Package ‘rpf’

April 29, 2021

Type Package
Title Response Probability Functions
Version 1.0.7
Date 2021-04-29
Maintainer Joshua Pritikin <jpritikin@pobox.com>

Description The purpose of this package is to factor out logic
and math common to Item Factor Analysis fitting, diagnostics, and
analysis. It is envisioned as core support code suitable for more
specialized IRT packages to build upon. Complete access to optimized C
functions are made available with R_RegisterCCallable().
This software is described in Pritikin & Falk (2020) <doi:10.1177/0146621620929431>.

License GPL (>= 3)

URL https://github.com/jpritikin/rpf

Depends methods,
      parallel,
      R (>= 2.14.0)
Imports Rcpp (>= 1.0.2),
       mvtnorm,
       lifecycle
Suggests testthat,
       roxygen2,
       ggplot2,
       reshape2,
       gridExtra,
       numDeriv,
       knitr,
       mirt,
       markdown

LinkingTo Rcpp,
        RcppEigen

VignetteBuilder knitr

RdMacros lifecycle
Encoding UTF-8
LazyData yes
LazyDataCompression xz
RoxygenNote 7.1.0
SystemRequirements GNU make
Collate 'init.R'
   'classes.R'
   'fit.R'
   'drm.R'
   'nrm.R'
   'mcm.R'
   'grm.R'
   'LSAT.R'
   'sample.R'
   'dataframe.R'
   'diagnose.R'
   'science.R'
   'kct.R'
   'openmx.R'
   'flexmirt.R'
   'util.R'
   'imp.R'
   'grmp.R'
   'gpcmp.R'
   'RcppExports.R'

R topics documented:
An introduction ......................................................... 4
as.IFAgroup .......................................................... 5
bestToOmit ............................................................. 6
ChenThissen1997 ....................................................... 7
Class rpf.1dim ......................................................... 9
Class rpf.1dim.drm .................................................... 10
Class rpf.1dim.gpcmp ................................................. 10
Class rpf.1dim.graded ................................................. 10
Class rpf.1dim.grm ................................................... 10
Class rpf.1dim.grmp .................................................. 10
Class rpf.1dim.lmp .................................................... 11
Class rpf.base ......................................................... 11
Class rpf.mdim ........................................................ 11
Class rpf.mdim.drm ................................................... 11
Class rpf.mdim.graded ............................................... 11
Class rpf.mdim.grm ................................................... 12
Class rpf.mdim.mcm ................................................... 12
Class rpf.mdim.nrm ................................................... 12
R topics documented:

collapseCategoricalCells ........................................ 12
compressDataFrame ............................................. 13
crosstabTest .................................................... 14
EAPscores ......................................................... 14
expandDataFrame .................................................. 16
fromFactorLoading ............................................... 16
fromFactorThreshold ........................................... 17
itemOutcomeBySumScore ......................................... 17
kct ................................................................. 19
logit ............................................................... 19
LSAT6 ............................................................... 20
LSAT7 ............................................................... 20
multinomialFit ..................................................... 21
observedSumScore ................................................ 23
omitItems ........................................................... 24
omitMostMissing .................................................. 25
orderCompletely .................................................. 27
ordinal.gamma ...................................................... 27
ptw2011.gof.test ................................................ 28
read.flexmirt ....................................................... 29
rpf.1dim.fit ....................................................... 30
rpf.1dim.moment .................................................. 32
rpf.1dim.residual ................................................ 33
rpf.1dim.stdresidual ............................................ 34
rpf.dLL ............................................................. 34
rpf.drm ............................................................. 35
rpf.dTheta .......................................................... 36
rpf.gpcmp .......................................................... 37
rpf.grm ............................................................. 38
rpf.grmp ........................................................... 39
rpf.id_of ............................................................ 41
rpf.info ............................................................. 41
rpf.lmp ............................................................. 42
rpf.logprob ........................................................ 44
rpf.mcm ............................................................ 44
rpf.mean.info ..................................................... 45
rpf.mean.info1 .................................................... 46
rpf.modify ........................................................ 46
rpf.nrm ............................................................. 47
rpf.numParam ...................................................... 48
rpf.numSpec ....................................................... 48
rpf.ogive .......................................................... 49
rpf.paramInfo ...................................................... 50
rpf.prob .......................................................... 50
rpf.rescale ........................................................ 51
rpf.rparam ........................................................ 52
rpf.sample ........................................................ 52
science ............................................................. 54
Description

Factor out logic and math common to Item Factor Analysis fitting, diagnostics, and analysis. It is envisioned as core support code suitable for more specialized IFA packages to build upon.

Details

This package provides optimized, low-level functions to map parameters to response probabilities for dichotomous (1PL, 2PL and 3PL) `rpf.drm` and polytomous (graded response `rpf.grm`, partial credit/generalized partial credit (via the nominal model), and nominal `rpf.nrm` items.

Item model parameters are passed around as a numeric vector. A 1D matrix is also acceptable. Regardless of model, parameters are always ordered as follows: discrimination/slope ("a"), difficulty/intercept ("b"), and pseudo guessing/upper-bound ("g"/"u"). If person ability ranges from negative to positive then probabilities are output from incorrect to correct. That is, a low ability person (e.g., ability = -2) will be more likely to get an item incorrect than correct. For example, a dichotomous model that returns \[.25, .75\] indicates a probability of .25 for incorrect and .75 for correct. A polytomous model will have the most incorrect probability at index 1 and the most correct probability at the maximum index.

All models are always in the logistic metric. To obtain normal ogive discrimination parameters, divide slope parameters by \(rpf.\text{ogive}\). Item models are estimated in slope-intercept form. Input/output matrices arranged in the way most convenient for low-level processing in C. The maximum absolute logit is 35 because \(f(x) := 1-\exp(x)\) loses accuracy around \(f(-35)\) and equals 1 at \(f(-38)\) due to the limited accuracy of double precision floating point.

This package could also accrete functions to support plotting (but not the actual plot functions).

References


See Also

See \texttt{rpf.rparam} to create item parameters.

\begin{verbatim}
\textbf{as.IFAgroup} \hspace{1cm} \textit{Convert an OpenMx MxModel object into an IFA group}
\end{verbatim}

\section*{Description}

When "\texttt{minItemsPerScore}" is passed, EAP scores will be computed from the data and stored. Scores are required for some diagnostic tests. See discussion of "\texttt{minItemsPerScore}" in \texttt{EAPscores}.

\section*{Usage}

\begin{verbatim}
as.IFAgroup(
  mxModel,
  data = NULL,
  container = NULL,
  \ldots,
  minItemsPerScore = NULL
)
\end{verbatim}

\section*{Arguments}

- \texttt{mxModel} \hspace{1cm} MxModel object
- \texttt{data} \hspace{1cm} observed data (otherwise the data will be taken from the \texttt{mxModel})
- \texttt{container} \hspace{1cm} an MxModel in which to search for the latent distribution matrices
- \ldots \hspace{1cm} Not used. Forces remaining arguments to be specified by name.
- \texttt{minItemsPerScore} \hspace{1cm} minimum number of items required to compute a score (also see description)

\section*{Value}

a groups with item parameters and latent distribution

\section*{Format of a group}

A model, or group within a model, is represented as a named list.

- \texttt{spec} \hspace{1cm} list of response model objects
- \texttt{param} \hspace{1cm} numeric matrix of item parameters
- \texttt{free} \hspace{1cm} logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- \texttt{mean} \hspace{1cm} numeric vector giving the mean of the latent distribution
- \texttt{cov} \hspace{1cm} numeric matrix giving the covariance of the latent distribution
- \texttt{data} \hspace{1cm} data.frame containing observed item responses, and optionally, weights and frequencies
The `param` matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in `spec` and order of columns in `param` are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by `param` colnames.

Currently only a multivariate normal distribution is available, parameterized by the `mean` and `cov`. If `mean` and `cov` are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, `qwidth=2` and `qpoints=5` would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by `EAPscores`.

See Also

`ifaTools`

---

### bestToOmit

**Identify the columns with most missing data**

**Description**

If a reference column is given then only rows that are not missing on the reference column are considered. Otherwise all rows are considered.

**Usage**

```r
bestToOmit(grp, omit, ref = NULL)
```

**Arguments**

- `grp` a list containing the model and data. See the details section.
- `omit` the maximum number of items to omit
- `ref` the reference column (optional)
Format of a group

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
- **cov**: numeric matrix giving the covariance of the latent distribution
- **data**: data.frame containing observed item responses, and optionally, weights and frequencies
- **score**: factors scores with response patterns in rows
- **weightColumn**: name of the data column containing the numeric row weights (optional)
- **freqColumn**: name of the data column containing the integral row frequencies (optional)
- **qwidth**: width of the quadrature expressed in Z units
- **qpoints**: number of quadrature points
- **minItemsPerScore**: minimum number of non-missing items when estimating factor scores

The **param** matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in **spec** and order of columns in **param** are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by **param** colnames.

Currently only a multivariate normal distribution is available, parameterized by the **mean** and **cov**. If **mean** and **cov** are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, **qwidth=2** and **qpoints=5** would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by **EAPscores**.

See Also

Other scoring: **EAPscores()**, **itemOutcomeBySumScore()**, **observedSumScore()**, **omitItems()**, **omitMostMissing()**, **sumScoreEAP()**

---

**Description**

Item Factor Analysis makes two assumptions: (1) that the latent distribution is reasonably approximated by the multivariate Normal and (2) that items are conditionally independent. This test examines the second assumption. The presence of locally dependent items can inflate the precision of estimates causing a test to seem more accurate than it really is.
Usage

ChenThissen1997(
  grp,
  ..., data = NULL,
  inames = NULL,
  qwidth = 6,
  qpoints = 49,
  method = "pearson",
  .twotier = TRUE,
  .parallel = TRUE
)

Arguments

  grp a list containing the model and data. See the details section.
  ... Not used. Forces remaining arguments to be specified by name.
  data [Deprecated] data
  inames a subset of items to examine
  qwidth [Deprecated]
  qpoints [Deprecated]
  method method to use to calculate P values. The default is the Pearson X^2 statistic. Use "lr" for the similar likelihood ratio statistic.
  .twotier whether to enable the two-tier optimization
  .parallel whether to take advantage of multiple CPUs (default TRUE)

Details

Statically significant entries suggest that the item pair has local dependence. Since log(.01)=-4.6, an absolute magnitude of 5 is a reasonable cut-off. Positive entries indicate that the two item residuals are more correlated than expected. These items may share an unaccounted for latent dimension. Consider a redesign of the items or the use of testlets for scoring. Negative entries indicate that the two item residuals are less correlated than expected.

Value

a list with raw, pval and detail. The pval matrix is a lower triangular matrix of log P values with the sign determined by relative association between the observed and expected tables (see ordinal.gamma)

Format of a group

A model, or group within a model, is represented as a named list.

  spec list of response model objects
  param numeric matrix of item parameters
Class rpf.1dim

free  logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
mean  numeric vector giving the mean of the latent distribution
cov  numeric matrix giving the covariance of the latent distribution
data  data.frame containing observed item responses, and optionally, weights and frequencies
score factors scores with response patterns in rows
weightColumn  name of the data column containing the numeric row weights (optional)
freqColumn  name of the data column containing the integral row frequencies (optional)
qwidth  width of the quadrature expressed in Z units
qpoints  number of quadrature points
minItemsPerScore  minimum number of non-missing items when estimating factor scores

The param matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in spec and order of columns in param are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by param colnames.

Currently only a multivariate normal distribution is available, parameterized by the mean and cov. If mean and cov are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, qwidth=2 and qpoints=5 would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by EAPscores.

References

See Also
ifaTools
Other diagnostic: SitemFit1(), SitemFit(), multinomialFit(), rpf.1dim.fit(), sumScoreEAPTest()

Class rpf.1dim  The base class for 1 dimensional response probability functions.

Description
The base class for 1 dimensional response probability functions.
Class `rpf.1dim.drm`  
*Unidimensional dichotomous item models (1PL, 2PL, and 3PL).*

**Description**

Unidimensional dichotomous item models (1PL, 2PL, and 3PL).

---

Class `rpf.1dim.gpcmp`  
*Unidimensional generalized partial credit monotonic polynomial.*

**Description**

Unidimensional generalized partial credit monotonic polynomial.

---

Class `rpf.1dim.graded`  
*The base class for 1 dimensional graded response probability functions.*

**Description**

This class contains methods common to both the generalized partial credit model and the graded response model.

---

Class `rpf.1dim.grm`  
*The unidimensional graded response item model.*

**Description**

The unidimensional graded response item model.

---

Class `rpf.1dim.grmp`  
*Unidimensional graded response monotonic polynomial.*

**Description**

Unidimensional graded response monotonic polynomial.
Class \texttt{rpf.1dim.lmp} \hspace{5cm} \textit{Unidimensional logistic function of a monotonic polynomial.}

\subsection*{Description}
Unidimensional logistic function of a monotonic polynomial.

---

Class \texttt{rpf.base} \hspace{5cm} \textit{The base class for response probability functions.}

\subsection*{Description}
Item specifications should not be modified after creation.

---

Class \texttt{rpf.mdim} \hspace{5cm} \textit{The base class for multi-dimensional response probability functions.}

\subsection*{Description}
The base class for multi-dimensional response probability functions.

---

Class \texttt{rpf.mdim.drm} \hspace{5cm} \textit{Multidimensional dichotomous item models (M1PL, M2PL, and M3PL).}

\subsection*{Description}
Multidimensional dichotomous item models (M1PL, M2PL, and M3PL).

---

Class \texttt{rpf.mdimgraded} \hspace{5cm} \textit{The base class for multi-dimensional graded response probability functions.}

\subsection*{Description}
This class contains methods common to both the generalized partial credit model and the graded response model.
### Class rpf.mdim.grm

*The multidimensional graded response item model.*

#### Description

The multidimensional graded response item model.

### Class rpf.mdim.mcm

*The multiple-choice response item model (both unidimensional and multidimensional models have the same parameterization).*

#### Description

The multiple-choice response item model (both unidimensional and multidimensional models have the same parameterization).

### Class rpf.mdim.nrm

*The nominal response item model (both unidimensional and multidimensional models have the same parameterization).*

#### Description

The nominal response item model (both unidimensional and multidimensional models have the same parameterization).

---

### collapseCategoricalCells

*Collapse small sample size categorical frequency counts*

#### Description

Collapse small sample size categorical frequency counts

#### Usage

```r
collapseCategoricalCells(observed, expected, minExpected = 1)
```
compressDataFrame

Arguments

- observed: the observed frequency table
- expected: the expected frequency table
- minExpected: the minimum expected cell frequency

Pearson’s $X^2$ test requires some minimum frequency per cell to avoid an inflated false positive rate. This function will merge cells with the lowest frequency counts until all the counts are above the minimum threshold. Cells that have been merged are filled with NAs. The resulting tables and number of merged cells is returned.

Examples

```
O = matrix(c(7,31,42,20,0), 1,5)
E = matrix(c(3,39,50,8,0), 1,5)
collapseCategoricalCells(O,E,9)
```

compressDataFrame

Compress a data frame into unique rows and frequencies

Description

Compress a data frame into unique rows and frequency counts.

Usage

```
compressDataFrame(tabdata, freqColName = "freq", .asNumeric = FALSE)
```

Arguments

- tabdata: An object of class data.frame
- freqColName: Column name to contain the frequencies
- .asNumeric: logical. Whether to cast the frequencies to the numeric type

Value

Returns a compressed data frame

Examples

```
df <- as.data.frame(matrix(c(sample.int(2, 30, replace=TRUE)), 10, 3))
compressDataFrame(df)
```
**crosstabTest**  
*Monte-Carlo test for cross-tabulation tables*

**Description**

[Experimental] This is for developers.

**Usage**

\[
\text{crosstabTest}(\text{ob, ex, trials})
\]

**Arguments**

- **ob**: observed table
- **ex**: expected table
- **trials**: number of Monte-Carlo trials

**EAPscores**  
*Compute Expected A Posteriori (EAP) scores*

**Description**

If you have missing data then you must specify `minItemsPerScore`. This option will set scores to NA when there are too few items to make an accurate score estimate. If you are using the scores as point estimates without considering the standard error then you should set `minItemsPerScore` as high as you can tolerate. This will increase the amount of missing data but scores will be more accurate. If you are carefully considering the standard errors of the scores then you can set `minItemsPerScore` to 1. This will mimic the behavior of most other IFA software wherein scores are estimated if there is at least 1 non-NA item for the score. However, it may make more sense to set `minItemsPerScore` to 0. When set to 0, all NA rows are scored to the prior distribution.

**Usage**

\[
\text{EAPscores}(<\text{grp}, \ldots, \text{compressed} = \text{FALSE})
\]

**Arguments**

- **grp**: a list containing the model and data. See the details section.
- **...**: Not used. Forces remaining arguments to be specified by name.
- **compressed**: output one score per observed data row even when freqColumn is set (default FALSE)

**Details**

Output is not affected by the presence of a weightColumn.
Format of a group

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
- **cov**: numeric matrix giving the covariance of the latent distribution
- **data**: data.frame containing observed item responses, and optionally, weights and frequencies
- **score**: factors scores with response patterns in rows
- **weightColumn**: name of the data column containing the numeric row weights (optional)
- **freqColumn**: name of the data column containing the integral row frequencies (optional)
- **qwidth**: width of the quadrature expressed in Z units
- **qpoints**: number of quadrature points
- **minItemsPerScore**: minimum number of non-missing items when estimating factor scores

The **param** matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in **spec** and order of columns in **param** are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by **param** colnames.

Currently only a multivariate normal distribution is available, parameterized by the **mean** and **cov**. If **mean** and **cov** are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, **qwidth=2** and **qpoints=5** would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by **EAPscores**.

See Also

Other scoring: **bestToOmit()**, **itemOutcomeBySumScore()**, **observedSumScore()**, **omitItems()**, **omitMostMissing()**, **sumScoreEAP()**

Examples

```R
spec <- list()
spec[1:3] <- list(rpf.grm(outcomes=3))
param <- sapply(spec, rpf.rparam)
data <- rpf.sample(5, spec, param)
colnames(param) <- colnames(data)
grp <- list(spec=spec, param=param, data=data, minItemsPerScore=1L)
EAPscores(grp)
```
expandDataFrame

Expand summary table of patterns and frequencies

Description

Expand a summary table of unique response patterns to a full sized data-set.

Usage

```r
expandDataFrame(tabdata, freqName = NULL)
```

Arguments

- `tabdata`: An object of class `data.frame` with the unique response patterns and the number of frequencies.
- `freqName`: Column name containing the frequencies.

Value

Returns a data frame with all the response patterns.

Author(s)

Based on code by Phil Chalmers <rphilip.chalmers@gmail.com>

Examples

```r
data(LSAT7)
expandDataFrame(LSAT7, freqName="freq")
```

fromFactorLoading

Convert factor loadings to response function slopes

Description

Convert factor loadings to response function slopes.

Usage

```r
fromFactorLoading(loading, ogive = rpf.ogive)
```

Arguments

- `loading`: a matrix with items in the rows and factors in the columns.
- `ogive`: the ogive constant (default `rpf.ogive`).

fromFactorThreshold

Value

a slope matrix with items in the columns and factors in the rows

See Also

Other factor model equivalence: fromFactorThreshold(), toFactorLoading(), toFactorThreshold()

--------------------------------------------------------

fromFactorThreshold  Convert factor thresholds to response function intercepts

Description

Convert factor thresholds to response function intercepts

Usage

fromFactorThreshold(threshold, loading, ogive = rpf.ogive)

Arguments

threshold a matrix with items in the columns and thresholds in the rows
loading a matrix with items in the rows and factors in the columns
ogive the ogive constant (default rpf.ogive)

Value

an item intercept matrix with items in the columns and intercepts in the rows

See Also

Other factor model equivalence: fromFactorLoading(), toFactorLoading(), toFactorThreshold()

--------------------------------------------------------

itemOutcomeBySumScore  Produce an item outcome by observed sum-score table

Description

Produce an item outcome by observed sum-score table

Usage

itemOutcomeBySumScore(grp, mask, interest)
Arguments

grp  a list containing the model and data. See the details section.
mask  a vector of logicals indicating which items to include
interest  index or name of the item of interest

Format of a group

A model, or group within a model, is represented as a named list.

spec  list of response model objects
param  numeric matrix of item parameters
free  logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
mean  numeric vector giving the mean of the latent distribution
cov  numeric matrix giving the covariance of the latent distribution
data  data.frame containing observed item responses, and optionally, weights and frequencies
score  factors scores with response patterns in rows
weightColumn  name of the data column containing the numeric row weights (optional)
freqColumn  name of the data column containing the integral row frequencies (optional)
qwidth  width of the quadrature expressed in Z units
qpoints  number of quadrature points
minItemsPerScore  minimum number of non-missing items when estimating factor scores

The param matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in spec and order of columns in param are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by param colnames.

Currently only a multivariate normal distribution is available, parameterized by the mean and cov. If mean and cov are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, qwidth=2 and qpoints=5 would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by EAPscores.

See Also

Other scoring: EAPscores(), bestToOmit(), observedSumScore(), omitItems(), omitMostMissing(), sumScoreEAP()

Examples

set.seed(1)
spec <- list()
spec[1:3] <- rpf.grm(outcomes=3)
param <- sapply(spec, rpf.rparam)
data <- rpf.sample(5, spec, param)
colnames(param) <- colnames(data)
grp <- list(spec=spec, param=param, data=data)
itemOutcomeBySumScore(grp, c(FALSE,TRUE,TRUE), 1L)

---

**kct**

*Knox Cube Test dataset*

**Description**

These data from Wright & Stone (1979, p. 31) were fit with Winsteps 3.73 using a 1PL model (slope fixed to 1).

**References**


**Examples**

data(kct)

---

**logit**

*Transform from [0,1] to the reals*

**Description**

The logit function is a standard transformation from [0,1] (such as a probability) to the real number line. This function is exactly the same as qlogis.

**Usage**

logit(p, location = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)

**Arguments**

- **p**: a number between 0 and 1
- **location**: see qlogis
- **scale**: see qlogis
- **lower.tail**: see qlogis
- **log.p**: see qlogis

**See Also**

qlogis, plogis
Examples

\[
\begin{align*}
\logit(0.5) & \approx 0 \\
\logit(0.25) & \approx -1.098 \\
\logit(0) & \approx -\infty
\end{align*}
\]

---

**LSAT6**

*Description of LSAT6 data*

**Description**

Data from Thissen (1982); contains 5 dichotomously scored items obtained from the Law School Admissions Test, section 6.

**Author(s)**

Phil Chalmers <rphilip.chalmers@gmail.com>

**References**


**Examples**

```r
data(LSAT6)
```

---

**LSAT7**

*Description of LSAT7 data*

**Description**

Data from Bock & Lieberman (1970); contains 5 dichotomously scored items obtained from the Law School Admissions Test, section 7.

**Author(s)**

Phil Chalmers <rphilip.chalmers@gmail.com>

**References**


**Examples**

```r
data(LSAT7)
```
multinomialFit

**Multinomial fit test**

**Description**

For degrees of freedom, we use the number of observed statistics (incorrect) instead of the number of possible response patterns (correct) (see Bock, Gibbons, & Muraki, 1998, p. 265). This is not a huge problem because this test becomes poorly calibrated when the multinomial table is sparse. For more accurate p-values, you can conduct a Monte-Carlo simulation study (see examples).

**Usage**

```r
multinomialFit(
  grp,
  independenceGrp,
  ...,  
  method = "lr",
  log = TRUE,
  .twotier = TRUE
)
```

**Arguments**

- `grp` a list containing the model and data. See the details section.
- `independenceGrp` the independence group
- `...` Not used. Forces remaining arguments to be specified by name.
- `method` `lr` (default) or `pearson`
- `log` whether to report p-value in log units
- `.twotier` whether to use the two-tier optimization (default TRUE)

**Details**

Rows with missing data are ignored.

The full information test is described in Bartholomew & Tzamourani (1999, Section 3). For CFI and TLI, you must provide an `independenceGrp`.

**Format of a group**

A model, or group within a model, is represented as a named list.

- `spec` list of response model objects
- `param` numeric matrix of item parameters
- `free` logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- `mean` numeric vector giving the mean of the latent distribution
multinomialFit

cov numeric matrix giving the covariance of the latent distribution
data data.frame containing observed item responses, and optionally, weights and frequencies
score factors scores with response patterns in rows
weightColumn name of the data column containing the numeric row weights (optional)
freqColumn name of the data column containing the integral row frequencies (optional)
qwidth width of the quadrature expressed in Z units
qpoints number of quadrature points
minItemsPerScore minimum number of non-missing items when estimating factor scores

The param matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in spec and order of columns in param are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by param colnames.

Currently only a multivariate normal distribution is available, parameterized by the mean and cov. If mean and cov are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, qwidth=2 and qpoints=5 would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by EAPscores.

References


See Also

Other diagnostic: ChenThissen1997(), SitemFit1(), SitemFit(), rpf.1dim.fit(), sumScoreEAPTest()

Examples

# Create an example IFA group
grp <- list(spec=list())
grp$spec[1:10] <- rpf.grm()
grp$param <- sapply(grp$spec, rpf.rparam)
colnames(grp$param) <- paste("i", 1:10, sep="")
grp$mean <- 0
grp$cov <- diag(1)
grp$uniqueFree <- sum(grp$param != 0)
grp$data <- rpf.sample(1000, grp=grp)

# Monte-Carlo simulation study
mcReps <- 3 # increase this to 10,000 or so
stat <- rep(NA, mcReps)
for (rx in 1:mcReps) {
observedSumScore

```r
observedSumScore(grp, ..., mask, summary = TRUE)
```

**Arguments**

- **grp**: a list containing the model and data. See the details section.
- **...**: Not used. Forces remaining arguments to be specified by name.
- **mask**: a vector of logicals indicating which items to include
- **summary**: whether to return a summary (default) or per-row scores

**Format of a group**

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
- **cov**: numeric matrix giving the covariance of the latent distribution
- **data**: data.frame containing observed item responses, and optionally, weights and frequencies
- **score**: factors scores with response patterns in rows
- **weightColumn**: name of the data column containing the numeric row weights (optional)
- **freqColumn**: name of the data column containing the integral row frequencies (optional)
- **qwidth**: width of the quadrature expressed in Z units
- **qpoints**: number of quadrature points
- **minItemsPerScore**: minimum number of non-missing items when estimating factor scores

**Description**

When `summary=TRUE`, tabulation uses row frequency multiplied by row weight.

**Usage**

```r
observedSumScore(grp, ..., mask, summary = TRUE)
```
The `param` matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in `spec` and order of columns in `param` are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by `param` colnames.

Currently only a multivariate normal distribution is available, parameterized by the mean and cov. If mean and cov are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, `qwidth=2` and `qpoints=5` would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by `EAPscores`.

See Also

Other scoring: `EAPscores`, `bestToOmit`, `itemOutcomeBySumScore`, `omitItems`, `omitMostMissing`, `sumScoreEAP`.

Examples

```r
spec <- list()
spec[1:3] <- rpf.grm(outcomes=3)
param <- sapply(spec, rpf.rparam)
data <- rpf.sample(5, spec, param)
colnames(param) <- colnames(data)
grp <- list(spec=spec, param=param, data=data)
observedSumScore(grp)
```

### omitItems

**Omit the given items**

#### Description

Omit the given items

#### Usage

```r
omitItems(grp, excol)
```

#### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>grp</code></td>
<td>a list containing the model and data. See the details section.</td>
</tr>
<tr>
<td><code>excol</code></td>
<td>vector of column names to omit</td>
</tr>
</tbody>
</table>
Format of a group

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
- **cov**: numeric matrix giving the covariance of the latent distribution
- **data**: data.frame containing observed item responses, and optionally, weights and frequencies
- **score**: factors scores with response patterns in rows
- **weightColumn**: name of the data column containing the numeric row weights (optional)
- **freqColumn**: name of the data column containing the integral row frequencies (optional)
- **qwidth**: width of the quadrature expressed in Z units
- **qpoints**: number of quadrature points
- **minItemsPerScore**: minimum number of non-missing items when estimating factor scores

The **param** matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in **spec** and order of columns in **param** are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by **param** colnames.

Currently only a multivariate normal distribution is available, parameterized by the **mean** and **cov**. If **mean** and **cov** are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, **qwidth=2** and **qpoints=5** would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by **EAPscores**.

See Also

Other scoring: **EAPscores()**, **bestToOmit()**, **itemOutcomeBySumScore()**, **observedSumScore()**, **omitMostMissing()**, **sumScoreEAP()**

---

**omitMostMissing**

*Omit items with the most missing data*

Description

Items with no missing data are never omitted, regardless of the number of items requested.

Usage

```r
omitMostMissing(grp, omit)
```
Arguments

- **grp**: a list containing the model and data. See the details section.
- **omit**: the maximum number of items to omit

Format of a group

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
- **cov**: numeric matrix giving the covariance of the latent distribution
- **data**: data.frame containing observed item responses, and optionally, weights and frequencies
- **score**: factors scores with response patterns in rows
- **weightColumn**: name of the data column containing the numeric row weights (optional)
- **freqColumn**: name of the data column containing the integral row frequencies (optional)
- **qwidth**: width of the quadrature expressed in Z units
- **qpoints**: number of quadrature points
- **minItemsPerScore**: minimum number of non-missing items when estimating factor scores

The **param** matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in **spec** and order of columns in **param** are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by **param** colnames.

Currently only a multivariate normal distribution is available, parameterized by the **mean** and **cov**. If **mean** and **cov** are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, **qwidth=2** and **qpoints=5** would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by **EAPscores**.

See Also

Other scoring: **EAPscores()**, **bestToOmit()**, **itemOutcomeBySumScore()**, **observedSumScore()**, **omitItems()**, **sumScoreEAP()**
**orderCompletely**

Order a data.frame by missingness and all columns

**Description**

Completely order all rows in a data.frame.

**Usage**

```r
orderCompletely(observed)
```

**Arguments**

- `observed` a data.frame holding ordered factors in every column

**Value**

the sorted order of the rows

**Examples**

```r
df <- as.data.frame(matrix(c(sample.int(2, 30, replace=TRUE)), 10, 3))
mask <- matrix(c(sample.int(3, 30, replace=TRUE)), 10, 3) == 1
df[mask] <- NA
df[orderCompletely(df),]
```

**ordinal.gamma**

Compute the ordinal gamma association statistic

**Description**

Compute the ordinal gamma association statistic

**Usage**

```r
ordinal.gamma(mat)
```

**Arguments**

- `mat` a cross tabulation matrix

**References**

ptw2011.gof.test

Examples

# Example data from Agresti (1990, p. 21)
jobsat <- matrix(c(20,22,13,7,24,38,28,18,80,104,81,54,82,125,113,92), nrow=4, ncol=4)
ordional.gamma(jobsat)

ptw2011.gof.test Compute the P value that the observed and expected tables come from the same distribution

Description

[Experimental] This test is an alternative to Pearson’s X^2 goodness-of-fit test. In contrast to Pearson’s X^2, no ad hoc cell collapsing is needed to avoid an inflated false positive rate in situations of sparse cell frequencies. The statistic rapidly converges to the Monte-Carlo estimate as the number of draws increases.

Usage

ptw2011.gof.test( observed, expected)

Arguments

observed observed matrix
expected expected matrix

Value

The P value indicating whether the two tables come from the same distribution. For example, a significant result (P < alpha level) rejects the hypothesis that the two matrices are from the same distribution.

