Package ‘ShinyItemAnalysis’

June 26, 2019

Type Package

Title Test and Item Analysis via Shiny

Version 1.3.1

Date 2019-06-26

Author Patricia Martinkova [aut, cre],
Adela Hladka [aut],
Ondrej Leder [ctb],
Jakub Houdek [ctb],
Lubomir Stepanek [ctb],
Tomas Jurica [ctb],
Jana Vorlickova [ctb]

Maintainer Patricia Martinkova <martinkova@cs.cas.cz>

Depends R (>= 3.5.0)

Imports corrplot, cowplot, CTT, data.table, deltaPlotR, DT, difNLR (>= 1.2.2), difR (>= 5.0), gg dendro, ggplot2 (>= 2.2.1), gridExtra, knitr, latticeExtra, ltm, mirt (>= 1.24), moments, msm, nnet, plotly, psych, psychometric, reshape2, rmarkdown, shiny (>= 1.0.3), shinyBS, shinydashboard, shinyjs (>= 0.9), stringr, VGAM, xtable

Description Interactive shiny application for analysis of educational tests and their items.

License GPL-3

LazyData TRUE

RoxygenNote 6.1.1

BugReports https://github.com/patriciamar/ShinyItemAnalysis/issues

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-06-26 11:40:17 UTC
Description

The `dataMedical` dataset consists of the responses of 2,392 subjects (750 males, 1,633 females and 9 subjects without gender specification) to admission test to a medical school. It contains 100 items. A correct answer is coded as 1 and incorrect answer as 0. Missing answers were evaluated as incorrect, i.e. 0.

Usage

data(dataMedical)

Format

A `dataMedical` is a data frame consisting of 2,392 observations on the following 102 variables. The first 100 columns represent dichotomously scored items of the test. The 101st column is vector of gender membership; values 0 and 1 refer to males and females. The 102nd columns in criterion variable; value 1 means that student studies standardly, 0 otherwise (e.g. leaving or interrupting studies).
Author(s)

Cestmir Stuka
First Faculty of Medicine, Charles University in Prague

Martin Vejrazka
First Faculty of Medicine, Charles University in Prague

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
martinkova@cs.cas.cz

Jakub Houdek
Institute of Computer Science, The Czech Academy of Sciences

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

References


See Also
dataMedicaltest, dataMedicalkey, dataMedicalgraded

dataMedicalgraded  Graded Dataset of Admission Test to Medical School

Description

The dataMedicalgraded dataset consists of the responses of 2,392 subjects (750 males, 1,633 females and 9 subjects without gender specification) to multiple-choice admission test to a medical school. It contains 100 items. Each item is graded with 0 to 4 points. Maximum of 4 points were set if all correct answers and none of incorrect answers were selected.

Usage
data(dataMedicalgraded)
Format

A dataMedicalgraded is a data.frame consisting of 2,392 observations on the following 102 variables. The first 100 columns represent graded answers of subject to items of the test. The 101st column is vector of gender membership; values 0 and 1 refer to males and females. The 102nd columns in criterion variable; value 1 means that student study standardly, 0 otherwise (e.g. leaving or interrupting studies).

Author(s)

Cestmir Stuka  
First Faculty of Medicine, Charles University in Prague

Martin Vejrazka  
First Faculty of Medicine, Charles University in Prague

Patricia Martinkova  
Institute of Computer Science, The Czech Academy of Sciences  
martinkova@cs.cas.cz

Jakub Houdek  
Institute of Computer Science, The Czech Academy of Sciences

Adela Hladka  
Institute of Computer Science, The Czech Academy of Sciences  
Faculty of Mathematics and Physics, Charles University  
hladka@cs.cas.cz

References


See Also

dataMedical, dataMedicaltest, dataMedicalkey

dataMedicalkey

<table>
<thead>
<tr>
<th>dataMedicalkey</th>
<th>Key of Correct Answers for dataMedicaltest Dataset</th>
</tr>
</thead>
</table>

Description

The dataMedicalkey is a vector of factors representing correct answers of dataMedicaltest data set.
Usage

data(dataMedicalkey)

Format

A data.frame with 100 values representing correct answers to items of dataMedicaltest dataset. For more details see dataMedicaltest.

Author(s)

Cestmir Stuka
First Faculty of Medicine, Charles University in Prague

Martin Vejrazka
First Faculty of Medicine, Charles University in Prague

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
martinkova@cs.cas.cz

Jakub Houdek
Institute of Computer Science, The Czech Academy of Sciences

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

References


See Also

dataMedical, dataMedicaltest, dataMedicalgraded
**dataMedicaltest**  

**Dataset of Admission Test to Medical School**

**Description**

The `dataMedicaltest` dataset consists of the responses of 2,392 subjects (750 males, 1,633 females and 9 subjects without gender specification) to multiple-choice admission test to a medical school. It contains 100 items, possible answers were A, B, C, D, while any combination of these can be correct.

**Usage**

data(dataMedicaltest)

**Format**

`dataMedicaltest` is a `data.frame` consisting of 2,392 observations on the following 102 variables. The first 100 columns represent answers of subject to items of the test. The 101st column is vector of gender membership; values 0 and 1 refer to males and females. The 102nd columns is criterion variable; value 1 means that student studies standardly, 0 otherwise (e.g. leaving or interrupting studies).

**Author(s)**

Cestmir Stuka  
First Faculty of Medicine, Charles University in Prague

Martin Vejrazka  
First Faculty of Medicine, Charles University in Prague

Patricia Martinkova  
Institute of Computer Science, The Czech Academy of Sciences  
martinkova@cs.cas.cz

Jakub Houdek  
Institute of Computer Science, The Czech Academy of Sciences

Adela Hladka  
Institute of Computer Science, The Czech Academy of Sciences  
Faculty of Mathematics and Physics, Charles University  
hladka@cs.cas.cz
DDplot

References


See Also
dataMedical, dataMedicalkey, dataMedicalgraded

---

DDplot

Graphical representation of difficulty and (generalized) discrimination in item analysis

Description

Plots difficulty and (generalized) discrimination for items ordered by difficulty.

Usage

DDplot(data, item.names, k = 3, l = 1, u = 3, discrim = "ULI", maxscore, minscore, bin = FALSE, cutscore, average.score = FALSE)

Arguments

data numeric: binary or ordinal data matrix or data frame. See Details.
item.names character: the names of items.
k numeric: number of groups to which may be data.frame x divided by the total score. Default value is 3. See Details.
l numeric: lower group. Default value is 1. See Details.
u numeric: upper group. Default value is 3. See Details.
discrim character: type of discrimination index to be calculated. Default value is "ULI". See Details.
maxscore vector or numeric: maximal scores of items. If numeric, the same maximal score is used for all items. If missing, vector of achieved maximal scores is calculated and used in calculations.
minscore vector or numeric: minimal scores of items. If numeric, the same minimal score is used for all items. If missing, vector of achieved minimal scores is calculated and used in calculations.
bin logical: should the ordinal data be binarized. Default value is FALSE. See Details.
cutscore vector or numeric: cutscore used to binarize the data set. If numeric, the same cutscore is used for all items. If missing, vector of maximal scores is used in calculations.
average.score logical: should average score of the item displayed instead of difficulty. Default value is FALSE. See Details.
Details

The data is a matrix or data frame whose rows represent examinee answers (1 correct, 0 incorrect, or ordinal item scores) and columns correspond to the items.

The item.names argument stands for names of items. If not specified, the names of dataset columns are used. Difficulty and discrimination indices are plotted for each item, items are ordered by their difficulty.

Discrimination is calculated using method specified in discrim. Default option "ULI" calculates difference in ratio of correct answers in upper and lower third of students. "RIT" index calculates correlation between item score and test total score. "RIR" index calculates correlation between item score and total score for the rest of the items. With option "none", only difficulty is displayed.

"ULI" index can be generalized using arguments k, l and u. Generalized ULI discrimination is then computed as follows: The function takes data on individuals, computes their total test score and then divides individuals into k groups. The lower and upper group are determined by l and u parameters, i.e. l-th and u-th group where the ordering is defined by increasing total score.

For ordinal data, difficulty is defined as relative score (achieved - minimal)/(maximal - minimal). Minimal score can be specified by minscore, maximal score can be specified by maxscore. Average score of items can be displayed with argument average.score = T. Note that for binary data difficulty estimate is the same as average score of the item.

Binarization of data is allowed in bin, for this purpose cutscore is used.

Note

Generalized discrimination is calculated by gDiscrim function.

Author(s)

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

Lubomir Stepanek
First Faculty of Medicine, Charles University

Jana Vorlickova
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
martinkova@cs.cas.cz
References


See Also

gDiscrim, discrim

Examples

```r
## Not run:
# loading 100-item medical admission test data sets
data(dataMedical, dataMedicalgraded)
# binary data set
dataBin <- dataMedical[, 1:100]
# ordinal data set
dataOrd <- dataMedicalgraded[, 1:100]

# DDplot of binary data set
DDplot(dataBin)
# compared to DDplot using ordinal data set and 'bin = TRUE'
DDplot(dataOrd, bin = TRUE)
# compared to binarized data set using bin = TRUE and cutscore equal to 3
DDplot(dataOrd, bin = TRUE, cutscore = 3)

# DDplot of binary data using generalized ULI
# discrimination based on 5 groups, comparing 4th and 5th
DDplot(dataBin, k = 5, l = 4, u = 5)

# DDplot of ordinal data set using ULI
DDplot(dataOrd)
# DDplot of ordinal data set using generalized ULI
# discrimination based on 5 groups, comparing 4th and 5th
DDplot(dataOrd, k = 5, l = 4, u = 5)
# DDplot of ordinal data set using RIT
DDplot(dataOrd, discrim = "RIT")
# DDplot of ordinal data set using RIR
DDplot(dataOrd, discrim = "RIR")
# DDplot of ordinal data set displaying only difficulty
DDplot(dataBin, discrim = "none")

# DDplot of ordinal data set displaying difficulty estimates
DDplot(dataOrd)
# DDplot of ordinal data set displaying average item scores
DDplot(dataOrd, average.score = TRUE)

## End(Not run)
```
DistractorAnalysis  Function for item distractor analysis

Description

Performs distractor analysis for each item and optional number of groups.

Usage

DistractorAnalysis(data, key, p.table = FALSE, num.groups = 3, matching = NULL, match.discrete = FALSE, cut.points)

Arguments

data character: data matrix or data frame. See Details.
key character: answer key for the items.
p.table logical: should the function return the proportions. If FALSE (default) the counts are returned.
num.groups numeric: number of groups to that should be respondents splitted.
matching numeric: numeric vector. If not provided, total score is calculated and distractor analysis is performed based on it.
match.discrete logical: is matching discrete? Default value is FALSE. See details.
cut.points numeric: numeric vector specifying cut points of matching. See details.

Details

This function is adapted version of distractor.analysis function from CTT package.
The data is a matrix or data frame whose rows represents unscored item response from a multiple-choice test and columns correspond to the items.
The key must be a vector of the same length as ncol(data).
In case, no matching is provided, the scores are calculated using the item data and key. The respondents are by default splitted into the num.groups-quantiles and the number (or proportion) of respondents in each quantile is reported with respect to their answers. In case that matching is discrete (match.discrete = TRUE), matching is splitted based on its unique levels. Other cut points can be specified via cut.points argument.

Author(s)

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
See Also
distractor.analysis

Examples

# Not run:
# loading 100-item medical admission test data
data(dataMedicaltest, dataMedicalkey)
data <- dataMedicaltest[, 1:100]
dataBin <- dataMedical[, 1:100]
key <- unlist(dataMedicalkey)

# distractor analysis for dataMedicaltest data set
DistractorAnalysis(data, key)

# distractor analysis for dataMedicaltest data set with proportions
DistractorAnalysis(data, key, p.table = T)

# distractor analysis for dataMedicaltest data set for 6 groups
DistractorAnalysis(data, key, num.group = 6)

# distractor analysis for dataMedicaltest using specified matching
matching <- round(rowSums(dataBin), -1)
DistractorAnalysis(data, key, matching = matching)

# distractor analysis for dataMedicaltest using discrete matching
DistractorAnalysis(data, key, matching = matching, match.discrete = T)

# distractor analysis for dataMedicaltest using groups specified by cut.points
DistractorAnalysis(data, key, cut.points = seq(10, 100, 10))

# End(Not run)

gDiscrim

Generalized Item Discrimination

Description

gDiscrim function computes various generalizations of discrimination index ULI. It enumerates the ability of item to distinguish between individuals from upper (U) vs. lower (L) ability groups, i.e. between respondents with high vs. low overall score on the test. Number of groups, as well as upper and lower groups can be specified by user. Maximal and minimal score in ordinal data sets can be specified by user.
Usage

gDiscrim(x, k = 3, l = 1, u = 3, maxscore, minscore)

Arguments

- **x**: matrix or data.frame of items to be examined. Rows represent persons, columns represent items.
- **k**: numeric: number of groups to which may be data.frame x divided by the total score. Default value is 3. See **Details**.
- **l**: numeric: lower group. Default value is 1. See **Details**.
- **u**: numeric: upper group. Default value is 3. See **Details**.
- **maxscore**: numeric: maximal score in ordinal items. If missing, vector of obtained maximal scores is imputed. See **Details**.
- **minscore**: numeric: minimal score in ordinal items. If missing, vector of obtained minimal scores is imputed. See **Details**.

Details

The function computes total test scores for all respondents and then divides the respondents into k groups. The lower and upper groups are determined by l and u parameters, i.e. l-th and u-th group where the ordering is defined by increasing total score.

In ordinal items, difficulty is calculated as difference of average score divided by range (maximal possible score maxscore minus minimal possible score minscore for given item).

Discrimination is calculated as difference in difficulty between upper and lower group.

Note

gDiscrim is used by **DPlot** function.

Author(s)

- Adela Hladka  
  Institute of Computer Science, The Czech Academy of Sciences  
  Faculty of Mathematics and Physics, Charles University  
  hladka@cs.cas.cz

- Lubomir Stepanek  
  First Faculty of Medicine, Charles University

- Jana Vorlickova  
  Institute of Computer Science, The Czech Academy of Sciences  
  Faculty of Mathematics and Physics, Charles University

- Patricia Martinkova  
  Institute of Computer Science, The Czech Academy of Sciences  
  martinkova@cs.cas.cz
ggWrightMap

Wright Map using ggplot

Description

This function allows to generate Wright Map (also called item-person map) using ggplot function from package ggplot2 and plot_grid function from cowplot. Wright Map is used to display histogram of factor scores and the item difficulty parameters estimated by the Rasch IRT model.

Usage

ggWrightMap(theta, b, binwidth = 0.5, color = "blue", size = 15, item.names)
Arguments

theta numeric: vector of ability estimates.
b numeric: vector of difficulty estimates.
binwidth numeric: the width of the bins of histogram.
color character: color of histogram.
size text size in pts.
item.names names of items to be displayed.

Author(s)

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
martinkova@cs.cas.cz

References


See Also

wrightMap

Examples

```r
## Not run:
library(mirt)

# loading 100-item medical admission test data sets
data(dataMedical)
# binary data set
dataBin <- dataMedical[, 1:100]

# fit Rasch model with mirt package
fit <- mirt(dataBin, model = 1, itemtype = "Rasch")
# factor scores
theta <- as.vector(fscores(fit))
# difficulty estimates
b <- coef(fit, simplify = T)$items[, "d"]

ggWrightMap(theta, b)

item.names <- paste("Item", 1:20)
ggWrightMap(theta, b, item.names = item.names)
```
**Homeostasis Concept Inventory Dichotomous Dataset**

**Description**

The (HCI) dataset consists of the dichotomously scored responses of 651 students (405 males, 246 females) to Homeostasis Concept Inventory multiple-choice test. It contains 20 items, vector of gender membership and identifier whether students plan to major.

**Usage**

```r
data(HCI)
```

**Format**

HCI is a data.frame consisting of 651 observations on the 22 variables. First 20 variables represent dichotomously scored responses to multiple-choice items (1 means correct, 0 is incorrect). 21st column is a vector of gender membership; values 0 and 1 refer to males and females. 22nd column is an identifier whether students planning to major in the life sciences.

**Author(s)**

Jenny L. McFarland  
Biology Department, Edmonds Community College

**References**


**See Also**

HCItest, HCIkey
Key of Correct Answers for Homeostasis Concept Inventory Dataset

Description

The HCIkey is a vector of factors representing correct answers of HCItest dataset.

Usage

data(HCIkey)

Format

A data.frame with 20 values representing correct answers to items of HCItest dataset. For more details see HCItest.

Author(s)

Jenny L. McFarland
Biology Department, Edmonds Community College

References


See Also

HCI, HCItest

Homeostasis Concept Inventory Dataset

Description

(HCIttest) dataset consists of the responses of 651 students (405 males, 246 females) to Homeostasis Concept Inventory multiple-choice test. It contains 20 items, vector of gender membership and identifier whether students plan to major.

Usage

data(HCIttest)
Format

$hcitest$ is a data frame consisting of 651 observations on the 22 variables. First 20 variables represent responses to multiple-choice items. 21st column is a vector of gender membership; values 0 and 1 refer to males and females. 22nd column is a identifier whether students planning to major in the life sciences.

Author(s)

Jenny L. McFarland
Biology Department, Edmonds Community College

References


See Also

$hci$, $hcikey$

---

**ItemAnalysis**  
*Item Analysis*

**Description**

ItemAnalysis function computes various traditional item analysis indices including difficulty, discrimination and item validity. For ordinal items the difficulty and discrimination indices take into account minimal item score as well as range.

**Usage**

```r
ItemAnalysis(data, y = NULL, k = 3, l = 1, u = 3, maxscore, minscore, cutscore, add.bin = FALSE)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>data</code></td>
<td>matrix or data.frame of items to be examined. Rows represent respondents, columns reperesent items.</td>
</tr>
<tr>
<td><code>y</code></td>
<td>vector of criterion values.</td>
</tr>
<tr>
<td><code>k</code></td>
<td>numeric: number of groups to which may be data.frame x divided by the total score. Default value is 3. See Details.</td>
</tr>
<tr>
<td><code>l</code></td>
<td>numeric: lower group. Default value is 1. See Details.</td>
</tr>
<tr>
<td><code>u</code></td>
<td>numeric: upper group. Default value is 3. See Details.</td>
</tr>
<tr>
<td><code>maxscore</code></td>
<td>numeric or vector: maximal score in ordinal items. If missing, vector of obtained maximal scores is imputed. See Details.</td>
</tr>
</tbody>
</table>
**minscore** numeric or vector: minimal score in ordinal items. If missing, vector of obtained minimal scores is imputed. See **Details**.

**cutscore** numeric or vector: cut score used for binarization of ordinal data. If missing, vector of maximal scores is imputed. See **Details**.

**add.bin** logical: If TRUE, indices are printed also for binarized data. See **Details**.

**Details**

For ordinal items the difficulty and discrimination indices take into account minimal item score as well as range.

For calculation of discrimination ULI index, it is possible to specify the number of groups $k$, and which two groups $l$ and $u$ are to be compared.

In ordinal items, difficulty is calculated as difference of average score divided by range (maximal possible score $maxscore$ minus minimal possible score $minscore$).

If **add.bin** is set to TRUE, item analysis of binarized data is included in the output table. In such a case, **cutscore** is used for binarization. When binarizing the data, values greater or equal to cut-score are set to 1, other values are set to 0.

**Value**

**ItemAnalysis** function computes various traditional item analysis indices. Output is a data.frame with following columns:

- **Difficulty** average score of the item divided by its range
- **Average score** average score of the item
- **SD** standard deviation of the item score
- **SD bin** standard deviation of the item score for binarized data
- **Correct answers** proportion of correct answers
- **Min score** minimal score specified in **minscore**; if not provided, observed minimal score
- **Max score** maximal score specified in **maxscore**; if not provided, observed maximal score
- **Obtained min** observed minimal score
- **Obtained max** observed maximal score
- **Cut score** cut-score specified in **cutscore**
- **ULI** generalized ULI
- **ULI default** discrimination with ULI
- **RIT** correlation between item score and overall test score
- **RIR** correlation between item score and overall test score
- **Item criterion** correlation of item score with criterion
- **Item reliability** item reliability index calculated as $\text{cor(item, test)} \times \sqrt{(N-1)/N} \times \text{var(item)}$, see Allen & Yen (1979), Ch.6.4
- **Item reliability woi** item reliability index (scored without item)
Item validity index calculated as \[ \text{cor(item, y)} \times \sqrt{((N-1)/N) \times \text{var(item)}} \], see Allen & Yen (1979), Ch.6.4

Item criterion correlation between item and criterion y

Alpha drop Cronbach's alpha without given item

With add.bin == TRUE, indices based on binarized data set are also provided and marked with bin.

Author(s)

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
martinkova@cs.cas.cz

Jana Vorlickova
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

References


See Also

DDplot, gDiscrim

Examples

```r
# Not run:
# loading 100-item medical admission test data sets
data(dataMedical, dataMedicalgraded)
# binary data set
dataBin <- dataMedical[, 1:100]
# ordinal data set
dataOrd <- dataMedicalgraded[, 1:100]
# study success is the same for both data sets
StudySuccess <- dataMedical[, 102]

# item analysis for binary data
head(ItemAnalysis(dataBin))
```
plotAdjacent

Function for plotting category probabilities of adjacent logistic regression model

Description

Function for plotting category probabilities function estimated by vglm() from VGAM package

Usage

plotAdjacent(x, matching.name = "matching")

Arguments

- `x`: object of class `vglm`
- `matching.name`: character; name of matching criterion used for estimation in `x`.

Author(s)

- Tomas Jurica  
  Faculty of Mathematics and Physics, Charles University

- Adela Hladka  
  Institute of Computer Science, The Czech Academy of Sciences  
  Faculty of Mathematics and Physics, Charles University  
  hladka@cs.cas.cz

- Patricia Martinkova  
  Institute of Computer Science, The Czech Academy of Sciences  
  martinkova@cs.cas.cz

See Also

- `vglm`
plotCumulative

Examples

## Not run:
# loading data
data <- dataMedicalgraded[, 1:100]

# total score calculation
score <- apply(data, 1, sum)
data[, 1] <- ordered(factor(data[, 1], levels = 0:max(data[, 1])))

# cumulative logistic model for item 1
fit <- vglm(data[, 1] ~ score, family = acat(reverse = FALSE, parallel = TRUE))
# coefficients for item 1
coefs <- coef(fit)

plotAdjacent(fit, matching.name = "Total score")

## End(Not run)

---

plotCumulative Function for plotting cumulative and category probabilities of cumulative logistic regression model

Description

Function for plotting cumulative and category probabilities function estimated by vglm() from VGAM package

Usage

plotCumulative(x, type = "cumulative", matching.name = "matching")

Arguments

- x: object of class vglm
- type: character; type of plot to be displayed. Options are "cumulative" (default) for cumulative probabilities and "category" for category probabilities.
- matching.name: character; name of matching criterion used for estimation in x.

Author(s)

- Tomas Jurica
  Faculty of Mathematics and Physics, Charles University

- Adela Hladka
  Institute of Computer Science, The Czech Academy of Sciences
  Faculty of Mathematics and Physics, Charles University
Function for characteristic curve of DIF IRT model

Plots characteristic curve of IRT model.

Usage

plotDIFirt(parameters, test = "Lord", item = "all", item.name, same.scale = F)

Arguments

parameters numeric: data matrix or data frame. See Details.

test character: type of statistic to be shown. See Details.

item either character ("all"), or numeric vector, or single number corresponding to column indicators. See Details.
plotDIForm

    item.name character: the name of item.
    same.scale logical: are the item parameters on the same scale? (default is "FALSE"). See Details.

Details

This function plots characteristic curve of DIF IRT model.

The parameters matrix has a number of rows equal to twice the number of items in the data set.
The first J rows refer to the item parameter estimates in the reference group, while the last J ones correspond to the same items in the focal group. The number of columns depends on the selected IRT model: 2 for the 1PL model, 5 for the 2PL model, 6 for the constrained 3PL model and 9 for the unconstrained 3PL model. The columns of irtParam have to follow the same structure as the output of itemParEst, difLord or difRaju command from difR package.

Two possible type of test statistics can be visualized - "Lord" gives only characteristic curves, "Raju" also highlights area between these curves.
For default option "all", all characteristic curves are plotted.

Author(s)

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
martinkova@cs.cas.cz

See Also

itemParEst
difLord
difRaju

Examples

## Not run:
# loading libraries
library(difR)
library(ltm)

  # loading data based on GMAT2
data(GMAT2, package = "difNLR")

  # Estimation of 2PL IRT model and Lord's statistic
  # by difR package
  fitLord <- difLord(GMAT2, group = 21, focal.name = 1, model = "2PL")
plotDIFLogistic

Function for characteristic curve of 2PL logistic DIF model

Description

Plots characteristic curve of 2PL logistic DIF model

Usage

plotDIFLogistic(data, group, type = "both", item, item.name,
IRT = F, p.adjust.method = "none", purify = F, match = "score",
group.names = c("Reference", "Focal"))

Arguments

data numeric: the data matrix. See Details.
group numeric: the vector of group membership. See Details.
type character: a character string specifying which DIF effects must be tested. Possible values are "both" (default), "udif" and "nudif". See Details.
item numeric: number of item to be plotted
item.name character: the name of item.
IRT logical: if IRT parameterization (TRUE, default) or classic logistic parameterization (FALSE) may be applied.
purify logical: if item purification may be applied.
match specifies the type of matching criterion. Can be either "score" (default) to compute the test score, or any continuous or discrete variable with the same length as the number of rows of Data.
group.names character: names of reference and focal group.

Details

This function plots characteristic curve of 2PL logistic DIF model.
plotDistractorAnalysis

Author(s)

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
martinkova@cs.cas.cz

Examples

```r
## Not run:
# loading libraries
library(difNLR, difR)

# loading data based on GMAT
data(GMAT, package = "difNLR")
data <- GMAT[, 1:20]
group <- GMAT[, 21]

# Characteristic curve by logistic regression model
plotDIFLogistic(data, group, item = 1)

# Characteristic curve by logistic regression model using scaled score
plotDIFLogistic(data, group, item = 1, IRT = T)

# Renaming reference and focal group
plotDIFLogistic(data, group, item = 1, group.names = c("Group 1", "Group 2"))

## End(Not run)
```

plotDistractorAnalysis

*Function for graphical representation of item distractor analysis*

Description

Plots graphical representation of item distractor analysis with proportions and optional number of groups.

Usage

```r
plotDistractorAnalysis(data, key, num.groups = 3, item = 1, item.name, multiple.answers = TRUE, matching = NULL, match.discrete = FALSE, cut.points)
```
Arguments

- **data**: character: data matrix or data frame. See Details.
- **key**: character: answer key for the items.
- **num.groups**: numeric: number of groups to that should be respondents splitted.
- **item**: numeric: the number of item to be plotted.
- **item.name**: character: the name of item.
- **multiple.answers**: logical: should be all combinations plotted (default) or should be answers splitted into distractors. See Details.
- **matching**: numeric: numeric vector. If not provided, total score is calculated and distractor analysis is performed based on it.
- **match.discrete**: logical: is matching discrete? Default value is FALSE. See details.
- **cut.points**: numeric: numeric vector specifying cut points of matching. See details.

Details

This function is graphical representation of `DistractorAnalysis` function. In case, no matching is provided, the scores are calculated using the item data and key. The respondents are by default splitted into the num.groups-quantiles and the proportions of respondents in each quantile are displayed with respect to their answers. In case that matching is discrete (match.discrete = TRUE), matching is splitted based on its unique levels. Other cut points can be specified via cut.points argument.

The data is a matrix or data frame whose rows represents unscored item response from a multiple-choice test and columns correspond to the items.

The key must be a vector of the same length as ncol(data).

If multiple.answers = TRUE (default) all reported combinations of answers are plotted. If multiple.answers = FALSE all combinations are splitted into distractors and only these are then plotted with correct combination.

Author(s)

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
martinkova@cs.cas.cz

See Also

- `DistractorAnalysis`
- `distractor.analysis`
Examples

```r
# