chance success on item-test correlations and on test reliability. *Psychometrika, 17*(1), 69-86.

Plumlee, L B. (1954). The predicted and observed effect of chance success on multiple-choice test validity. *Psychometrika, 19*(1), 65-70.

See Also

Other packages are also very useful to manipulate IRT objects. The R psychometric view is instructive at this point. See `http://cran.stat.sfu.ca/web/views/Psychometrics.html` for further details.

`4pl` *One, Two, Three and Four Parameters Logistic Distributions*

Description

Density, distribution function, quantile function and random generation for the one, two, three and four parameters logistic distributions.

Usage

```
p4pl(theta = 0, a = 1, b = 0, c = 0, d = 1, lower.tail = TRUE, log.p = FALSE)

d4pl(theta = 0, a = 1, b = 0, c = 0, d = 1,                     log.p = FALSE)

q4pl(p   = 0.05, a = 1, b = 0, c = 0, d = 1, lower.tail = TRUE, log.p = FALSE)

r4pl(N   = 100,  a = 1, b = 0, c = 0, d = 1)
```

Arguments

N	numeric; number of observations.
p	numeric; vector of probability.
theta	numeric; vector of person proficiency levels scaled on a normal z score.
a	numeric; positive vector of item discrimination parameters.
b	numeric; vector of item difficulty parameters.
c	numeric; positive vector of item pseudo-guessing parameters (a probability between 0 and 1).
d	numeric; positive vector of item inattention parameters (a probability between 0 and 1).
lower.tail	logical; if TRUE (default), probabilities are $P(X_j <= x_{ij})$, otherwise, $P(X_j > x_{ij})$.
log.p	logical; if TRUE probabilities p are given as log(p).

Details

The 4 parameters logistic distribution (cdf) is equal to:

$$P(x_{ij} = 1|\theta_j, a_i, b_i, c_i, d_i) = c_i + \frac{d_i - c_i}{1 + e^{-Da_i(\theta_j - b_i)}},$$

where the parameters are defined in the section arguments and *i* and *j* are respectively the items and the persons indices. A normal version of the 4PL model was described by McDonald (1967, p. 67), Barton and Lord (1981), like Hambleton and Swaminathan (1985, p. 48-50).

Value

p4pl	numeric; gives the distribution function (cdf).
d4pl	numeric; gives the density (derivative of p4pl).
q4pl	numeric; gives the quantile function (inverse of p4pl).
r4pl	numeric; generates theta random deviates.

Note

Code inspired by the `pnorm` function structure from the R base package.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Barton, M. A. and Lord, F. M. (1981). *An upper asymptote for the tree-parameter logistic item-response model*. Research Bulletin 81-20. Princeton, NJ: Educational Testing Service.

Hambleton, R. K. and Swaminathan, H. (1985). *Item response theory - Principles and applications*. Boston, Massachuset: Kluwer.

Lord, F. M. and Novick, M. R. (1968). *Statistical theories of mental test scores, 2nd edition*. Reading, Massacusett: Addison-Wesley.

McDonald, R. P. (1967). Non-linear factor analysis. *Psychometric Monographs, 15*.

See Also

gr4pl, ggr4pl, ctt2irt, irt2ctt

Examples

```
## ..........................................................................
# probability of a correct response
 p4pl(theta = 3, b = 0)

# Verification of the approximation of N(0,1) by a logistic (D=1.702)
 a <- 1; b <- 0; c <- 0; d <- 1; theta <- seq(-4, 4, length = 100)

# D constant 1.702 gives an approximation of a N(0,1) by a logistic
 prob.irt  <- p4pl(theta, a*1.702, b, c, d)
 prob.norm <- pnorm(theta, 0, 1)
 plot(theta, prob.irt)
 lines(theta, prob.norm, col = "red")

# Maximal difference between the two functions: less than 0.01
 max(prob.irt - prob.norm)

# Recovery of the value of the probability of a correct response p4pl() from
# the quantile value q4pl()
 p4pl(theta = q4pl(p = 0.20))

# Recovery of the quantile value from the probability of a correct response
 q4pl(p=p4pl(theta=3))

# Density Functions [derivative of p4pl()]
 d4pl(theta = 3, a = 1.702)
 theta   <- seq(-4, 4, length = 100)
 a       <- 3.702; b <- 0; c <- 0; d <- 1
 density <- d4pl(theta = theta, a = a, b = b, c = c, d = d)
 label   <- expression("Density - First Derivative")
 plot(theta, density, ylab = label, col = 1, type = "l")
 lines(theta, dnorm(x = theta, sd = 1.702/a), col = "red", type = "l")

## Generation of proficiency levels from r4pl() according to a N(0,1)
 data <- (r4pl(N = 10000, a = 1.702, b = 0, c = 0, d = 0))
 c(mean = mean(data), sd = sd(data))
## ..........................................................................
```

checkAdequation *Function to Check the Adequation of the Second Derivatives*

Description

Check the adequation of the second derivatives (hessian) for computation of variance and covariance. Use inside the function `m4plEstimateMore` to assure that computations can be applied to this second derivative. Problems encountered can be of not positive definitetess, singular matrix, diagonal not complitely positive. In these cases inversion of the matrix is not posssible or variances cannot be computed from the inverse of the diagonal.

Usage

```
checkAdequation(x)
```

Arguments

`x`	matrix: second derivative (hessian).

Value

`isNumbers`	logical: check if the elements of the x matrix are all numerics.
`correctClass`	logical: check if the the x matrix is of class `matrix`.
`squareMatrix`	logical: check if the the x matrix is square.
`diagPositive`	logical: check if the elements of the x matrix are all positive and > 0.
`positiveDefinite`	logical: check if the the x matrix is positiveDefinite.
`nonSingular`	logical: check if the the x matrix is nonSingular.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Seber, G. E. A. F. (2008). *A matrix handbook for statisticians*. New York, New Jersey: Wiley.

See Also

`eigen`, `det`, `diag`

Examples

```
## Complete adequation of the matrix
## ..........................................................
 x <- matrix(c(4.867054, 16.66902, 16.669023, 107.36390), ncol=2)
 checkAdequation(x)
 all(checkAdequation(x))
 eigen(x)$values
 det(x)
 diag(x)
## ..........................................................

## Not positiveDefinite matrix
## ..........................................................
 x <- matrix(1:4, ncol=2)
 checkAdequation(x)
 all(checkAdequation(x))
 eigen(x)$values
 det(x)
 diag(x)
## ..........................................................

## More problems
## ..........................................................
 x <- matrix(c("Inf",2,5,10), ncol=2)
 checkAdequation(x)
 all(checkAdequation(x))
 det(x)
 diag(x)
# eigen(x)$values
 x <- matrix(c("NaN",2,5,10), ncol=2)
 checkAdequation(x)
 all(checkAdequation(x))
 det(x)
 diag(x)
# eigen(x)$values
## ..........................................................
```

`conversion` *Functions to Convert from and to CTT and IRT Models*

Description

`ctt2irt` and `irt2ctt` are converter functions to change the parametrization of item parameters from and to classical test theory (difficulty and discrimination parameters) and item response theory (difficulty and discrimination parameters). Consequently, the conversion is only valid between ctt and 2 parameters logistic or normal models.

Usage

```
ctt2irt(rpbis = 0.7071068, difficulty = 0.5)

irt2ctt(a = 1, b = 0, c = 0, d = 1, model = "LOGISTIC")
```

Arguments

`rpbis`	numeric; vector of discrimination parameters: point biserial correlation between the item response and the total score.
`difficulty`	vector of difficulty parameters: proportion of corrected responses.
`a`	numeric; vector of discrimination parameters.
`b`	numeric; vector of difficulty parameters.
`c`	numeric; vector of pseudo-guessing parameters (not used for the moment).
`d`	numeric; vector of inattention parameters (not used for the moment).
`model`	character; if NORMAL the constant D (1.702) is used. Default to LOGISTIC with constant D=1.

Details

Eventually the 3 and 4 parameters logistic and normal models will be taken in account according to Urry approximation (1974).

Value

`For ctt2irt`	....
`note`	character; warnings about the use of the `c` and `d` item parameters.
`normal.parameters`	numeric; vector returning difficulty `b` and discrimination `a` parameters from the normal model.
`irt.parameters`	numeric; vector returning difficulty `b` and discrimination `a` parameters from the logistic model.
`For irt2ctt`	....
`note`	character; warnings about the use of the `c` and `d` item parameters.
`normal.parameters`	numeric; vector returning difficulty `p` and discrimination `rpbis` parameters from the normal model.
`irt.parameters`	numeric; vector returning difficulty `p` and discrimination `rpbis` parameters from the logistic model.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Lord, F. M. and Novick, M. R. (1968). *Statistical theories of mental test scores, 2nd edition*. Reading, Massacusett: Addison-Wesley.

Urry, V. W. (1974). Approximations to item parameters of mental tests models and their uses. *Educational and psychological measurement, 34*, 253-269.

See Also

`gr4pl`, `ggr4pl`, `ctt2irt`, `irt2ctt`

Examples

```
## ..............................................................
# Values of p and rbis according to de a, b, c and d values
#  MODEL means that item parameters are from a NORMAL or LOGISTIC model type
 irt2ctt()
 nItems <- 5
 b      <- seq(-3, 3, length=nItems)
 a      <-  rep(1, nItems)
 c      <-  rep(0, nItems)
 d      <-  rep(1, nItems)

# Difference between classical item parameters and IRT ones
 irt2ctt(b=b,a=a,c=c,d=d,model="LOGISTIC")
 irt2ctt(b=b,a=a,c=c,d=d,model="NORMAL")

# Values of a and b according p and rpbis
 ctt2irt()

# Verification of the recovery of original ctt item parameters
 nItems <- 5
 p      <- seq(0.10, 0.90, length=nItems)
 rpbis  <- seq(0.50, 0.95, length=nItems)
 irt    <- ctt2irt(dif=p,rpbis=rpbis)
 clas   <- irt2ctt(b=irt$normal[2],a=irt$normal[1],model="NORMAL")
 round(c(NORMAL=irt$normal, IRT=irt$irt, CTT=clas$normal), 3)
## ..............................................................
```

`derivatives` *First and Second Derivatives*

Description

Three points first and second derivatives numerical approximation used with the `m4plEstimateMore` function.

Usage

```
fprime( x, FUN = "FP", h = 0.001, names = paste("x", c(1:length(x)), sep = ""))
fsecond(x, FUN = "FP", h = 0.001, names = paste("x", c(1:length(x)), sep = ""))
```

Arguments

`x`	numeric; vector of values at which the derivation is to be done.
`FUN`	function; function to derive.
`h`	numeric; neighbouring value.
`names`	character; names given to each results. DEFAULT to the existing names of the vector x.

Details

This function could be used for numerical derivation in general, but is dedicated to be used internally by `m4plEstimateMore`. For other general purposes `D`, `deriv` and `deriv3` are preferred.

Value

`fprime`	numeric; vector of first derivatives.
`fsecond`	numeric; matrix of second derivatives.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Press, W. H., Vetterling, W. T., Teukolsky, S. A. and Flannery, B. P. (2002). *Numerical recipees in C++. The art of scientific computing, 2nd edition*. Cambridge, United Kingdom: Cambridge University press.

Yakowitz, S. and Szidarovszky, F. (1986). *An introduction to numerical computations*. New York, New Jersey: MacMilla.

See Also

`D`, `deriv` and `deriv3`

Examples

```
## ....................................
 test  <- function(x) 2*x+5
  test(   x=0)
  fprime( x=0, FUN=test)
  fsecond(x=0, FUN=test)
## ....................................
 test2 <- function(x) 2*(x[1]*x[2])+5
  test2(  x=c(0,0))
  fprime( x=c(0,0), FUN=test2)
  fsecond(x=c(0,0), FUN=test2)
```

`graphics` *Graphic Functions to Illustrate Response Curves and Parameter Estimation*

Description

Graphic functions to illustrate response curves and parameter estimation.

Usage

```
PCC(theta  = 0, S = 0, C = 0, D = 0,
    s      = 1/1.702, b = seq(-5, 5, length = 300), c = 0, d = 1,
    groups = TRUE, ID = "ID",
    main   = "Person Characteristic Curve",
    xlab   = "Item Difficulty Parameter (b)", ylab = "P(x = 1)",
    type   = c("g", "a"))
```

Arguments

`theta`	numeric; vector of person proficiency (θ) levels scaled on a normal `z` score.
`S`	numeric: positive vector of personal fluctuation parameters (σ).
`C`	numeric: positive vector of personal pseudo-guessing parameters (χ, a probability between 0 and 1).
`D`	numeric: positive vector of personal inattention parameters (δ, a probability between 0 and 1).
`s`	numeric: vector of item fluctuation parameter or the inverse of item discrimination (s= 1/a).
`b`	numeric: vector of item discrimination parameter.
`c`	numeric: vector of item pseudo-guessing parameter.
`d`	numeric: vector of item inattention parameter.
`ID`	character: curves identification information displayed ("ID", "ALL", "THETA2 or NULL)

`groups` logical: default to `TRUE`. If `TRUE`, Lattice `xyplot` by groups. If `FALSE`, `xyplot` with shingles.

`main` character: first line of main title.

`xlab` character: label of x axis.

`ylab` character: label of y axis.

`type` character: type of `xyplot` graphic. One of the following: "p", "l", "h", "b", "o", "s", "S", "r", "a", "g", "smooth".

Value

`PCC` returns a list:

`graphic` trellis object: figures for each subject (group or shingle representation).

`probability` data.frame: item snd person parameters, like th eprobability of a correct response.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

Examples

```
## PCC curves grouped on a single figure
 res1 <- PCC(theta=c(-2,-2,-2),S=0, C=c(0.0, 0.1, 0.6), D=0.2,
             b=seq(-5,5,length=3000), ID=NULL, groups=TRUE, type=c("g","a"))
 res1

## PCC curves shingled on a single figure for each subject
 res2 <- PCC(theta=c(-2,-1,0),S=c(4.0,0.0, 1.0), C=c(0.0, 0.1, 0.6), D=0.2,
             b=seq(-5,5,length=3000), ID=NULL, groups=FALSE, type=c("g","a"))
 res2
```

`likelihoodCurve` *Functions to Graph m4pl Likelihood Curves*

Description

`likelihoodCurve` and `groupLikelihoodCurve` are used to graph the likelihood function curves according to the only theta, theta anc pseudo-guessing, theta and fluctuation, like theta and inattention m4pl models: only two simultaneous person parameters are taken in account.

Usage

```
likelihoodCurve(x, s, b, c, d,
                limitT = c(min = -4, max = 4), limitS = c(min = 0, max = 4),
                limitC = c(min = 0, max = 1), limitD = c(min = 0, max = 1),
                grain         = 150, annotate = TRUE,
                logLikelihood = FALSE, color = TRUE,
                main          = "Likelihood Curve",
                xlab          = expression(theta), ylab = NULL, zlab = "P(X)",
                type          = "levelplot", m = 0)

groupLikelihoodCurves(plotT, plotS, plotC, plotD, main=NULL, cex=0.7)
```

Arguments

`x`	numeric: binary (0,1) response pattern.
`s`	numeric: vector of inverse a discrimination item parameters.
`b`	numeric: vector of b difficulty item parameters.
`c`	numeric: vector of c pseudo-guessing item parameters.
`d`	numeric: vector of d inattention item parameters.
`limitT`	numeric: minimum and maximum of the proficiency person parameter used for the x axis.
`limitS`	numeric: minimum and maximum of the fluctuation person parameter used for the y axis.
`limitC`	numeric: minimum and maximum of the pseudo-guessing person parameter used for the y axis.
`limitD`	numeric: minimum and maximum of the inattention person parameter used for the y axis.
`grain`	numeric: number of theta values used to compute pattern distribution probability.
`annotate`	logical: does annotation is applied to the graphs?
`logLikelihood`	numeric: data.frame of the log likelihood of the studied models.
`color`	logical: does color is applied to contourplot or wireframe.
`main`	character: main title.
`xlab`	character: x axis label.
`ylab`	character: y axis label.
`zlab`	character: z axis label.
`type`	character: type of 3D plot ("levelplot", "contourplot" or "wireframe").
`m`	numeric: mean of the a priori probability distribution.
`plotT`	trellis: 2D theta likelihood curve.
`plotS`	trellis: 3D theta * S likelihood curve.

`plotC`	trellis: 3D theta * C likelihood curve.
`plotD`	trellis: 3D theta * D likelihood curve.
`cex`	numeric: zaxis label size.

Value

`likelihoodCurve`

`plotT`	trellis: theta likelihood functions curves.
`plotS`	trellis: theta * S likelihood functions curves.
`plotC`	trellis: theta * C likelihood functions curves.
`plotD`	trellis: theta * D likelihood functions curves.
`parameters`	numeric: list of data.frame of person parameters for each model studied. Each element of the list shows estimation with different a priori probability distributions (uniform, normal and none).
`logLikelihood`	numeric: data.frame of the log likelihood for each model studied.
`graphic`	graphic: all the likelihood functions curves are displayed.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

Examples

```
## SIMULATION OF A RESPONSE PATTERN WITH 60 ITEMS
nItems <- 60
a      <- rep(1.702,nItems); b <- seq(-4,4,length=nItems)
c      <- rep(0,nItems);     d <- rep(1,nItems)
nSubjects <-  1
theta     <-  -1
S         <-  0.0
C         <-  0.5
D         <-  0.0

set.seed(seed = 100)
x         <- ggrm4pl(n=nItems, rep=1,
                     theta=theta, S=S, C=C, D=D,
                     s=1/a, b=b,c=c,d=d)

## Likelihood curves, person parameters estimates
 # and log likelihood of models graphed
test <- likelihoodCurve(x=x, s=1/a, b=b, c=c, d=d, color=TRUE,
                        main="Likelihood Curve",
                        xlab=expression(theta), ylab=NULL, zlab="P(X)",
                        type="wireframe" , grain=50, limitD=c(0,1),
```

```
                            logLikelihood=FALSE, annotate=TRUE )

# Contentd of the object test
test$plotT
test$plotC
test$plotS
test$plotD
test$par
round(test$logLikelihood,2)

## Graph of all the likelihood function curves
groupLikelihoodCurves(test$plotT, test$plotS, test$plotC, test$plotD,
                      main=NULL, cex=0.7)
```

---

m4pl — *Multidimensional One, Two, Three and Four Person Parameters Logistic Distributions*

Description

Density, distribution function, quantile function and random generation for the multidimensional one, two, three and four person parameters logistic distributions.

Usage

```
pm4pl(theta = 0, S = 0, C = 0, D = 0, s = 1/1.702, b = 0, c = 0, d = 1,
      lower.tail = TRUE, log.p = FALSE)

dm4pl(theta = 0, S = rep(0,length(theta)), C = rep(0,length(theta)),
      D = rep(0,length(theta)), b = 0, s = 1/1.702, c = 0, d = 1, log.p = FALSE)

qm4pl(p = 0.05, S = 0, C = 0, D = 0, s = 1/1.702, b = 0, c = 0, d = 1,
      lower.tail = TRUE, log.p = FALSE)

rm4pl(N = 100, S = 0, C = 0, D = 0, s = 1/1.702, b = 0, c = 0, d = 1)
```

Arguments

`theta`	numeric; vector of person proficiency (θ) levels scaled on a normal `z` score.
`S`	numeric; positive vector of personal fluctuation parameters (σ).
`C`	numeric; positive vector of personal pseudo-guessing parameters (χ, a probability between 0 and 1).
`D`	numeric; positive vector of personal inattention parameters (δ, a probability between 0 and 1).
`N`	numeric; number of observations.

`p`	numeric; vector of probability.
`s`	numeric; positive vector of item fluctuation parameters.
`b`	numeric; vector of item difficulty parameters.
`c`	numeric; positive vector of item pseudo-guessing parameters (a probability between 0 and 1).
`d`	numeric; positive vector of item inattention parameters (a probability between 0 and 1).
`lower.tail`	logical; if TRUE (default), probabilities are $P(X_j <= x_{ij})$, otherwise, $P(X_j > x_{ij})$.
`log.p`	logical; if TRUE, probabilities p are given as log(p).

Details

The multidimensional 4 persons parameters logistic distribution (cdf) is equal to:

$$P(x_{ij} = 1|\theta_j, \sigma_j, \chi_j, \delta_j, s_i, b_i, c_i, d_i) = (\chi_j + c_i) + \frac{(d_i - \delta_j) - (\chi_j + c_i)}{1 + e^{\frac{-Da_i(\theta_j - b_i)}{\sqrt{\sigma_j^2 + s_i^2}}}}$$

where the parameters are defined in the section arguments and *i* and *j* are respectively the items and the persons indices. The σ_j, χ_j and δ_j parameters are respectively the personnal fluctuation, pseudo-guessing and inattention parameters. The multidimensional 4 persons parameters logistic model (M4PL) was described by Raiche, Magis and Blais (2008; Raiche, Magis and Beland, 2009; Raiche, Blais and Magis, 2009).

Value

`pm4pl`	numeric; gives the distribution function (`cdf`).
`dm4pl`	numeric; gives the density (derivative of p4pl).
`qm4pl`	numeric; gives the quantile function (inverse of p4pl).
`rm4pl`	numeric; generates theta random deviates.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Blais, J.-G., Raiche, G. and Magis, D. (2009). La detection des patrons de reponses problematiques dans le contexte des tests informatises. *In* Blais, J.-G. (Ed.): *Evaluation des apprentissages et technologies de l'information et de la communication : enjeux, applications et modeles de mesure*. Ste-Foy, Quebec: Presses de l'Universite Laval.

Raiche, G., Magis, D. and Beland, S. (2009). *La correction du resultat d'un etudiant en presence de tentatives de fraudes*. Communication presentee a l'Universite du Quebec a Montreal. Retrieved from http://www.camri.uqam.ca/camri/camriBase/

Raiche, G., Magis, D. and Blais, J.-G. (2008). *Multidimensional item response theory models integrating additional inattention, pseudo-guessing, and discrimination person parameters*. Communication at the annual international Psychometric Society meeting, Durham, New Hamshire. Retrieved from http://www.camri.uqam.ca/camri/camriBase/

See Also

grm4pl, ggrm4pl, pggrm4pl

Examples

```
## ......................................................................
# Approximation of p4pl() by pm4pl()
 theta  <- 0
 S      <- 0; C <- 0; D <- 0
 a      <- 1.702; s <- sqrt(1/a^2); b <- 0; c <- 0; d <- 1
 p.4pl  <-  p4pl(theta,a=a,b,c,d);
 p.m4pl <- pm4pl(theta,S,C,D,s,b,c,d)
 print(c(p4pl=p.4pl, m4pl=p.m4pl))

# Comparison of p4pl() and pm4pl() according to diverse values of theta
#  while person parameters vary
 theta  <- seq(-4,4,length=100); N <- length(theta)
 S      <- rep(0.9,N); C <- rep(0,N); D <- rep(0.9,N)
 a      <- 1.702; s <- 1/1.702
 p.4pl  <- p4pl(theta=theta, a=a, b=b, c=c, d=d)
 p.m4pl <- pm4pl(theta=theta, S=S, C=C, D=D, s=s, b=b, c=c, d=d)
 print(c(difference.maximale = max(p.4pl - p.m4pl)))
 round(rbind(theta=theta, p4pl=p.4pl, m4pl=p.m4pl, dif=p.4pl-p.m4pl),3)
 plot(theta,p.4pl,type="l", col="black", ylab="Probability")
 lines(theta,p.m4pl,col="red"); max(p.4pl-p.m4pl)

# Recovery of probability by quantile
 pm4pl(theta=3,b=0); pm4pl(theta=qm4pl(p=0.20))

# Density function dm4pl()
# Comparison 01 between d4pl() and dm4pl()
 theta  <- seq(-4,4, length=100)
 # theta <- 0 # We can also experiment with an unique value of theta
 N      <- length(theta)
 S      <- rep(0,N); C <- rep(0,N); D <- rep(0,N)
 a      <- 1.702
 d.4pl  <- d4pl(theta=theta,a=a)
 d.m4pl <- dm4pl(theta=theta,S=0,s=1/a)
 stats  <- round(cbind(theta=theta, d4pl=d.4pl, m4pl=d.m4pl, dif=d.4pl-d.m4pl),3)
 print(stats)
 print(max(d.4pl - d.m4pl))
 plot(theta,d.4pl,type="l", col="black", ylab="Density")
 lines(theta,d.m4pl,col="red")

# Comparison 02 between d4pl() and dm4pl()
 theta  <- seq(-4,4, length=10)
```

```
 N      <- length(theta)
 S      <- rep(.3,N); C <- rep(0,N); D <- rep(0,N)
 a      <- 1.702
 d.4pl  <- d4pl(theta=theta,a=a)
 d.m4pl <- dm4pl(theta=theta, S=S, C=C, D=D, s=1/a)
 stats  <- round(cbind(theta=theta, d4pl=d.4pl, m4pl=d.m4pl, dif=d.4pl-d.m4pl),3)
 print(stats)
 print(max(d.4pl - d.m4pl))
 plot(theta,d.4pl,type="l", col="black", ylab="Density")
 lines(theta,d.m4pl,col="red")

# Comparison of q4pl and qm4pl
#  followed by recovery of quantiles
 pm4pl(theta=1, ); p4pl(theta=1, a=1.702)
 qm4pl(p=0.99,); q4pl(p=0.99, a=1.702)
 qm4pl(pm4pl(theta=1)); q4pl(p4pl(theta=1, a=1.702), a=1.702)
 qm4pl(p=pm4pl(theta=3))
 qm4pl(p=seq(0.01,0.99, length=10))

# Generation of theta values by rm4pl()
# ... Exemple 01 - A 4pl() equivalent distribution must be recovered when a=1.702
 res   <- rm4pl(N=1000)
 stats <- c(mean=mean(res), sd=sd(res), skewness=skewness(res),
            kurtosis=3-kurtosis(res))
 print(stats)
 # pdf of this distribution
 theta   <- seq(-4,4,length=100); C=rep(0,N); D=rep(0,N)
 density <- dm4pl(theta,C=C,D=D)
 plot(theta, density, type="l")

# ... Exemple 02 - Distribution with D != 0
 require(moments)
 S     <- 1/1.702
 B     <- 0
 C     <- 0.00
 D     <- 0.90
 res   <- rm4pl(N=1000, S=S, C=C, D=D)
 stats <- c(mean=mean(res), sd=sd(res), skewness=skewness(res), kurtosis=3-kurtosis(res))
 print(stats)
 # pdf of this distribution
 theta   <- seq(-4,4,length=100)
 density <- dm4pl(theta, S=S, C=C, D=D)
 print(c(max=max(density)))
 plot(theta, density, type="l")
## ...............................................................
```

`m4plModelShow` *Results For Each Subject To Each Model*

Description

Show all the information about the estimation of all the possible m4pl models for each subjects.

Usage

```
m4plModelShow(x, ...)
```

Arguments

`x`	data.frame: a matrix of binary 0-1 item responses.
`...`	varying: parameters to be passed to the `m4plPersonParameters` function.

Value

`ID`	integer: subject identificator.
`MODEL`	charavter: model identification (T,TS,TC,TD,TSC,TSD,TCD or TSCD)
`LL`	numeric: loglikelihood.
`AIC`	numeric: Akaike information criteria.
`BIC`	numeric: Bayes (Schwartz) information criteria.
`T`	numeric: theta parameter value.
`SeT`	numeric: theta parameter theoretical standard error.
`S`	numeric: person fluctuation parameter value.
`SeS`	numeric: person fluctuation theoretical standard error
`C`	numeric: person pseudo-guessing parameter value.
`SeC`	numeric: person pseudo-guessing theoretical standard error
`D`	numeric: person inattention parameter value.
`SeD`	numeric: person inattention theoretical standard error

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Blais, J.-G., Raiche, G. and Magis, D. (2009). La detection des patrons de reponses problematiques dans le contexte des tests informatises. *In* Blais, J.-G. (Ed.): *Evaluation des apprentissages et technologies de l'information et de la communication : enjeux, applications et modeles de mesure*. Ste-Foy, Quebec: Presses de l'Universite Laval.

Raiche, G., Magis, D. and Beland, S. (2009). *La correction du resultat d'un etudiant en presence de tentatives de fraudes*. Communication presentee a l'Universite du Quebec a Montreal. Retrieved from http://www.camri.uqam.ca/camri/camriBase/

Raiche, G., Magis, D. and Blais, J.-G. (2008). *Multidimensional item response theory models integrating additional inattention, pseudo-guessing, and discrimination person parameters*. Communication at the annual international Psychometric Society meeting, Durham, New Hamshire. Retrieved from http://www.camri.uqam.ca/camri/camriBase/

See Also

m4plPersonParameters

Examples

```
## GENERATION OF VECTORS OF RESPONSES
 # NOTE THE USUAL PARAMETRIZATION OF THE ITEM DISCRIMINATION,
 # THE VALUE OF THE PERSONNAL FLUCTUATION FIXED AT 0,
 # AND THE VALUE OF THE PERSONNAL PSEUDO-GUESSING FIXED AT 0.30.
 # IT COULD BE TYPICAL OF PLAGIARISM BEHAVIOR.
 nItems <- 40
 a      <- rep(1.702,nItems); b <- seq(-5,5,length=nItems)
 c      <- rep(0,nItems); d <- rep(1,nItems)
 nSubjects <- 1; rep <- 100
 theta     <- seq(-1,-1,length=nSubjects)
 S         <- runif(n=nSubjects,min=0.0,max=0.0)
 C         <- runif(n=nSubjects,min=0.3,max=0.3)
 D         <- runif(n=nSubjects,min=0.0,max=0.0)
 set.seed(seed = 100)
 X         <- ggrm4pl(n=nItems, rep=rep,
                      theta=theta, S=S, C=C, D=D,
                      s=1/a, b=b,c=c,d=d)

## Results for each subjects for each models
 essai <- m4plModelShow(X, b=b, s=1/a, c=c, d=d, m=0, prior="uniform")

## Mean results for some speficic models
 median(essai[which(essai$MODEL == "TSCD") ,]$SeT, na.rm=TRUE)
 mean(  essai[which(essai$MODEL == "TSCD") ,]$SeT, na.rm=TRUE)
 mean(  essai[which(essai$MODEL ==   "TD") ,]$SeT, na.rm=TRUE)
 sd(    essai[which(essai$MODEL ==   "TD") ,]$T, na.rm=TRUE)

## Result for each models for the first subject
 essai[which(essai$ID == 1) ,]
 max(essai[which(essai$ID == 1) ,]$LL)

## Difference between the estimated values with the T and TSCD models for the
## first subject
 essai[which(essai$ID == 1 & essai$MODEL == "T"),]$T
       - essai[which(essai$ID == 1 & essai$MODEL == "TSCD"),]$T
```

---

m4plSummary — *Summary of the Results of Estimation with the m4pl Models*

Description

Summary of the results of estimation with the m4pl models.

Usage

```
m4plSummary(      X, ...)

m4plMoreSummary(  x, out = "result", thetaInitial = NULL)

m4plNoMoreSummary(x)
```

Arguments

`X`	data.frame or list: if a `list` results from `m4plPersonParameters` function, if a `data.frame`, any all numeric `data.frame`.
`x`	list: result from `m4plPersonParameters` with `more` set to `TRUE`.
`out`	character: if `out="results"`, the output is for each subjects. If `out="report"`, statistics on all results are computed.
`thetaInitial`	numeric: if initial theta valeus are used the error of estimation is also reported.
`...`	generic: to be able to pass parameters from the `m4plMoreSummary` function.

Value

....

`m4plSummary`

....

The result of `m4plSummary` depends of the `out` condition and the class of `X`. If `X` is a `data.frame`, the function `m4plNoMoreSummary` is called and a data.frame with 2 rows is returned: mean and sd rows.

If `out="result"` and `X` is a list, the function `m4plMoreSummary` is called and a data.frame with the mean of the parameters and their theoretical standard errors is returned:

If `out="report"` and `X` is a list, `m4plMoreSummary` is called and the following list taking in account each parameters is returned:

`parameters`	data.frame: with mean, median, sd an N observations for each parameters.
`se`	data.frame: with mean, median, sd an N observations for the theoretical values of the standard error for each parameters.
`logLikelihood`	data.frame: mean, median, sd an N observations loglikelihood, AIC and BIC for the model.
`eCorrelation`	matrix: empirical correlations between the parameters.
`tCorrelation`	matrix: theoretical correlations between the parameters.

....

m4plNoMoreSummary

....

A data.frame with 2 rows is returned: mean and sd rows.

....

m4plMoreSummary

....

All other outputs from the m4plSummary function.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Blais, J.-G., Raiche, G. and Magis, D. (2009). La detection des patrons de reponses problematiques dans le contexte des tests informatises. *In* Blais, J.-G. (Ed.): *Evaluation des apprentissages et technologies de l'information et de la communication : enjeux, applications et modeles de mesure*. Ste-Foy, Quebec: Presses de l'Universite Laval.

Raiche, G., Magis, D. and Beland, S. (2009). *La correction du resultat d'un etudiant en presence de tentatives de fraudes*. Communication presentee a l'Universite du Quebec a Montreal. Retrieved from http://www.camri.uqam.ca/camri/camriBase/

Raiche, G., Magis, D. and Blais, J.-G. (2008). *Multidimensional item response theory models integrating additional inattention, pseudo-guessing, and discrimination person parameters*. Communication at the annual international Psychometric Society meeting, Durham, New Hamshire. Retrieved from http://www.camri.uqam.ca/camri/camriBase/

See Also

m4plPersonParameters

Examples

```
## GENERATION OF VECTORS OF RESPONSE
 # NOTE THE USUAL PARAMETRIZATION OF THE ITEM DISCRIMINATION,
 # THE VALUE OF THE PERSONNAL FLUCTUATION FIXED AT 0,
 # AND THE VALUE OF THE PERSONNAL PSEUDO-GUESSING FIXED AT 0.30.
 # IT COULD BE TYPICAL OF PLAGIARISM BEHAVIOR.
 nSubjects <- 1
 nItems <- 40
 a       <- rep(1.702,nItems); b <- seq(-5,5,length=nItems)
 c       <- rep(0,nItems); d <- rep(1,nItems)
 theta      <- seq(-2,-2,length=nSubjects)
 S          <- runif(n=nSubjects,min=0.0,max=0.0)
```

```
 C         <- runif(n=nSubjects,min=0.3,max=0.3)
 D         <- runif(n=nSubjects,min=0.0,max=0.0)
 rep <- 100
 set.seed(seed = 10)
 X         <- ggrm4pl(n=nItems, rep=rep,
                      theta=theta, S=S, C=C, D=D,
                      s=1/a, b=b,c=c,d=d)

## Estimation of the model integrating the T and the C parameters
 model <- "C"
 test  <- m4plPersonParameters(x=X, b=b, s=1/a, c=c, d=d, m=0, model=model,
                               prior="uniform", more=TRUE)

## Summary of the preceding model (report and first 5 subjects)
 essai <- m4plSummary(X=test, out="report")
 # Rounding the result of the list to two decimals
 lapply(essai, round, 2)
 essai <- m4plSummary(X=test, out="result")[1:5,]
 lapply(essai, round, 2)
 essai <- m4plSummary(X=test, out="report", thetaInitial=theta)
 lapply(essai, round, 2)
 essai <- m4plSummary(X=test, out="result", thetaInitial=theta)[1:5,]
 lapply(essai, round, 2)

## Results directly from m4plMoreSummary()
 essai <- m4plMoreSummary(x=test, out="report")
 lapply(essai, round, 2)
 essai <- m4plMoreSummary(x=test, out="result")[1:5,]
 round(essai, 2)

## To obtain more general statistics on the result report
 essai <- m4plMoreSummary(x=test, out="result")
 m4plNoMoreSummary(essai)
 summary(m4plMoreSummary(x=test, out="result"))
```

`modelChoose` *Functions For Choosing The m4pl Best Model(s)*

Description

Functions to help to choose the best models from the m4pl person response models family. The function `meanModels` gives the mean of selected and available statistics from a `m4plModelShow` result. The function `modelChoose` shows which model(s) are choosen for each subject, while the function `modelChooseAdd` adds a supplementary logical variable indicating which model(s) are choosen for each subject to the data.frame returned previously by a `m4plModelShow` call .

Usage

```
meanModels(    modelShow, statistics=c("T","S"))
```

```
modelChoose(   modelShow, criteria = "BIC", tol = 0.2)

modelChooseAdd(modelShow, criteria="LL")
```

Arguments

`modelShow`	data.frame: result returned by a `m4plModelShow` call.
`criteria`	character: criteria used to choose between models (LL, AIC or BIC).
`tol`	numeric: tolerance around the choose criteria, so that more models can be considered.
`statistics`	character: a vector of variables for which means of statistics from `m4plModelShow` will be computed.

Details

A tolerance varying between 0.10 and 0.20 is suggested. Lower the value, less models are choosen.

Value

`meanModels`	data.frame: this function return a data.frame of means for each choosen variables.
`modelChoose`	list: return a list of the models(s) choosen for each subject.
`modelChooseAdd`	data.frame: return a data.frame adding a logical indicator of the choosen model(s).

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Blais, J.-G., Raiche, G. and Magis, D. (2009). La detection des patrons de reponses problematiques dans le contexte des tests informatises. *In* Blais, J.-G. (Ed.): *Evaluation des apprentissages et technologies de l'information et de la communication : enjeux, applications et modeles de mesure*. Ste-Foy, Quebec: Presses de l'Universite Laval.

Raiche, G., Magis, D. and Beland, S. (2009). *La correction du resultat d'un etudiant en presence de tentatives de fraudes*. Communication presentee a l'Universite du Quebec a Montreal. Retrieved from http://www.camri.uqam.ca/camri/camriBase/

Raiche, G., Magis, D. and Blais, J.-G. (2008). *Multidimensional item response theory models integrating additional inattention, pseudo-guessing, and discrimination person parameters*. Communication at the annual international Psychometric Society meeting, Durham, New Hamshire. Retrieved from http://www.camri.uqam.ca/camri/camriBase/

See Also

m4plModelShow

Examples

```
## GENERATION OF VECTORS OF RESPONSE
 # NOTE THE USUAL PARAMETRIZATION OF THE ITEM DISCRIMINATION,
 # THE VALUE OF THE PERSONNAL FLUCTUATION FIXED AT 0,
 # AND THE VALUE OF THE PERSONNAL PSEUDO-GUESSING FIXED AT 0.30.
 # IT COULD BE TYPICAL OF PLAGIARISM BEHAVIOR.
 nItems <- 40
 a      <- rep(1.702,nItems); b <- seq(-5,5,length=nItems)
 c      <- rep(0,nItems); d <- rep(1,nItems)
 nSubjects <- 1; rep <- 100
 theta     <- seq(-1,-1,length=nSubjects)
 S         <- runif(n=nSubjects,min=0.0,max=0.0)
 C         <- runif(n=nSubjects,min=0.3,max=0.3)
 D         <- runif(n=nSubjects,min=0,max=0)
 set.seed(seed = 100)
 X         <- ggrm4pl(n=nItems, rep=rep,
                     theta=theta, S=S, C=C, D=D,
                     s=1/a, b=b,c=c,d=d)

## Results for each subjects for each models
 essai <- m4plModelShow(X, b=b, s=1/a, c=c, d=d, m=0, prior="uniform")

## Means of choosen variables from m4plModleShow previous call
 round(meanModels(essai, statistic=c("LL","BIC","T","C")), 2)

## Which model(s) are the best for each subject (LL criteria)
 res <- modelChoose(essai, criteria = "LL", tol = 0.2); res

## Which model(s) are the best for each of the first 20 subjects (BIC criteria)
 res <- modelChoose(essai[which(essai$ID < 20) ,], criteria="BIC"); res

## A look at the 15th subject parameters estimation for each models
 essai[which(essai$ID == 15) ,]

## Add the logical critLL variable to the data frame
 criteria <- "LL"
 res1    <- modelChooseAdd(essai, criteria=criteria)
 # Create a charcater string composed from "crit" and criteria
 crit    <- paste("crit",criteria,sep="")
 # To Show only the lines where the models is choosen according to critLL
 res2    <- res1[which(res1[crit] == TRUE),]; mean(res2$T);sd(res2$T,na.rm=TRUE);res2
 # Give the mean for choosen variables from m4plModelShow
 meanModels(essai, statistic=c("LL","BIC","T","SeT","S","C","D"))
 # Tabulate only the choosen models
 table(res2$MODEL)
 # Show the information for the subject for which the model TSCD was choosen
 res2[which(res2$MODEL == "TSCD") ,]
 # Show only the results for the 5th subject, the with the TSCD model choosen
```

```
 res1[which(res1$ID == 5) ,]
 # Same, but without critLL
 essai[which(essai$ID == 5) ,]

## Simulation whith cheating
 # High proficiency students responding at random to 20
 # easiest ones)
 XHProficiency             <- X
 pourcHasard               <- 0.20; nHasard <- abs(dim(X)[2]*pourcHasard)
 XHProficiency[,1:nHasard] <- rbinom(dim(X)[1]*nHasard, 1, 0.5)
 XHProficiency             <- m4plModelShow(XHProficiency, b=b, s=1/a, c=c, d=d,
                                  m=0, prior="uniform")
 XHProficiency             <- modelChooseAdd(XHProficiency, criteria=criteria)
 XHProficiency             <- XHProficiency[which(XHProficiency$critLL == TRUE),]
 mean(XHProficiency$T);sd(XHProficiency$T,na.rm=TRUE);XHProficiency[1:10,]
 meanModels(XHProficiency, statistic=c("LL","BIC","T","SeT","S","C","D"))
 table(XHProficiency$MODEL)

 # Low proficiency students responding at random to 20
 # (more difficult ones)
 XLProficiency                                <- X
 pourcHasard                                  <- 0.20
 nHasard                                      <- abs(dim(X)[2]*pourcHasard)
 XLProficiency[,(nItems-nHasard+1):nItems] <- rbinom(dim(X)[1]*nHasard, 1, 0.5)
 XLProficiency                                <- m4plModelShow(XLProficiency, b=b,
                                                   s=1/a, c=c, d=d, m=0, prior="uniform")
 XLProficiency                                <- modelChooseAdd(XLProficiency,
                                                                criteria=criteria)
 XLProficiency <- XLProficiency[which(XLProficiency$critLL == TRUE),]
 mean(XLProficiency$T);sd(XLProficiency$T,na.rm=TRUE);XLProficiency[1:10,]
 meanModels(XLProficiency, statistic=c("LL","BIC","T","SeT","S","C","D"))
 table(XLProficiency$MODEL)

 # High proficiency students giving incorrect responses to 20
 # (easiest ones)
 XHProficiency             <- X
 pourcCheat                <- 0.20; nCheat  <- abs(dim(X)[2]*pourcCheat)
 XHProficiency[,1:nCheat]  <- rep(0, dim(X)[1]*dim(X)[2]*pourcCheat)
 XHProficiency             <- m4plModelShow(XHProficiency, b=b, s=1/a, c=c, d=d,
                                  m=0, prior="uniform")
 XHProficiency             <- modelChooseAdd(XHProficiency, criteria=criteria)
 XHProficiency             <- XHProficiency[which(XHProficiency$critLL == TRUE),]
 mean(XHProficiency$T);sd(XHProficiency$T,na.rm=TRUE);XHProficiency[1:10,]
 meanModels(XHProficiency, statistic=c("LL","BIC","T","SeT","S","C","D"))
 table(XHProficiency$MODEL)

 # Low proficiency students giving correct responses to 20
 # (more difficult ones)
 XLProficiency                                <- X
 pourcCheat                                   <- 0.20
 nCheat                                       <- abs(dim(X)[2]*pourcCheat);
 XLProficiency[,(nItems-nCheat+1):nItems]    <- rep(1, dim(X)[1]*dim(X)[2]*pourcCheat)
 XLProficiency                                <- m4plModelShow(XLProficiency, b=b,
```

```
                                        s=1/a, c=c, d=d, m=0, prior="uniform")
XLProficiency                          <- modelChooseAdd(XLProficiency,
                                                      criteria=criteria)
XLProficiency <- XLProficiency[which(XLProficiency$critLL == TRUE),]
mean(XLProficiency$T);sd(XLProficiency$T,na.rm=TRUE);XLProficiency[1:10,]
meanModels(XLProficiency, statistic=c("LL","BIC","T","SeT","S","C","D"))
table(XLProficiency$MODEL)
```

`modelShowClassFunctions` *Functions To Manipulate modelShow Class Objects*

Description

A set of functions to manipulate modelShow class objects. `is.modelShow` is used to verify if the object correspond to a `modelShow` class object. `Round.modelShow`is used to control the number of decimals of the report. Finally, `summary.modelShow` produces different summary reports according th the `report` value.

Usage

```
## S3 method for class 'modelShow':
is(x)

Round(x, digits)

## S3 method for class 'modelShow':
Round(x, digits = 6)

## S3 method for class 'modelShow':
summary(object, ..., report="means",
                              statistics=c("LL","BIC","T"),
                              criteria="LL", digits=6, tol=0.20, color="grey")
```

Arguments

`x`	modelShow: modelShow object.
`object`	modelShow: modelShow object.
`digits`	numeric: number of report decimal digits.
`report`	character: summary type to be reported (means, choose, table or add).
`statistics`	character: a vector of variables for which means of statistics from `m4plModelShow` will be computed.
`criteria`	character: criteria used to choose between models (LL, AIC or BIC).

`tol`	numeric: tolerance around the choose criteria, so that more models can be considered.
`color`	character: color of the bars if a histogram is choosen for the report.
`...`	generic: to be able to pass parameters from the generic `summary` function.

Value

Generic functions for the modelShow class:

- `is.modelShow` logical: is the object of the class modelShow?
- `Round.modelShow` data.frame: return the modelShow object results rounded to `digits` decimals.
- `report="means"` data.frame: means of the `statistics` choosen for each models.
- `report="choosen"` list: which model is choosen for each subject.
- `report="table"` table: table of frequencies of each model chhosen.
- `report="histogram"` histogram: histogram of frequencies of each model choosen.
- `report="add"` data.frame: the choosen model is added to the essai modelShow object.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

See Also

m4plModelShow

Examples

```
## GENERATION OF VECTORS OF RESPONSE
 # NOTE THE USUAL PARAMETRIZATION OF THE ITEM DISCRIMINATION,
 # THE VALUE OF THE PERSONNAL FLUCTUATION FIXED AT 0,
 # AND THE VALUE OF THE PERSONNAL PSEUDO-GUESSING FIXED AT 0.30.
 # IT COULD BE TYPICAL OF PLAGIARISM BEHAVIOR.
 nItems <- 40
 a      <- rep(1.702,nItems); b <- seq(-5,5,length=nItems)
 c      <- rep(0,nItems); d <- rep(1,nItems)
 nSubjects <- 1; rep <- 100
 theta     <- seq(-2,-2,length=nSubjects)
 S         <- runif(n=nSubjects,min=0.0,max=0.0)
 C         <- runif(n=nSubjects,min=0.3,max=0.3)
 D         <- runif(n=nSubjects,min=0,max=0)
 set.seed(seed = 100)
```

```
 X         <- ggrm4pl(n=nItems, rep=rep,
                       theta=theta, S=S, C=C, D=D,
                       s=1/a, b=b,c=c,d=d)

## Results for each subjects for each models
 essai <- m4plModelShow(X, b=b, s=1/a, c=c, d=d, m=0, prior="uniform")

## Is essai of class modelShow?
 is.modelShow(essai)

## Rounding to 2 decimals the first 5 results of essai
 Round(essai[1:5,], 2)

## Means for each models rounded to 3 decimals
 summary(essai, report="means", statistics=c("LL","AIC","BIC","T","SeT"), digits=3)

## Model choosen for each of the first 5 subjects
## and the frequency of these choices with the BIC criteria
 summary(essai[which(essai$ID == (1:5)),], report="choose", criteria="BIC")

## Frequency of the models choosen for all the subjects
## with the LL, AIC and BIC criteria
## Generally, BIC chooses the less models AIC the more.
 summary(essai, report="table", criteria="LL")
 summary(essai, report="table", criteria="AIC")
 summary(essai, report="table", criteria="BIC")

## Frequency of the models choosen for all the subjects
## with the BIC criteria, but with a histogram
 summary(essai, report="histogram", criteria="BIC", color="blue")

## The choosen model is added to the essai modelShow object for all the subjects
## with the LL, AIC and BIC criteria and statistics about theta are computed
## Recall thet rhe generating theta was fixed at -2.00
## The LL criteria seems the best one her according to bias and standard error
 resultLL  <- summary(essai, report="add", criteria="LL")
 resultAIC <- summary(essai, report="add", criteria="AIC")
 resultBIC <- summary(essai, report="add", criteria="BIC")
 # LL
 summary(resultLL[which(resultLL$critLL == TRUE),]$T)
 sd(resultLL[which(resultLL$critLL == TRUE),]$T, na.rm=TRUE)
 # AIC
 summary(resultAIC[which(resultAIC$critAIC == TRUE),]$T)
 sd(resultAIC[which(resultAIC$critAIC == TRUE),]$T, na.rm=TRUE)
 # BIC
 summary(resultBIC[which(resultBIC$critBIC == TRUE),]$T)
 sd(resultBIC[which(resultBIC$critBIC == TRUE),]$T, na.rm=TRUE)
```

PersonParametersEstimate
Estimation of the Personal Parameters from the mpl4 Logistic Model

Description

Estimation of the Personal Parameters from the mpl4 Logistic Model.

Usage

```
m4plEstimate(        x , s = 1/1.702, b = 0, c = 0, d = 1, m = 0,
                     model = "T", prior = "uniform")

m4plEstimateMore(    x, s = 1/1.702, b = 0, c = 0, d = 1, m = 0,
                     model = "T", prior = "uniform")

m4plPersonParameters(x, s = 1/1.702, b = 0, c = 0, d = 1, m = 0,
                     model = "T", prior = "uniform", more = FALSE)
```

Arguments

`x`	integer; vector of item responses for only one subject. Cannot be a matrix for the moment.
`s`	numeric; vector of item fluctuation parameter or the inverse of item discrimination (s= 1/a).
`b`	numeric; vector of item discrimination parameter.
`c`	numeric; vector of item pseudo-guessing parameter.
`d`	numeric; vector of item inattention parameter.
`m`	a priori distribution mean.
`model`	character; different combinations of personnal parameters can be estimated (`"T"`, `"S"`, `"C"`, `"D"`, `"SC"`, `"SD"`, `"CD"` `or` `"SCD"`).
`prior`	character; a priori distribution can be "uniform" (`U(m-4,m+4)`) or "normal" (`N(m,1)`).
`more`	logical: if `TRUE` use `m4plEstimateMore`, if `FALSE` use `m4plEstimate`.

Details

The multidimensional 4 persons parameters logistic distribution (cdf) is equal to:

$$P(x_{ij} = 1|\theta_j, \sigma_j, \chi_j, \delta_j, s_i, b_i, c_i, d_i) = (\chi_j + c_i) + \frac{(d_i - \delta_j) - (\chi_j + c_i)}{1 + e^{\frac{-Da_i(\theta_j - b_i)}{\sqrt{\sigma_j^2 + s_i^2}}}}$$

where the parameters are defined in the section arguments and *i* and *j* are respectively the items and the persons indices. The σ_j, χ_j and δ_j parameters are respectively the personnal fluctuation, pseudo-guessing and inattention parameters. The multidimensional 4 persons parameters logistic model (M4PL) was described by Raiche, Magis and Blais (2008; Raiche, Magis and Beland, 2009; Raiche, Blais and Magis, 2009).

Value

Function `m4plEstimate`

`m4plEstimate`
: numeric; return a vector of personal parameters estimates.

`res` numeric; return a vector of personal parameters estimates.

`se` numeric; return a vector of standard errors.

`corr` numeric; return a correlation matrix between the estimated parameters.

`llikelihood` numeric; return a vector of -log likelihood.

Function `m4plPersonParameters` Values returned are from `m4plEstimate` or `m4plEstimateMore` depending on the `more` condition.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Blais, J.-G., Raiche, G. and Magis, D. (2009). La detection des patrons de reponses problematiques dans le contexte des tests informatises. *In* Blais, J.-G. (Ed.): *Evaluation des apprentissages et technologies de l'information et de la communication : enjeux, applications et modeles de mesure*. Ste-Foy, Quebec: Presses de l'Universite Laval.

Raiche, G., Magis, D. and Beland, S. (2009). *La correction du resultat d'un etudiant en presence de tentatives de fraudes*. Communication presentee a l'Universite du Quebec a Montreal. Retrieved from http://www.camri.uqam.ca/camri/camriBase/

Raiche, G., Magis, D. and Blais, J.-G. (2008). *Multidimensional item response theory models integrating additional inattention, pseudo-guessing, and discrimination person parameters*. Communication at the annual international Psychometric Society meeting, Durham, New Hamshire. Retrieved from http://www.camri.uqam.ca/camri/camriBase/

See Also

`grm4pl`, `ggrm4pl`, `pggrm4pl`

Examples

```
## GENERATION OF VECTORS OF RESPONSE
 # NOTE THE USUAL PARAMETRIZATION OF THE ITEM DISCRIMINATION,
 # THE VALUE OF THE PERSONNAL FLUCTUATION FIXED AT 0,
 # AND THE VALUE OF THE PERSONNAL PSEUDO-GUESSING FIXED AT 0.30.
 # IT COULD BE TYPICAL OF PLAGIARISM BEHAVIOR.
 nItems <- 40
 a      <- rep(1.702,nItems); b <- seq(-5,5,length=nItems)
 c      <- rep(0,nItems); d <- rep(1,nItems)
 nSubjects <- 1; rep <- 2
```

```
 theta     <- seq(-1,-1,length=nSubjects)
 S         <- runif(n=nSubjects,min=0.0,max=0.0)
 C         <- runif(n=nSubjects,min=0.3,max=0.3)
 D         <- runif(n=nSubjects,min=0,max=0)
 set.seed(seed = 100)
 X         <- ggrm4pl(n=nItems, rep=rep,
                      theta=theta, S=S, C=C, D=D,
                      s=1/a, b=b,c=c,d=d)

## Genreric function m4plPersonParameters to use
## prefered to the more specific one: m4plEstimate and m4plEstimateMore
 # .....................................................................
 model <- "C"
 test1 <- m4plPersonParameters(x=X, b=b, s=1/a, c=c, d=d, m=0, model=model,
                               prior="uniform", more=FALSE)
 test2 <- m4plPersonParameters(x=X, b=b, s=1/a, c=c, d=d, m=0, model=model,
                               prior="uniform", more=TRUE)
 # .....................................................................

## ESTIMATION OF THE PERSONNAL PARAMETERS BY ALL MODELS.
 # THE CHOOSEN PRIOR IS UNIFORM WITH m=0.
 # .....................................................................
 m4plEstimate(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "T",    prior="uniform")
 m4plEstimate(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "S",    prior="uniform")
 m4plEstimate(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "C",    prior="uniform")
 m4plEstimate(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "D",    prior="uniform")
 m4plEstimate(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "SC",   prior="uniform")
 m4plEstimate(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "SD",   prior="uniform")
 m4plEstimate(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "CD",   prior="uniform")
 m4plEstimate(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "SCD",  prior="uniform")
 # .....................................................................

## THE SAME ESTIMATION, BUT WITH INFORMATION ABOUT
 # THE STANDARD ERROR, THE CORRELATION AND THE LOG LIKELIKOOD
 m4plEstimateMore(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "T",    prior="uniform")
 m4plEstimateMore(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "S",    prior="uniform")
 m4plEstimateMore(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "C",    prior="uniform")
 m4plEstimateMore(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "D",    prior="uniform")
 m4plEstimateMore(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "SC",   prior="uniform")
 m4plEstimateMore(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "SD",   prior="uniform")
 m4plEstimateMore(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "CD",   prior="uniform")
 m4plEstimateMore(x=X, b=b, s=1/a, c=c, d=d, m=0, model= "SCD",  prior="uniform")
 # .....................................................................

## Same simulation, but with replications
 # .....................................................................
 rep <- 100
 set.seed(seed = 100)
 X         <- ggrm4pl(n=nItems, rep=rep,
                      theta=theta, S=S, C=C, D=D,
                      s=1/a, b=b,c=c,d=d)

## Function used to extract each parameters of the list return by m4plEstimateMore
```

```
extract <- function(x,i) x[[i]]

## Usual IRT model integrating only the parameters theta
apply(         X, 1, m4plEstimate,     b=b, s=1/a, c=c, d=d, m=0, model= "T",
               prior="uniform")
tests <- apply(X, 1, m4plEstimateMore, b=b, s=1/a, c=c, d=d, m=0, model= "T",
               prior="uniform")
personParameters <- matrix(unlist(lapply(tests,extract,1)),nrow=rep)
personSe          <- matrix(unlist(lapply(tests,extract,2)),nrow=rep)
personCor         <- matrix(unlist(lapply(tests,extract,3)),nrow=rep)
personLL          <- matrix(unlist(lapply(tests,extract,4)),nrow=rep)
results           <- data.frame(Parameter=personParameters, Se=personSe,
                                LL=personLL)
round(c(mean=mean(results, na.rm=TRUE)),2); round(c(se=sd(results, na.rm=TRUE)),2)

## Model integrating the parameters theta and C
# Same response patterns for comparison of estimates
t(       apply(X, 1, m4plEstimate,     b=b, s=1/a, c=c, d=d, m=0, model= "C",
               prior="uniform"))
tests <- apply(X, 1, m4plEstimateMore, b=b, s=1/a, c=c, d=d, m=0, model= "C",
               prior="uniform")
personParameters <- t(matrix(unlist(lapply(tests,extract,1)),ncol=rep))
personSe          <- t(matrix(unlist(lapply(tests,extract,2)),ncol=rep))
personLL          <- t(matrix(unlist(lapply(tests,extract,4)),ncol=rep))
results           <- data.frame(Parameter=personParameters, Se=personSe,
                                LL=personLL)
round(c(mean=mean(results, na.rm=TRUE)),2); round(c(se=sd(results, na.rm=TRUE)),2)
## Average correlation between the person parameters
nParameters <- length(personParameters[1,])
personCor   <- unlist(lapply(tests,extract,3))
personCor   <- matrix( mean(data.frame(t(matrix(personCor,ncol=rep))), na.rm=TRUE),
               ncol=nParameters)
personCor

## Model integrating the parameters theta and C
# Same response patterns for comparison of estimates
t(apply(       X,1, m4plEstimate,      b=b, s=1/a, c=c, d=d, m=0, model= "SCD",
               prior="uniform"))
tests <- apply(X, 1, m4plEstimateMore, b=b, s=1/a, c=c, d=d, m=0, model= "SCD",
               prior="uniform")
personParameters <- t(matrix(unlist(lapply(tests,extract,1)),ncol=rep))
personSe          <- t(matrix(unlist(lapply(tests,extract,2)),ncol=rep))
personLL          <- t(matrix(unlist(lapply(tests,extract,4)),ncol=rep))
results           <- data.frame(Parameter=personParameters, Se=personSe,
                                LL=personLL)
round(c(mean=mean(results, na.rm=TRUE)),2); round(c(se=sd(results, na.rm=TRUE)),2)
## Average correlation between the person parameters
nParameters <- length(personParameters[1,])
personCor   <- unlist(lapply(tests,extract,3))
personCor   <- matrix( mean(data.frame(t(matrix(personCor,ncol=rep))), na.rm=TRUE),
               ncol=nParameters)
personCor
# ....................................................................
```

responses4pl *Simulation of Response Patterns and Computation of the Probability of the Patterns*

Description

Simulation of response patterns and computation of the probability of the patterns according to the one, two, three and four parameters logistic item response models.~

Usage

```
gr4pl(N = 10, theta = 0, a = 1, b = 0, c = 0, d = 1)

ggr4pl(n = 5, rep = 1, theta = 0, a = rep(1, n), b = rep(0, n),
       c = rep(0, n), d = rep(1, n))

pggr4pl(x = ggr4pl(rep = 1), rep = 1, n = dim(x)[2], N = dim(x)[1],
        theta = rep(0, N), a = rep(1, n), b = rep(0, n), c = rep(0, n),
        d = rep(1, n), log.p=FALSE, TCC = FALSE)
```

Arguments

`theta`	numeric; vector of proficiency levels (z sscores).
`x`	numeric matrix; response patterns (0 or 1).
`rep`	numeric; number of replications of the simulation of the response patterns.
`n`	numeric; number of items.
`N`	numeric; number of response patterns
`a`	numeric; item discrimination parameters.
`b`	numeric; item difficulty parameters.
`c`	numeric; item pseudo-guessing parameters.
`d`	numeric; item inattention parameters.
`log.p`	logical; if TRUE, probabilities p are given as log(p).
`TCC`	logical; if TRUE generate the TCC figures for each response patterns. Default FALSE.

Details

The function `gr4pl` generates `N` responses to an item according to the theta parameter and the items parameters. The funcfion `ggr4pl` will be used to generate `rep` respose patterns at `n` items. To compute the probability of the response patterns, according to known person and item parameters, the function `pggr4pl` will be applied.

Value

`gr4pl`	numeric; vector of item responses (0 or 1).
`ggr4pl`	numeric; data.frame of responses at n items.
`pggr4pl`	logical; if (TCC ==TRUE) return(list(prob=prob, tcc=tcc)); if (TCC==FALSE) return(prob)

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Hambleton, R. K. and Swaminathan, H. (1985). *Item response theory - Principles and applications*. Boston, Massachuset: Kluwer.

See Also

`grm4pl`, `ggrm4pl`, `pggrm4pl`, `ctt2irt`, `irt2ctt`

Examples

```
## ..................................................................
# Generation of reponses (0,1) from r4pl() for N subjects (default value of N= 10)
 gr4pl(c = 1)
 gr4pl(N = 5, theta = c(-4, 4), c = 0)

# Generation of a 7 responses pattern (0,1) for [rep * length(theta)] subjects
#  The subjects number is equal to [rep * length(theta)]]
#  a,b,c et d are item parameters vectors
 nitems <- 7
 N      <- 10
 a      <- rep(1, nitems)
 b      <- rnorm(nitems)
 c      <- rep(0, nitems)
 d      <- rep(1, nitems)
 theta  <- seq(-4,4,length=5)
 x      <- ggr4pl(n = nitems, rep = N, theta = theta, a = a, b = b, c = c, d = d)
 x

## Probability of a 10 responses pattern and test caracteristic curve (TCC)
 nitems <- 10
 a      <- rep(1,nitems)
 b      <- seq(-4,4,length=nitems)
 c      <- rep(0,nitems)
 d      <- rep(1,nitems)
 N      <- 3
 theta  <- seq(-1,1,length=12)
 # Generation of the response patterns
```

```
 x      <- ggr4pl(n = nitems, rep = N, theta = theta, a = a, b = b, c = c, d = d)
 x
 # Without TCC
 res    <- pggr4pl(x=x, rep=N, theta=theta,a=a,c=c,d=d,TCC=FALSE); res
 # With TCC for each response pattern
 res    <- pggr4pl(x=x, rep=N, theta=theta,a=a,c=c,d=d,TCC=TRUE); res
## ...........................................................
```

`responsesm4pl` *Simulation of Response Patterns and Computation of the Probability of the Patterns from m4pl*

Description

Simulation of response patterns and computation of the probability of the patterns according to the multidimensional one, two, three and four person parameters logistic item response models.

Usage

```
grm4pl(N = 10, theta = 0, S = 0, C = 0, D = 0, s = 1/1.702, b = 0, c = 0, d = 1)

ggrm4pl(n=5,rep=1,theta=0,S=rep(0,length(theta)),C=rep(0,length(theta)),
        D=rep(0,length(theta)),s=rep(1/1.702,n),b=rep(0,n),c=rep(0,n),
        d=rep(1, n))

pggrm4pl(x=ggrm4pl(rep=1),rep=1,n=dim(x)[2],N=dim(x)[1],theta=rep(0,N),
         S=0,C=0,D=0,s=rep(1/1.702,n),b=rep(0,n),c=rep(0,n),d=rep(1,n),
         log.p=FALSE, TCC=FALSE)
```

Arguments

`x`	integer matrix; response patterns (0 or 1).
`rep`	numeric; number of replications of the simulation of the response patterns.
`n`	numeric; number of items.
`N`	numeric; number of response patterns
`theta`	numeric; vector of proficiency levels (z sscores).
`S`	numeric; person fluctuation parameter.
`C`	numeric; person pseud0-guessing parameter.
`D`	numeric; person inattention parameter.
`s`	numeric; item fluctuation parameters.
`b`	numeric; item difficulty parameters.
`c`	numeric; item pseudo-guessing parameters.

`d`	numeric; item inattention parameters.
`log.p`	logical; if TRUE, probabilities p are given as log(p).
`TCC`	logical; if TRUE generate the TCC figures for each response patterns. Default FALSE.

Details

The function `grm4pl` generates `N` responses to an item according to the person parameters and the items parameters. The funcfion `ggrm4pl` will be used to generate `rep` respose patterns at `n` items. To compute the probability of the response patterns, according to known person and item parameters, the function `pggrm4pl` will be applied.

Value

`grm4pl`	integer; vector of item responses (0 or 1).
`ggrm4pl`	integer data.frame; responses for n items.
`pggrm4pl`	graphic; if (TCC ==TRUE) return(list(prob=prob, tcc=tcc)). If (TCC==FALSE) return(prob).

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

References

Ferrando, P. J. (2004). Person reliability in personality measurement: an item response theory analysis. *Applied Psychological Measurement, 28*(2), 126-140.

Hulin, C. L., Drasgow, F., and Parsons, C. K. (1983). *Item response theory*. Homewood, IL: Irwin.

Levine, M. V., and Drasgow, F. (1983). Appropriateness measurement: validating studies and variable ability models. *In* D. J. Weiss (Ed.): *New horizons in testing*. New York, NJ: Academic Press.

Magis, D. (2007). Enhanced estimation methods in IRT. *In* D. Magis (Ed.): *Influence, information and item response theory in discrete data analysis*. Doctoral dissertation, Liege, Belgium: University de Liege.

Trabin, T. E., and Weiss, D. J. (1983). The person response curve: fit of individuals to item response theory models. *In* D. J. Weiss (Ed.): *New horizons in testing*. New York, NJ: Academic Press.

See Also

`gr4pl`, `ggr4pl`, `pggr4pl`, `ctt2irt`, `irt2ctt`

Examples

```
## .................................................................
# Generation of response patterns (0,1) from r4pl() for N subjects (default value
# of N = 10)

# Generation of a response (0,1) from rm4pl for N subjects
 grm4pl(theta=0)
 grm4pl(N=5, theta=c(-4,4), c=0)

# Generation of n m4pl response patterns (0,1) for [rep * length(theta)] subjects
#  The subject number ia equal to [rep * length(theta)]
#  a,b,c et d are item parameters vectors
 nitems <- n <- 7; N <- 1
 s     <- rep(0,nitems); b <- seq(-4,4,length=nitems); c <- rep(0,nitems)
 d     <- rep(1,nitems)
 theta <- seq(-4,4,length=5)
 x     <- ggrm4pl(n=nitems, rep=N, theta=theta,s=s,b=b,c=c,d=d)
 x

# TO BE REWORKED - Probability of a response pattern and test caracteristic curve
# (TCC)
 nItems <- n <- 7; N <- 1
 s      <- rep(0,nItems); b <- seq(-4,4,length=nItems)
 c      <- rep(0,nItems); d <- rep(1,nItems)
 theta <- seq(-4,4,length=5);     S <- rep(1/1.702,length(theta));
 C     <- rep(0.3,length(theta)); D <- rep(0,length(theta))
 x <- ggrm4pl(n=nItems, rep=N, theta=theta, S=S, C=C, D=D, s=s, b=b, c=c, d=d)
 x
 res <- pggrm4pl(x=x, rep=N, theta=theta, S=1/1.702, C=0.3, D=0, s=s, c=c, d=d,
                 TCC=TRUE)
 res
 res <- pggrm4pl(x=x, rep=N, theta=rep(2,length(theta)), S=1/1.702, C=0, D=0,
                 s=s, c=c, d=d, TCC=FALSE)
 res
 pggrm4pl(theta=3)
 pggrm4pl(n=10, theta=seq(-4,4,length=5), x=ggrm4pl(rep=1), TCC=TRUE)
## .................................................................
```

`utilities` *Utility Functions*

Description

Some functions to help in the generation of binary data or to interpret m4pl models results.

`rmultinomial` is used to draw `n` values from a `x` vector of values according to a multinomial probability distribution. `rmultinomial` is different from the `stats::rmultinom` function in that it return only the value of the selected draw and not a binary vector corresponding tho the position of the draw propCorrect computes the expected proportion of correct responses to a test according to the item and person parameters of a m4pl models.

Usage

```
rmultinomial(x, n = 100, prob = rep(1, length(x))/length(x))

propCorrect(theta, S, C, D, s, b, c, d)
```

Arguments

x	numeric: vector of values to draw from.
n	numeric: number of draws.
prob	numeric: probability assigned to each value of the x vector. Default to equiprobability.
theta	numeric; vector of person proficiency (θ) levels scaled on a normal z score.
S	numeric: positive vector of personal fluctuation parameters (σ).
C	numeric: positive vector of personal pseudo-guessing parameters (χ, a probability between 0 and 1).
D	numeric: positive vector of personal inattention parameters (δ, a probability between 0 and 1).
s	numeric: positive vector of item fluctuation parameters.
b	numeric: vector of item difficulty parameters.
c	numeric: positive vector of item pseudo-guessing parameters (a probability between 0 and 1).
d	numeric: positive vector of item inattention parameters (a probability between 0 and 1).

Value

propCorrect

numeric:	return n draws from a multinimial distribution.
numeric:	return the expected proportion of correcte responses to a test.

Author(s)

Gilles Raiche, Universite du Quebec a Montreal (UQAM),

Departement d'education et pedagogie

⟨Raiche.Gilles@uqam.ca⟩, http://www.er.uqam.ca/nobel/r17165/

See Also

Multinom

Examples

```
## Comparison of the results from the multinomial and multinom functions
 x <- c(1,4,9)
 # Values draws
 rmultinomial(x=x, n=10)
 # Binary vectors draws
 rmultinom(n=10, size = 1, prob=rep(1,length(x))/length(x))

## Computation of the expected proportion of correct responses varying values
 # of theta (-3 to 3) and of pseudo-guessing (C = 0.0 to 0.6) person parameters
 nItems  <- 40
 a       <- rep(1.702,nItems); b <- seq(-3,3,length=nItems)
 c       <- rep(0,nItems); d <- rep(1,nItems)
 theta   <- seq(-3.0, 3.0, by=1.0)
 C       <- seq( 0.0, 1.0, by=0.1)
 D       <- S <- 0

 results <- matrix(NA, ncol=length(C), nrow=length(theta))
 colnames(results) <- C; rownames(results) <- theta
 for (i in (1:length(theta))) {
  results[i,] <- propCorrect(theta = theta[i], S = 0, C = C, D = 0,
                             s = 1/a, b = b, c = c, d = d)
  }
  round(results, 2)

## Computation of the expected proportion of correct responses varying values
 # of theta (-3 to 3) and of pseudo-guessing (C = 0.0 to 0.6) person parameters
 # if we choose the correct modelisation itegrating the C pseudo-guessing papameter
 # and if we choose according to a model selection by LL criteria
 nItems <- 40
 a      <- rep(1.702,nItems); b <- seq(-3,3,length=nItems)
 c      <- rep(0,nItems); d <- rep(1,nItems)
 nSubjects <- 300
 theta     <- rmultinomial(c(-1), nSubjects)
 S         <- rmultinomial(c(0), nSubjects)
 C         <- rmultinomial(seq(0,0.9,by=0.1), nSubjects)
 D         <- rmultinomial(c(0), nSubjects)
 set.seed(seed = 100)
 X         <- ggrm4pl(n=nItems, rep=1,
                   theta=theta, S=S, C=C, D=D,
                   s=1/a, b=b,c=c,d=d)
 # Results for each subjects for each models
 essai     <- m4plModelShow(X, b=b, s=1/a, c=c, d=d, m=0, prior="uniform")
 total     <- rowSums(X)
 pourcent  <- total/nItems * 100
 pCorrect  <- numeric(dim(essai)[1])
 for ( i in (1:dim(essai)[1]))
  pCorrect[i] <- propCorrect(essai$T[i],0,0,0,s=1/a,b=b,c=c,d=d)
 resultLL  <- summary(essai, report="add", criteria="LL")
 resultLL  <- data.frame(resultLL, theta=theta, TS=S, TC=C, errorT=resultLL$T - theta,
                        total=total, pourcent=pourcent, tpcorrect=pCorrect)
 # If the only theta model is badly choosen
```

```
results <- resultLL[which(resultLL$MODEL == "T" ),]
byStats <- "TC"; ofStats <- "tpcorrect"
MeansByThetaT <- cbind(
 aggregate(results[ofStats], by=list(Theta=factor(results[,byStats]) ),
          mean, na.rm=TRUE),
 aggregate(results[ofStats], by=list(Theta=factor(results[,byStats])), sd, na.rm=TRUE),
 aggregate(results["SeT"], by=list(Theta=factor(results[,byStats])), mean, na.rm=TRUE),
 aggregate(results[ofStats], by=list(theta=factor(results[,byStats])), length)
 )[,-c(3,5,7)]
 names(MeansByThetaT) <- c("C", "pCorrect", "seE", "SeT", "n")
 MeansByThetaT[,-c(1,4,5)] <- round(MeansByThetaT[,-c(1,4,5)], 2)
 MeansByThetaT[,-c(4,5)]
# Only for the TC model
results <- resultLL[which(resultLL$MODEL == "TC" ),]
byStats <- "TC"; ofStats <- "tpcorrect"
MeansByThetaC <- cbind(
 aggregate(results[ofStats], by=list(Theta=factor(results[,byStats]) ),
          mean, na.rm=TRUE),
 aggregate(results[ofStats], by=list(Theta=factor(results[,byStats])), sd, na.rm=TRUE),
 aggregate(results["SeT"], by=list(Theta=factor(results[,byStats])), mean, na.rm=TRUE),
 aggregate(results[ofStats], by=list(theta=factor(results[,byStats])), length)
 )[,-c(3,5,7)]
 names(MeansByThetaC) <- c("C", "pCorrect", "seE", "SeT", "n")
 MeansByThetaC[,-c(1,4,5)] <- round(MeansByThetaC[,-c(1,4,5)], 2)
 MeansByThetaC[,-c(4,5)]
# For the model choosen according to the LL criteria
results <- resultLL[which(resultLL$critLL == TRUE),]
byStats <- "TC"; ofStats <- "tpcorrect"
MeansByThetaLL <- cbind(
 aggregate(results[ofStats], by=list(Theta=factor(results[,byStats]) ),
          mean, na.rm=TRUE),
 aggregate(results[ofStats], by=list(Theta=factor(results[,byStats])), sd, na.rm=TRUE),
 aggregate(results["SeT"], by=list(Theta=factor(results[,byStats])), mean, na.rm=TRUE),
 aggregate(results[ofStats], by=list(theta=factor(results[,byStats])), length)
 )[,-c(3,5,7)]
 names(MeansByThetaLL) <- c("C", "pCorrect", "seE", "SeT", "n")
 MeansByThetaLL[,-c(1,4,5)] <- round(MeansByThetaLL[,-c(1,4,5)], 2)
 MeansByThetaLL[,-c(4,5)]
 # Grapical comparison of the estimation of the
 # by means of the 3 preceeding models
 plot(MeansByThetaT$pCorrect   ~ levels(MeansByThetaT$C),  type="l", lty=1,
      xlab="Pseudo-Guessing", ylab="% of Correct Responses")
 lines(MeansByThetaC$pCorrect  ~ levels(MeansByThetaC$C),  type="l", lty=2)
 lines(MeansByThetaLL$pCorrect ~ levels(MeansByThetaLL$C), type="l", lty=3)
 text(x=0.60, y=0.80, "Without correction", cex=0.8)
 text(x=0.50, y=0.38, "Without Knowledge of the Correct Model", cex=0.8)
 text(x=0.65, y=0.50, "With Knowledge of the Correct Model", cex=0.8)
```

Index