References


Examples

draws <- 17
observed <- matrix(c(.294, .176, .118, .411), nrow=2) * draws
expected <- matrix(c(.235, .235, .176, .353), nrow=2) * draws
ptw2011.gof.test(observed, expected) # not significant
Description

[Experimental] This was last updated in 2017 and may no longer work.

Usage

read.flexmirt(fname)

Arguments

fname  file name

Details

Load the item parameters from a flexMIRT PRM file.

Value

a list of groups as described in the details

Format of a group

A model, or group within a model, is represented as a named list.

spec  list of response model objects
param  numeric matrix of item parameters
free  logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
mean  numeric vector giving the mean of the latent distribution
cov  numeric matrix giving the covariance of the latent distribution
data  data.frame containing observed item responses, and optionally, weights and frequencies
score  factors scores with response patterns in rows
weightColumn  name of the data column containing the numeric row weights (optional)
freqColumn  name of the data column containing the integral row frequencies (optional)
qwidth  width of the quadrature expressed in Z units
qpoints  number of quadrature points
minItemsPerScore  minimum number of non-missing items when estimating factor scores

The param matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in spec and order of columns in param are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by param colnames.
Currently only a multivariate normal distribution is available, parameterized by the mean and cov. If mean and cov are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, qwidth=2 and qpoints=5 would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by EAPscores.

---

**rpf.1dim.fit**  
*Calculate item and person Rasch fit statistics*

**Description**

Note: These statistics are only appropriate if all discrimination parameters are fixed equal and items are conditionally independent (see ChenThissen1997). A best effort is made to cope with missing data.

**Usage**

```r
rpf.1dim.fit(
  spec,
  params,
  responses,
  scores,
  margin,
  group = NULL,
  wh.exact = TRUE
)
```

**Arguments**

- `spec`: list of item response models [Deprecated]
- `params`: matrix of item parameters, 1 per column [Deprecated]
- `responses`: persons in rows and items in columns [Deprecated]
- `scores`: model derived person scores [Deprecated]
- `margin`: for people 1, for items 2
- `group`: spec, params, data, and scores can be provided in a list instead of as arguments
- `wh.exact`: whether to use the exact Wilson-Hilferty transformation

**Details**

Exact distributional properties of these statistics are unknown (Masters & Wright, 1997, p. 112). For details on the calculation, refer to Wright & Masters (1982, p. 100).

The Wilson-Hilferty transformation is biased for less than 25 items. Consider wh.exact=FALSE for less than 25 items.
Format of a group

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
- **cov**: numeric matrix giving the covariance of the latent distribution
- **data**: data.frame containing observed item responses, and optionally, weights and frequencies
- **score**: factors scores with response patterns in rows
- **weightColumn**: name of the data column containing the numeric row weights (optional)
- **freqColumn**: name of the data column containing the integral row frequencies (optional)
- **qwidth**: width of the quadrature expressed in Z units
- **qpoints**: number of quadrature points
- **minItemsPerScore**: minimum number of non-missing items when estimating factor scores

The **param** matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in **spec** and order of columns in **param** are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by **param** colnames.

Currently only a multivariate normal distribution is available, parameterized by the **mean** and **cov**. If **mean** and **cov** are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, **qwidth=2** and **qpoints=5** would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by **EAPscores**.

A model, or group within a model, is represented as a named list.
The `param` matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in `spec` and order of columns in `param` are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting `rownames`. Item names are assigned by `param` `colnames`.

Currently only a multivariate normal distribution is available, parameterized by the `mean` and `cov`. If `mean` and `cov` are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, `qwidth=2` and `qpoints=5` would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by `EAPscores`.

**References**


**See Also**

Other diagnostic: `ChenThissen1997()`, `SitemFit1()`, `SitemFit()`, `multinomialFit()`, `sumScoreEAPTest()`

**Examples**

```r
data(kct)
responses <- kct.people[,paste("V",2:19, sep="")]
rownames(responses) <- kct.people$NAME
colnames(responses) <- kct.items$NAME
scores <- kct.people$MEASURE
params <- cbind(1, kct.items$MEASURE, logit(0), logit(1))
rownames(params) <- kct.items$NAME
items <- list()
items[1:18] <- rpf.drm()
params[,2] <- -params[,2]

rpf.1dim.fit(items, t(params), responses, scores, 2, wh.exact=TRUE)
```

**rpf.1dim.moment**

*Calculate cell central moments*

**Description**

Popular central moments include 2 (variance) and 4 (kurtosis).

**Usage**

`rpf.1dim.moment(spec, params, scores, m)`
**Arguments**

- **spec**: list of item models
- **params**: data frame of item parameters, 1 per row
- **scores**: model derived person scores
- **m**: which moment

**Value**

moment matrix

---

**rpf.1dim.residual** *Calculate residuals*

**Description**

Calculate residuals

**Usage**

```r
rpf.1dim.residual(spec, params, responses, scores)
```

**Arguments**

- **spec**: list of item models
- **params**: data frame of item parameters, 1 per row
- **responses**: persons in rows and items in columns
- **scores**: model derived person scores

**Value**

residuals
Calculate standardized residuals

Usage

```r
rpf.1dim.stdresidual(spec, params, responses, scores)
```

Arguments

- **spec**: list of item models
- **params**: data frame of item parameters, 1 per row
- **responses**: persons in rows and items in columns
- **scores**: model derived person scores

Value

standardized residuals

Item parameter derivatives

Evaluate the partial derivatives of the log likelihood with respect to each parameter at where with weight.

Usage

```r
rpf.dLL(m, param, where, weight)
```

Arguments

- **m**: item model
- **param**: item parameters
- **where**: location in the latent space
- **weight**: per outcome weights (typically derived by observation)
Details

It is not easy to write an example for this function. To evaluate the derivative, you need to sum the
derivatives across a quadrature. You also need response outcome weights at each quadrature point.
It is not anticipated that this function will be often used in R code. It’s mainly to expose a C-level
function for occasional debugging.

Value

first and second order partial derivatives of the log likelihood evaluated at \(w\). For \(p\) parameters,
the first \(p\) values are the first derivative and the next \(p(p+1)/2\) columns are the lower triangle of the
second derivative.

See Also

The numDeriv package.

---

### rpf.drm

Create a dichotomous response model

---

Description

For slope vector \(a\), intercept \(c\), pseudo-guessing parameter \(g\), upper bound \(u\), and latent ability vector
theta, the response probability function is

\[
\begin{align*}
P(\text{pick} = 0|a, c, g, u, \theta) &= 1 - P(\text{pick} = 1|a, c, g, u, \theta) \\
P(\text{pick} = 1|a, c, g, u, \theta) &= g + (u - g) \frac{1}{1 + \exp(-(a \theta + c))}
\end{align*}
\]

Usage

\[
rpf.drm(factors = 1, multidimensional = TRUE, poor = FALSE)
\]

Arguments

- **factors**: the number of factors
- **multidimensional**: whether to use a multidimensional model. Defaults to `TRUE`.
- **poor**: if `TRUE`, use the traditional parameterization of the 1d model instead of the
  slope-intercept parameterization

Details

The pseudo-guessing and upper bound parameter are specified in logit units (see `logit`).
For discussion on the choice of priors see Cai, Yang, and Hansen (2011, p. 246).
Value

an item model

References


See Also

Other response model: `rpf.gpcmp()`, `rpf.grmp()`, `rpf.grm()`, `rpf.lmp()`, `rpf.mcm()`, `rpf.nrm()`

Examples

```r
spec <- rpf.drm()
        rpf.prob(spec, rpf.rparam(spec), 0)
```

---

### rpf.dTheta

*Item derivatives with respect to the location in the latent space*

**Description**

Evaluate the partial derivatives of the response probability with respect to ability. See `rpf.info` for an application.

**Usage**

```r
rpf.dTheta(m, param, where, dir)
```

**Arguments**

- `m` item model
- `param` item parameters
- `where` location in the latent distribution
- `dir` if more than 1 factor, a basis vector
Create monotonic polynomial generalized partial credit (GPC-MP) model

Description
This model is a polytomous model proposed by Falk & Cai (2016) and is based on the generalized partial credit model (Muraki, 1992).

Usage
rpf.gpcmp(outcomes = 2, q = 0, multidimensional = FALSE)

Arguments
- outcomes: The number of possible response categories.
- q: a non-negative integer that controls the order of the polynomial (2q+1) with a default of q=0 (1st order polynomial = generalized partial credit model).
- multidimensional: whether to use a multidimensional model. Defaults to FALSE. The multidimensional version is not yet available.

Details
The GPC-MP replaces the linear predictor part of the generalized partial credit model with a monotonic polynomial, \( m(\theta; \omega, \xi, \alpha, \tau) \). The response function for category \( k \) is:

\[
P(pick = k | \omega, \xi, \alpha, \tau, \theta) = \frac{\exp(\sum_{v=0}^{k} (\xi_k + m(\theta; \omega, \xi, \alpha, \tau)))}{\sum_{u=0}^{K-1} \exp(\sum_{v=0}^{n} (\xi_u + m(\theta; \omega, \xi, \alpha, \tau)))}
\]

where \( \alpha \) and \( \tau \) are vectors of length \( q \). The GPC-MP uses the same parameterization for the polynomial as described for the logistic function of a monotonic polynomial (LMP). See also (rpf.lmp).

The order of the polynomial is always odd and is controlled by the user specified non-negative integer, \( q \). The model contains \( 1+(\text{outcomes}-1)+2*q \) parameters and are used as input to the rpf.prob function in the following order: \( \omega \) - natural log of the slope of the item model when \( q=0 \), \( \xi \) - a (outcomes-1)-length vector of intercept parameters, \( \alpha \) and \( \tau \) - two parameters that control bends in the polynomial. These latter parameters are repeated in the same order for models with \( q>0 \). For example, a \( q=2 \) polynomial with 3 categories will have an item parameter vector of: \( \omega, \xi_1, \xi_2, \alpha_1, \alpha_2, \tau_1, \alpha_2, \tau_2 \).

Note that the GPC-MP reduces to the LMP when the number of categories is 2, and the GPC-MP reduces to the generalized partial credit model when the order of the polynomial is 1 (i.e., \( q=0 \)).

Value
an item model
**rpf.grm**

Create a graded response model

**Description**

For outcomes k in 0 to K, slope vector a, intercept vector c, and latent ability vector theta, the response probability function is

\[
P(\text{pick} = 0|a, c, \theta) = 1 - P(\text{pick} = 1|a, c_1, \theta)\\
P(\text{pick} = k|a, c, \theta) = \frac{1}{1 + \exp(-(a\theta + c_k))} - \frac{1}{1 + \exp(-(a\theta + c_{k+1}))}\\
P(\text{pick} = K|a, c, \theta) = \frac{1}{1 + \exp(-(a\theta + c_K))}
\]

**Usage**

```
> rpf.grm(outcomes = 2, factors = 1, multidimensional = TRUE)
```

**Arguments**

- `outcomes` The number of choices available
- `factors` the number of factors
- `multidimensional` whether to use a multidimensional model. Defaults to TRUE.
Details
The graded response model was designed for an item with a series of dependent parts where a higher score implies that easier parts of the item were surmounted. If there is any chance your polytomous item has independent parts then consider \texttt{rpf.nrm}. If your categories cannot cross then the graded response model provides a little more information than the nominal model. Stronger a priori assumptions offer provide more power at the cost of flexibility.

Value
an item model

See Also
Other response model: \texttt{rpf.drm()}, \texttt{rpf.gpcmp()}, \texttt{rpf.grmp()}, \texttt{rpf.lmp()}, \texttt{rpf.mcm()}, \texttt{rpf.nrm()}

Examples
\begin{verbatim}
spec <- rpf.grm()
rpf.prob(spec, rpf.rparam(spec), 0)
\end{verbatim}

---

\texttt{rpf.grmp} \hspace{1cm} Create monotonic polynomial graded response (GR-MP) model

Description
The GR-MP model replaces the linear predictor of the graded response model (Samejima, 1969, 1972) with a monotonic polynomial (Falk, conditionally accepted).

Usage
\texttt{rpf.grmp(outcomes = 2, q = 0, multidimensional = FALSE)}

Arguments
\begin{itemize}
  \item \texttt{outcomes} The number of possible response categories. When equal to 2, the model reduces to the logistic function of a monotonic polynomial (LMP).
  \item \texttt{q} a non-negative integer that controls the order of the polynomial (2q+1) with a default of q=0 (1st order polynomial = graded response model).
  \item \texttt{multidimensional} whether to use a multidimensional model. Defaults to \texttt{FALSE}. The multidimensional version is not yet available.
\end{itemize}
Details

Given its relationship to the graded response model, the GR-MP is constructed in an analogous way:

\[
P(pick = 0|\lambda, \alpha, \tau, \theta) = 1 - \frac{1}{1 + \exp(-(-\xi_1 + m(\theta; \lambda, \alpha, \tau)))}
\]

\[
P(pick = k|\lambda, \alpha, \tau, \theta) = \frac{1}{1 + \exp(-(\xi_k + m(\theta; \lambda, \alpha, \tau)))} - \frac{1}{1 + \exp(-(-\xi_{k+1} + m(\theta; \lambda, \alpha, \tau)))}
\]

\[
P(pick = K|\lambda, \alpha, \tau, \theta) = \frac{1}{1 + \exp(-(-\xi_K + m(\theta; \lambda, \alpha, \tau)))}
\]

The order of the polynomial is always odd and is controlled by the user specified non-negative integer, q. The model contains 1+(outcomes-1)+2*q parameters and are used as input to the \texttt{rpf.prob} or \texttt{rpf.dTheta} functions in the following order: \lambda - slope of the item model when q=0, \xi - a (outcomes-1)-length vector of intercept parameters, \alpha and \tau - two parameters that control bends in the polynomial. These latter parameters are repeated in the same order for models with q>0. For example, a q=2 polynomial with 3 categories will have an item parameter vector of: \lambda, \xi_1, \xi_2, \alpha_1, \tau_1, \alpha_2, \tau_2.

As with other monotonic polynomial-based item models (e.g., \texttt{rpf.lmp}), the polynomial looks like the following:

\[
m(\theta; \lambda, \alpha, \tau) = b_1 \theta + b_2 \theta^2 + \ldots + b_{2q+1} \theta^{2q+1}
\]

However, the coefficients, b, are not directly estimated, but are a function of the item parameters, and the parameterization of the GR-MP is different than that currently appearing for the logistic function of a monotonic polynomial (LMP; \texttt{rpf.lmp}) and monotonic polynomial generalized partial credit (GPC-MP; \texttt{rpf.gpcmp}) models. In particular, the polynomial is parameterized such that boundary discrimination functions for the GR-MP will be all monotonically increasing or decreasing for any given item. This allows the possibility of items that load either negatively or positively on the latent trait, as is common with reverse-worded items in non-cognitive tests (e.g., personality).

The derivative \(m'(\theta; \lambda, \alpha, \tau)\) is parameterized in the following way:

\[
m'(\theta; \lambda, \alpha, \tau) = \begin{cases} 
\frac{\lambda}{\prod_{u=1}^{q} (1 - 2\alpha_u \theta + (\alpha_u^2 + \exp(\tau_u))\theta^2)} & \text{if } q > 0 \\
\lambda & \text{if } q = 0
\end{cases}
\]

Note that the only difference between the GR-MP and these other models is that \lambda is not re-parameterized and may take on negative values. When \lambda is negative, it is analogous to having a negative loading or a monotonically decreasing function.

Value

an item model

References


See Also

Other response model: `rpf.drm()`, `rpf.gpcmp()`, `rpf.grm()`, `rpf.lmp()`, `rpf.mcm()`, `rpf.nrm()`

Examples

```r
spec <- rpf.grmp(5, 2) # 5-category, 3rd order polynomial
theta <- seq(-3, 3, .1)
p <- rpf.prob(spec, c(2.77, 2, 1, 0, -1, .89, -8.7, -.74, -8.99), theta)
```

---

**rpf.id_of**

*Convert an rpf item model name to an ID*

**Description**

This is an internal function and should not be used.

**Usage**

```r
rpf.id_of(name)
```

**Arguments**

- **name**: name of the item model (string)

**Value**

the integer ID assigned to the given model

---

**rpf.info**

*Map an item model, item parameters, and person trait score into a information vector*

**Description**

Map an item model, item parameters, and person trait score into a information vector

**Usage**

```r
rpf.info(ii, ii.p, where, basis = 1)
```

**Arguments**

- **ii**: an item model
- **ii.p**: item parameters
- **where**: the location in the latent distribution
- **basis**: if more than 1 factor, a positive basis vector
Value

Fisher information

References


Examples

```r
i1 <- rpf.drm()
i1.p <- c(.6,1,.1,.95)
theta <- seq(0,3,.05)
plot(theta, rpf.info(i1, i1.p, t(theta)), type="l")
```

---

**rpf.lmp**  
*Create logistic function of a monotonic polynomial (LMP) model*

**Description**

This model is a dichotomous response model originally proposed by Liang (2007) and is implemented using the parameterization by Falk & Cai (2016).

**Usage**

```r
rpf.lmp(q = 0, multidimensional = FALSE)
```

**Arguments**

- `q`  
a non-negative integer that controls the order of the polynomial (2q+1) with a default of q=0 (1st order polynomial = 2PL).

- `multidimensional`  
whether to use a multidimensional model. Defaults to FALSE. The multidimensional version is not yet available.

**Details**

The LMP model replaces the linear predictor part of the two-parameter logistic function with a monotonic polynomial, \( m(\theta, \omega, \xi, \alpha, \tau) \),

\[
P(pick = 1|\omega, \xi, \alpha, \tau, \theta) = \frac{1}{1 + \exp(-(\xi + m(\theta; \omega, \alpha, \tau)))}
\]

where \( \alpha \) and \( \tau \) are vectors of length q.

The order of the polynomial is always odd and is controlled by the user specified non-negative integer, q. The model contains 2+2*q parameters and are used in conjunction with the `rpf.prob` or `rpf.dTheta` function in the following order: \( \omega \) - the natural log of the slope of the item model when
q=0, ξ - the intercept, α and τ - two parameters that control bends in the polynomial. These latter parameters are repeated in the same order for models with q>0. For example, a q=2 polynomial with have an item parameter vector of: ω, ξ, α₁, τ₁, α₂, τ₂.

In general, the polynomial looks like the following:

\[ m(\theta; \omega, \alpha, \tau) = b_1 \theta + b_2 \theta^2 + \ldots + b_{2q+1} \theta^{2q+1} \]

However, the coefficients, b, are not directly estimated, but are a function of the item parameters. In particular, the derivative \( m'(\theta; \omega, \alpha, \tau) \) is parameterized in the following way:

\[
m'(\theta; \omega, \alpha, \tau) = \begin{cases} 
\exp(\omega) \prod_{u=1}^{q} (1 - 2\alpha_u \theta + (\alpha_u^2 + \exp(\tau_u))\theta^2) & \text{if } q > 0 \\
\exp(\omega) & \text{if } q = 0
\end{cases}
\]

See Falk & Cai (2016) for more details as to how the polynomial is constructed. At the lowest order polynomial (q=0) the model reduces to the two-parameter logistic (2PL) model. However, parameterization of the slope parameter, ω, is currently different than the 2PL (i.e., slope = \exp(ω)). This parameterization ensures that the response function is always monotonically increasing without requiring constrained optimization.

For an alternative parameterization that releases constraints on ω, allowing for monotonically decreasing functions, see \texttt{rpf.grmp}. And for polytomous items, see both \texttt{rpf.grmp} and \texttt{rpf.gpcmp}.

\section*{Value}

an item model

\section*{References}


\section*{See Also}

Other response model: \texttt{rpf.drm()}, \texttt{rpf.gpmp()}, \texttt{rpf.grmp()}, \texttt{rpf.grm()}, \texttt{rpf.mcm()}, \texttt{rpf.nrm()}

\section*{Examples}

```r
spec <- rpf.lmp(1) # 3rd order polynomial
theta<-seq(-3,3,.1)
p<-rpf.prob(spec, c(-.11,.37,.24,-.21),theta)

spec <- rpf.lmp(2) # 5th order polynomial
p<-rpf.prob(spec, c(.69,.71,-.5,-8.48,.52,-3.32),theta)
```
**rpf.logprob**

Map an item model, item parameters, and person trait score into a probability vector

**Description**

Note that in general, exp(rpf.logprob(..)) != rpf.prob(..) because the range of logits is much wider than the range of probabilities due to limitations of floating point numerical precision.

**Usage**

```r
rpf.logprob(m, param, theta)
```

**Arguments**

- `m` an item model
- `param` item parameters
- `theta` the trait score(s)

**Value**

a vector of probabilities. For dichotomous items, probabilities are returned in the order incorrect, correct. Although redundant, both incorrect and correct probabilities are returned in the dichotomous case for API consistency with polytomous item models.

**Examples**

```r
i1 <- rpf.drm()
i1.p <- rpf.rparam(i1)
rpf.logprob(i1, c(i1.p), -1)  # low trait score
rpf.logprob(i1, c(i1.p), c(0,1))  # average and high trait score
```

---

**rpf.mcm**

Create a multiple-choice response model

**Description**

[Experimental]

**Usage**

```r
rpf.mcm(outcomes = 2, numChoices = 5, factors = 1)
```
Arguments

outcomes: the number of possible outcomes
numChoices: the number of choices available
factors: the number of factors

Details

This function instantiates a multiple-choice response model.

Value

an item model

Author(s)

Jonathan Weeks <weeksjp@gmail.com>

See Also

Other response model: rpf.drm(), rpf.gpcmp(), rpf.grmp(), rpf.grm(), rpf.lmp(), rpf.nrm()

rpf.mean.info

Find the point where an item provides mean maximum information

Description

[Experimental] This is a point estimate of the mean difficulty of items that do not offer easily interpretable parameters such as the Generalized PCM. Since the information curve may not be unimodal, this function integrates across the latent space.

Usage

rpf.mean.info(spec, param, grain = 0.1)

Arguments

spec: list of item specs
param: list or matrix of item parameters
grain: the step size for numerical integration (optional)
rpf.mean.info1  

*Find the point where an item provides mean maximum information*

**Description**

[Experimental]

**Usage**

```r
rpf.mean.info1(spec, iparam, grain = 0.1)
```

**Arguments**

- `spec`: an item spec
- `iparam`: an item parameter vector
- `grain`: the step size for numerical integration (optional)

---

rpf.modify  

*Create a similar item specification with the given number of factors*

**Description**

Create a similar item specification with the given number of factors

**Usage**

```r
rpf.modify(m, factors)
```

**Arguments**

- `m`: item model
- `factors`: the number of factors/dimensions

**Examples**

```r
s1 <- rpf.grm(factors=3)
rpf.rparam(s1)
s2 <- rpf.modify(s1, 1)
rpf.rparam(s2)
```
Create a nominal response model

Description

This function instantiates a nominal response model.

Usage

```r
rpf.nrm(outcomes = 3, factors = 1, T.a = "trend", T.c = "trend")
```

Arguments

- **outcomes**: The number of choices available
- **factors**: The number of factors
- **T.a**: The T matrix for slope parameters
- **T.c**: The T matrix for intercept parameters

Details

The transformation matrices $T_a$ and $T_c$ are chosen by the analyst and not estimated. The $T$ matrices must be invertible square matrices of size outcomes-1. As a shortcut, either $T$ matrix can be specified as "trend" for a Fourier basis or as "id" for an identity basis. The response probability function is

$$a = T_a \alpha$$

$$c = T_c \gamma$$

$$P(\text{pick} = k| s, a_k, c_k, \theta) = C \frac{1}{1 + \exp(- (s \theta a_k + c_k))}$$

where $a_k$ and $c_k$ are the result of multiplying two vectors of free parameters $\alpha$ and $\gamma$ by fixed matrices $T_a$ and $T_c$, respectively; $a_0$ and $c_0$ are fixed to 0 for identification; and $C$ is a normalizing factor to ensure that $\sum_k P(\text{pick} = k) = 1$.

Value

an item model

References


See Also

Other response model: `rpf.drm()`, `rpf.gpcmp()`, `rpf.grmp()`, `rpf.grm()`, `rpf.lmp()`, `rpf.mcm()`
Examples

```r
spec <- rpf.nrm()
rpf.prob(spec, rpf.rparam(spec), 0)
# typical parameterization for the Generalized Partial Credit Model
gpcm <- function(outcomes) rpf.nrm(outcomes, T.c=lower.tri(diag(outcomes-1),TRUE) * -1)
spec <- gpcm(4)
rpf.prob(spec, rpf.rparam(spec), 0)
```

---

**rpf.numParam**

*Length of the item parameter vector*

**Description**

Length of the item parameter vector

**Usage**

```r
rpf.numParam(m)
```

**Arguments**

- `m`: item model

**Examples**

```r
rpf.numParam(rpf.grm(outcomes=3))
rpf.numParam(rpf.nrm(outcomes=3))
```

---

**rpf.numSpec**

*Length of the item model vector*

**Description**

Length of the item model vector

**Usage**

```r
rpf.numSpec(m)
```

**Arguments**

- `m`: item model

**Examples**

```r
rpf.numSpec(rpf.grm(outcomes=3))
rpf.numSpec(rpf.nrm(outcomes=3))
```
**Description**

The ogive constant can be multiplied by the discrimination parameter to obtain a response curve very similar to the Normal cumulative distribution function (Haley, 1952; Molenaar, 1974). Recently, Savalei (2006) proposed a new constant of 1.749 based on Kullback-Leibler information.

**Usage**

```
rfp.ogive
```

**Format**

An object of class `numeric` of length 1.

**Details**

In recent years, the logistic has grown in favor, and therefore, this package does not offer any special support for this transformation (Baker & Kim, 2004, pp. 14-18).

**References**


**rpf.paramInfo**

*Retrieve a description of the given parameter*

**Description**

Retrieve a description of the given parameter

**Usage**

```r
rpf.paramInfo(m, num = NULL)
```

**Arguments**

- `m` : item model
- `num` : vector of parameters (defaults to all)

**Value**

a list containing the type, upper bound, and lower bound

**Examples**

```r
rpf.paramInfo(rpf.drm())
```

---

**rpf.prob**

*Map an item model, item parameters, and person trait score into a probability vector*

**Description**

This function is known by many names in the literature. When plotted against latent trait, it is often called a traceline, item characteristic curve, or item response function. Sometimes the word 'category' or 'outcome' is used in place of 'item'. For example, 'item response function' might become 'category response function'. All these terms refer to the same thing.

**Usage**

```r
rpf.prob(m, param, theta)
```

**Arguments**

- `m` : an item model
- `param` : item parameters
- `theta` : the trait score(s)
Value

A vector of probabilities. For dichotomous items, probabilities are returned in the order incorrect, correct. Although redundant, both incorrect and correct probabilities are returned in the dichotomous case for API consistency with polytomous item models.

Examples

```r
i1 <- rpf.drm()
i1.p <- rpf.rparam(i1)
rpf.prob(i1, c(i1.p), -1)  # low trait score
rpf.prob(i1, c(i1.p), c(0, 1))  # average and high trait score
```

---

**rpf.rescale**

**Rescale item parameters**

Description

Adjust item parameters for changes in mean and covariance of the latent distribution.

Usage

```
rpf.rescale(m, param, mean, cov)
```

Arguments

- **m**: item model
- **param**: item parameters
- **mean**: vector of means
- **cov**: covariance matrix

Examples

```r
spec <- rpf.grm()
p1 <- rpf.rparam(spec)
testPoint <- rnorm(1)
move <- rnorm(1)
cov <- as.matrix(rlnorm(1))
Icov <- solve(cov)
padj <- rpf.rescale(spec, p1, move, cov)
pr1 <- rpf.prob(spec, padj, (testPoint - move) %% Icov)
pr2 <- rpf.prob(spec, p1, testPoint)
abs(pr1 - pr2) < 1e9
```
Generates item parameters

Description

This function generates random item parameters. The version argument is available if you are writing a test that depends on reproducible random parameters (using `set.seed`).

Usage

```r
rpf.rparam(m, version = 2L)
```

Arguments

- `m`: an item model
- `version`: the version of random parameters

Value

item parameters

Examples

```r
i1 <- rpf.drm()
rpf.rparam(i1)
```

Randomly sample response patterns given a list of items

Description

Returns a random sample of response patterns given a list of item models and parameters. If `grp` is given then theta, items, params, mean, and cov can be omitted.

Usage

```r
rpf.sample(  
  theta,  
  items,  
  params,  
  ...,  
  prefix = "i",  
  mean = NULL,  
  cov = NULL,  
  mcar = 0,  
  grp = NULL  
)
```
Arguments

theta
either a vector (for 1 dimension) or a matrix (for >1 dimension) of person abilities or the number of response patterns to generate randomly

items
a list of item models

params
a list or matrix of item parameters. If omitted, random item parameters are generated for each item model.

... Not used. Forces remaining arguments to be specified by name.

prefix
Column names are taken from param or items. If no column names are available, some will be generated using the given prefix.

mean
mean vector of latent distribution (optional)

cov
covariance matrix of latent distribution (optional)

mcar
proportion of generated data to set to NA (missing completely at random)

grp
a list containing the model and data. See the details section.

Value

Returns a data frame of response patterns

Format of a group

A model, or group within a model, is represented as a named list.

spec
list of response model objects

param numeric matrix of item parameters

free logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)

mean numeric vector giving the mean of the latent distribution

cov numeric matrix giving the covariance of the latent distribution

data data.frame containing observed item responses, and optionally, weights and frequencies

score factors scores with response patterns in rows

weightColumn name of the data column containing the numeric row weights (optional)

dataColumn name of the data column containing the integral row frequencies (optional)

qwidth width of the quadrature expressed in Z units

qpoints number of quadrature points

minItemsPerScore minimum number of non-missing items when estimating factor scores

The param matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in spec and order of columns in param are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by param colnames.

Currently only a multivariate normal distribution is available, parameterized by the mean and cov. If mean and cov are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, qwidth=2 and qpoints=5 would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by EAPscores.
See Also

sample

Examples

# 1 dimensional items
i1 <- rpf.drm()
i1.p <- rpf.rparam(i1)
i2 <- rpf.nrm(outcomes=3)
i2.p <- rpf.rparam(i2)
rpf.sample(5, list(i1,i2), list(i1.p, i2.p))

science  Liking for Science dataset

Description

These data are from Wright & Masters (1982, p. 18).

Details

All items were fit to a 3 category Partial Credit Model (PCM) using Ministep 3.75.0.

References


Examples

data(science)

SitemFit  Compute the S fit statistic for a set of items

Description

Runs SitemFit1 for every item and accumulates the results.
Usage

SitemFit(
  grp,
  ..., 
  method = "pearson",
  log = TRUE,
  qwidth = 6,
  qpoints = 49L,
  alt = FALSE,
  omit = 0L,
  .twotier = TRUE,
  .parallel = TRUE
)

Arguments

- **grp**: a list containing the model and data. See the details section.
- **...**: Not used. Forces remaining arguments to be specified by name.
- **method**: whether to use a pearson or rms test
- **log**: whether to return p-values in log units
- **qwidth**: [Deprecated]
- **qpoints**: [Deprecated]
- **alt**: whether to include the item of interest in the denominator
- **omit**: number of items to omit (a single number) or a list of the length the number of items
- **.twotier**: whether to enable the two-tier optimization
- **.parallel**: whether to take advantage of multiple CPUs (default TRUE)

Value

- a list of output from SitemFit1

Format of a group

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
- **cov**: numeric matrix giving the covariance of the latent distribution
- **data**: data.frame containing observed item responses, and optionally, weights and frequencies
- **score**: factors scores with response patterns in rows
- **weightColumn**: name of the data column containing the numeric row weights (optional)
freqColumn name of the data column containing the integral row frequencies (optional)
qwidth width of the quadrature expressed in Z units
qpoints number of quadrature points
minItemsPerScore minimum number of non-missing items when estimating factor scores

The param matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in spec and order of columns in param are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by param colnames.

Currently only a multivariate normal distribution is available, parameterized by the mean and cov. If mean and cov are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, qwidth=2 and qpoints=5 would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by EAPscores.

See Also
Other diagnostic: ChenThissen1997(), SitemFit1(), multinomialFit(), rpf.1dim.fit(), sumScoreEAPTest()

Examples

```r
grp <- list(spec=list())
grp$spec[1:20] <- list(rpf.grm())
grp$param <- sapply(grp$spec, rpf.rparam)
colnames(grp$param) <- paste("i", 1:20, sep="")
grp$mean <- 0
grp$cov <- diag(1)
grp$free <- grp$param != 0
grp$data <- rpf.sample(500, grp=grp)
SitemFit(grp)
```

---

### SitemFit1

Compute the S fit statistic for 1 item

**Description**

Implements the Kang & Chen (2007) polytomous extension to S statistic of Orlando & Thissen (2000). Rows with missing data are ignored, but see the omit option.

**Usage**

```
SitemFit1(
  grp,
  item,
  free = 0,
)```

...,
  method = "pearson",
  log = TRUE,
  qwidth = 6,
  qpoints = 49L,
  alt = FALSE,
  omit = 0L,
  .twotier = TRUE
)

Arguments

- **grp**: a list containing the model and data. See the details section.
- **item**: the item of interest
- **free**: the number of free parameters involved in estimating the item (to adjust the df)
- ...: Not used. Forces remaining arguments to be specified by name.
- **method**: whether to use a pearson or rms test
- **log**: whether to return p-values in log units
- **qwidth**: [Deprecated]
- **qpoints**: [Deprecated]
- **alt**: whether to include the item of interest in the denominator
- **omit**: number of items to omit or a character vector with the names of the items to omit when calculating the observed and expected sum-score tables
- **.twotier**: whether to enable the two-tier optimization

Details

This statistic is good at finding a small number of misfitting items among a large number of well fitting items. However, be aware that misfitting items can cause other items to misfit.

Observed tables cannot be computed when data is missing. Therefore, you can optionally omit items with the greatest number of responses missing relative to the item of interest.

Pearson is slightly more powerful than RMS in most cases I examined.

Setting alt to TRUE causes the tables to match published articles. However, the default setting of FALSE probably provides slightly more power when there are less than 10 items.

The name of the test, "S", probably stands for sum-score.

Format of a group

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
cov numeric matrix giving the covariance of the latent distribution
data data.frame containing observed item responses, and optionally, weights and frequencies
score factors scores with response patterns in rows
weightColumn name of the data column containing the numeric row weights (optional)
freqColumn name of the data column containing the integral row frequencies (optional)
qwidth width of the quadrature expressed in Z units
qpoints number of quadrature points
minItemsPerScore minimum number of non-missing items when estimating factor scores

The param matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in spec and order of columns in param are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by param colnames.

Currently only a multivariate normal distribution is available, parameterized by the mean and cov. If mean and cov are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, qwidth=2 and qpoints=5 would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by EAPscores.

References


See Also

Other diagnostic: `ChenThissen1997()`, `SitemFit()`, `multinomialFit()`, `rpf.1dim.fit()`, `sumScoreEAPTest()`

---

**Description**

In addition, the freqColumn and weightColumn are reset to NULL.

**Usage**

`stripData(grp)`

**Arguments**

- `grp` a list containing the model and data. See the details section.
Value

The same group without associated data.

Format of a group

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
- **cov**: numeric matrix giving the covariance of the latent distribution
- **data**: data.frame containing observed item responses, and optionally, weights and frequencies
- **score**: factors scores with response patterns in rows
- **weightColumn**: name of the data column containing the numeric row weights (optional)
- **freqColumn**: name of the data column containing the integral row frequencies (optional)
- **qwidth**: width of the quadrature expressed in Z units
- **qpoints**: number of quadrature points
- **minItemsPerScore**: minimum number of non-missing items when estimating factor scores

The `param` matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in `spec` and order of columns in `param` are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by `param` colnames.

Currently only a multivariate normal distribution is available, parameterized by the `mean` and `cov`. If `mean` and `cov` are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, `qwidth=2` and `qpoints=5` would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by `EAPscores`.

Examples

```r
spec <- list()
spec[1:3] <- list(rpf.grm(outcomes=3))
param <- sapply(spec, rpf.rparam)
data <- rpf.sample(5, spec, param)
colnames(param) <- colnames(data)
grp <- list(spec=spec, param=param, data=data, minItemsPerScore=1L)
grpscore <- EAPscores(grp)
str(grp)
grp <- stripData(grp)
str(grp)
```
Compute the sum-score EAP table

Description

Observed tables cannot be computed when data is missing. Therefore, you can optionally omit items with the greatest number of responses missing when conducting the distribution test.

Usage

sumScoreEAP(grp, ..., qwidth = 6, qpoints = 49L, .twotier = TRUE)

Arguments

- **grp**: a list containing the model and data. See the details section.
- **...**: Not used. Forces remaining arguments to be specified by name.
- **qwidth**: DEPRECATED
- **qpoints**: DEPRECATED
- **.twotier**: whether to enable the two-tier optimization

Details

When two-tier covariance structure is detected, EAP scores are only reported for primary factors. It is possible to compute EAP scores for specific factors, but it is not clear why this would be useful because they are conditional on the specific factor sum scores. Moreover, the algorithm to compute them efficiently has not been published yet (as of Jun 2014).

Format of a group

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
- **cov**: numeric matrix giving the covariance of the latent distribution
- **data**: data.frame containing observed item responses, and optionally, weights and frequencies
- **score**: factors scores with response patterns in rows
- **weightColumn**: name of the data column containing the numeric row weights (optional)
- **freqColumn**: name of the data column containing the integral row frequencies (optional)
- **qwidth**: width of the quadrature expressed in Z units
- **qpoints**: number of quadrature points
- **minItemsPerScore**: minimum number of non-missing items when estimating factor scores
The `param` matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in `spec` and order of columns in `param` are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by `param` colnames.

Currently only a multivariate normal distribution is available, parameterized by the mean and `cov`. If `mean` and `cov` are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, `qwidth=2` and `qpoints=5` would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by `EAPscores`.

### See Also
Other scoring: `EAPscores()`, `bestToOmit()`, `itemOutcomeBySumScore()`, `observedSumScore()`, `omitItems()`, `omitMostMissing()`

### Examples
```r
# see Thissen, Pommerich, Billeaud, & Williams (1995, Table 2)
spec <- list()
spec[1:3] <- list(rpf.grm(outcomes=4))

param <- matrix(c(1.87, .65, 1.97, 3.14,
                   2.66, .12, 1.57, 2.69,
                   1.24, .08, 2.03, 4.3), nrow=4)
# fix parameterization
param <- apply(param, 2, function(p) c(p[1], p[2:4] * -p[1]))

grp <- list(spec=spec, mean=0, cov=matrix(1,1,1), param=param)
sumScoreEAP(grp)
```

---

### sumScoreEAPTest

**Conduct the sum-score EAP distribution test**

### Description
Conduct the sum-score EAP distribution test

### Usage
```
sumScoreEAPTest(grp, ..., qwidth = 6, qpoints = 49L, .twotier = TRUE)
```

### Arguments
- **grp**
  - a list containing the model and data. See the details section.
- **...**
  - Not used. Forces remaining arguments to be specified by name.
- **qwidth**
  - DEPRECATED
qpoints               DEPRECATED
.twotier             whether to enable the two-tier optimization

Format of a group

A model, or group within a model, is represented as a named list.

spec  list of response model objects
param numeric matrix of item parameters
free  logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
mean numeric vector giving the mean of the latent distribution
cov   numeric matrix giving the covariance of the latent distribution
data  data.frame containing observed item responses, and optionally, weights and frequencies
score factors scores with response patterns in rows
weightColumn name of the data column containing the numeric row weights (optional)
freqColumn name of the data column containing the integral row frequencies (optional)
qwidth width of the quadrature expressed in Z units
qpoints number of quadrature points
minItemsPerScore minimum number of non-missing items when estimating factor scores

The param matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in spec and order of columns in param are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by param colnames.

Currently only a multivariate normal distribution is available, parameterized by the mean and cov. If mean and cov are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, qwidth=2 and qpoints=5 would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by EAPscores.

References


See Also

Other diagnostic: ChenThissen1997(), SitemFit1(), SitemFit(), multinomialFit(), rpf.1dim.fit()
**tabulateRows**

Tabulate data.frame rows

**Description**

Like `tabulate` but entire rows are the unit of tabulation. The data.frame is not sorted, but must be sorted already.

**Usage**

```
tabulateRows(observed)
```

**Arguments**

- `observed`: a sorted data.frame holding ordered factors in every column

**See Also**

`orderCompletely`

**Examples**

```r
df <- as.data.frame(matrix(c(sample.int(2, 30, replace=TRUE)), 10, 3))
df <- df[orderCompletely(df),]
tabulateRows(df)
```

**toFactorLoading**

Convert response function slopes to factor loadings

**Description**

All slopes are divided by the ogive constant. Then the following transformation is applied to the slope matrix.

**Usage**

```
toFactorLoading(slope, ogive = rpf.ogive)
```

**Arguments**

- `slope`: a matrix with items in the columns and slopes in the rows
- `ogive`: the ogive constant (default `rpf.ogive`)

**Details**

\[
\text{slope} \quad \left[ 1 + \text{rowSums}(\text{slope}^2) \right]^{\frac{1}{2}}
\]
Value
a factor loading matrix with items in the rows and factors in the columns

See Also
Other factor model equivalence: fromFactorLoading(), fromFactorThreshold(), toFactorThreshold()

toFactorThreshold    Convert response function intercepts to factor thresholds

Description
Convert response function intercepts to factor thresholds

Usage
toFactorThreshold(intercept, slope, ogive = rpf.ogive)

Arguments
intercept           a matrix with items in the columns and intercepts in the rows
slope               a matrix with items in the columns and slopes in the rows
ogive               the ogive constant (default rpf.ogive)

Value
a factor threshold matrix with items in the columns and factor thresholds in the rows

See Also
Other factor model equivalence: fromFactorLoading(), fromFactorThreshold(), toFactorLoading()

write.flexmirt     Write a flexMIRT PRM file

Description
[Experimental] This was last updated in 2017 and may no longer work.

Usage
write.flexmirt(groups, file = NULL, fileEncoding = "")
Arguments

- **groups**: a list of groups each with items and latent parameters
- **file**: the destination file name
- **fileEncoding**: how to encode the text file (optional)

Details

Formats item parameters in the way that flexMIRT expects to read them.

NOTE: Support for the graded response model may not be complete.

Format of a group

A model, or group within a model, is represented as a named list.

- **spec**: list of response model objects
- **param**: numeric matrix of item parameters
- **free**: logical matrix of indicating which parameters are free (TRUE) or fixed (FALSE)
- **mean**: numeric vector giving the mean of the latent distribution
- **cov**: numeric matrix giving the covariance of the latent distribution
- **data**: data.frame containing observed item responses, and optionally, weights and frequencies
- **score**: factors scores with response patterns in rows
- **weightColumn**: name of the data column containing the numeric row weights (optional)
- **freqColumn**: name of the data column containing the integral row frequencies (optional)
- **qwidth**: width of the quadrature expressed in Z units
- **qpoints**: number of quadrature points
- **minItemsPerScore**: minimum number of non-missing items when estimating factor scores

The **param** matrix stores items parameters by column. If a column has more rows than are required to fully specify a model then the extra rows are ignored. The order of the items in **spec** and order of columns in **param** are assumed to match. All items should have the same number of latent dimensions. Loadings on latent dimensions are given in the first few rows and can be named by setting rownames. Item names are assigned by **param** colnames.

Currently only a multivariate normal distribution is available, parameterized by the **mean** and **cov**. If **mean** and **cov** are not specified then a standard normal distribution is assumed. The quadrature consists of equally spaced points. For example, `qwidth=2` and `qpoints=5` would produce points -2, -1, 0, 1, and 2. The quadrature specification is part of the group and not passed as extra arguments for the sake of consistency. As currently implemented, OpenMx uses EAP scores to estimate latent distribution parameters. By default, the exact same EAP scores should be produced by **EAPscores**.
# Index

* **datasets**
  - rpf.ogive, 49

* **data**
  - kct, 19
  - LSAT6, 20
  - LSAT7, 20
  - science, 54

* **diagnostic**
  - ChenThissen1997, 7
  - multinomialFit, 21
  - rpf.1dim.fit, 30
  - SitemFit, 54
  - SitemFit1, 56
  - sumScoreEAPTest, 61

* **factor model equivalence**
  - fromFactorLoading, 16
  - fromFactorThreshold, 17
  - toFactorLoading, 63
  - toFactorThreshold, 64

* **response model**
  - rpf.drm, 35
  - rpf.gpcmp, 37
  - rpf.grm, 38
  - rpf.grmp, 39
  - rpf.lmp, 42
  - rpf.mcm, 44
  - rpf.nrm, 47

* **scoring**
  - bestToOmit, 6
  - EAPscores, 14
  - itemOutcomeBySumScore, 17
  - observedSumScore, 23
  - omitItems, 24
  - omitMostMissing, 25
  - sumScoreEAP, 60

$rpf\_base\_method(Class\ rpf\_base), 11$

$_rpf\_logprob(rpf\_logprob), 44$

$_rpf\_prob(rpf\_prob), 50$

$_rpf\_rescale(rpf\_rescale), 51$

An introduction, 4

as.IFAgroup, 5

bestToOmit, 6, 15, 18, 24–26, 61

chen.thissen.1997(ChenThissen1997), 7

ChenThissen1997, 7, 22, 30, 32, 56, 58, 62

Class rpf.1dim, 9

Class rpf.1dim.drm, 10

Class rpf.1dim.gpcmp, 10

Class rpf.1dim graded, 10

Class rpf.1dim grm, 10

Class rpf.1dim grmp, 10

Class rpf.1dim lmp, 11

Class rpf.base, 11

Class rpf.mdim, 11

Class rpf.mdim.drm, 11

Class rpf.mdim graded, 11

Class rpf.mdim grm, 12

Class rpf.mdim grmp, 12

Class rpf.mdim nrm, 12

collapseCategoricalCells, 12

crosstabTest, 14

collapseCategoricalCells, 12

crosstabTest, 14

EAPscores, 5–7, 9, 14, 15, 18, 22, 24–26,

30–32, 53, 56, 58, 59, 61, 62, 65

expandDataFrame, 16

fromFactorLoading, 16, 17, 64

fromFactorThreshold, 17, 64

itemOutcomeBySumScore, 7, 15, 17, 24–26,

61

kct, 19

66
logit, 19, 35
LSAT6, 20
LSAT7, 20

multinomialFit, 9, 21, 32, 56, 58, 62
observedSumScore, 7, 15, 18, 23, 25, 26, 61
omitItems, 7, 15, 18, 24, 24, 26, 61
omitMostMissing, 7, 15, 18, 24, 25, 26, 61
orderCompletely, 27, 63
ordinal.gamma, 8, 27
ptw2011.gof.test, 28

read.flexmirt, 29
rpf.1dim-class (Class rpf.1dim), 9
rpf.1dim.drm-class (Class rpf.1dim.drm), 10
rpf.1dim.fit, 9, 22, 30, 56, 58, 62
rpf.1dim.gpcmp-class (Class rpf.1dim.gpcmp), 10
rpf.1dim.graded-class (Class rpf.1dim.graded), 10
rpf.1dim.grm-class (Class rpf.1dim.grm), 10
rpf.1dim.grmp-class (Class rpf.1dim.grmp), 10
rpf.1dim.lmp-class (Class rpf.1dim.lmp), 11
rpf.1dim.moment, 32
rpf.1dim.residual, 33
rpf.1dim.std residual, 34
rpf.base-class (Class rpf.base), 11
rpf.dLL, 34
rpf.dLL, rpf.base, numeric, NULL, numeric-method (rpf.dLL), 34
rpf.dLL, rpf.base, numeric, numeric-method (rpf.dLL), 34
rpf.drm, 4, 35, 38, 39, 41, 43, 45, 47
rpf.dTheta, 36, 40, 42
rpf.dTheta, rpf.base, numeric, matrix, numeric-method (rpf.dTheta), 36
rpf.dTheta, rpf.base, numeric, numeric-method (rpf.dTheta), 36
rpf.gpcmp, 36, 37, 39–41, 43, 45, 47
rpf.grm, 4, 36, 38, 39, 41, 43, 45, 47
rpf.grmp, 36, 38, 39, 43, 45, 47
rpf.id.of, 41
rpf.info, 36, 41
rpf.lmp, 36–41, 42, 45, 47
rpf.logprob, 44
rpf.logprob, rpf.1dim, numeric, matrix-method (rpf.logprob), 44
rpf.logprob, rpf.1dim, numeric-method (rpf.logprob), 44
rpf.logprob, rpf.mdim, numeric, matrix-method (rpf.logprob), 44
rpf.logprob, rpf.mdim, numeric, NULL-method (rpf.logprob), 44
rpf.logprob, rpf.mdim, numeric, numeric-method (rpf.logprob), 44
rpf.mcm, 36, 38, 39, 41, 43, 44, 47
rpf.mdim-class (Class rpf.mdim), 11
rpf.mdim.drm-class (Class rpf.mdim.drm), 11
rpf.mdim.graded-class (Class rpf.mdim.graded), 11
rpf.mdim.grm-class (Class rpf.mdim.grm), 12
rpf.mdim.mcm-class (Class rpf.mdim.mcm), 12
rpf.mdim.nrm-class (Class rpf.mdim.nrm), 12
rpf.mean.info1, 45
rpf.mean.info1, 46
rpf.modify, 46
rpf.modify, rpf.mdim.drm, numeric-method (rpf.modify), 46
rpf.modify, rpf.mdim.graded, numeric-method (rpf.modify), 46
rpf.modify, rpf.mdim.nrm, numeric-method (rpf.modify), 46
rpf.numSpec, 43, 46, 47
rpf.numSpec, rpf.base-method (rpf.numSpec), 48
rpf.paramInfo, rpf.base-method (rpf.paramInfo), 50
rpf.prob, 37, 40, 42, 50
rpf.prob, rpf.1dim, numeric, matrix-method (rpf.prob), 50
rpf.prob, rpf.1dim, numeric, numeric-method
(rpf.prob), 50
rpf.prob, rpf.1dim.grm, numeric, numeric-method
  (rpf.prob), 50
rpf.prob, rpf.base, data.frame, numeric-method
  (rpf.prob), 50
rpf.prob, rpf.base, matrix, matrix-method
  (rpf.prob), 50
rpf.prob, rpf.base, matrix, numeric-method
  (rpf.prob), 50
rpf.prob, rpf.mdim, numeric, matrix-method
  (rpf.prob), 50
rpf.prob, rpf.mdim, numeric, NULL-method
  (rpf.prob), 50
rpf.prob, rpf.mdim, numeric, numeric-method
  (rpf.prob), 50
rpf.prob, rpf.mdim.grm, numeric, matrix-method
  (rpf.prob), 50
rpf.prob, rpf.mdim.grm, numeric, numeric-method
  (rpf.prob), 50
rpf.prob, rpf.mdim.mcm, numeric, matrix-method
  (rpf.prob), 50
rpf.prob, rpf.mdim.nrm, numeric, matrix-method
  (rpf.prob), 50
rpf.rescale, 51
rpf.rescale, rpf.base, numeric, numeric, matrix-method
  (rpf.rescale), 51
rpf.rparam, 5, 52
rpf.rparam, rpf.1dim.drm-method
  (rpf.rparam), 52
rpf.rparam, rpf.1dim.gpcmp-method
  (rpf.rparam), 52
rpf.rparam, rpf.1dim.graded-method
  (rpf.rparam), 52
rpf.rparam, rpf.1dim.grmp-method
  (rpf.rparam), 52
rpf.rparam, rpf.1dim.lmp-method
  (rpf.rparam), 52
rpf.rparam, rpf.mdim.drm-method
  (rpf.rparam), 52
rpf.rparam, rpf.mdim.graded-method
  (rpf.rparam), 52
rpf.rparam, rpf.mdim.mcm-method
  (rpf.rparam), 52
rpf.rparam, rpf.mdim.nrm-method
  (rpf.rparam), 52
rpf.sample, 52
rpf_numParam_wrapper (rpf.numParam), 48
rpf_numSpec_wrapper (rpf.numSpec), 48
rpf_paramInfo_wrapper (rpf.paramInfo), 50
sample, 54
science, 54
sfif (science), 54
sfpf (science), 54
sfsf (science), 54
sfxf (science), 54
SitemFit, 9, 22, 32, 54, 58, 62
SitemFit1, 9, 22, 32, 54–56, 56, 62
stripData, 58
sumScoreEAP, 7, 15, 18, 24–26, 60
sumScoreEAPTest, 9, 22, 32, 56, 58, 61
tabulateRows, 63
toFactorLoading, 17, 63, 64
toFactorThreshold, 17, 64, 64
write.flexmirt, 64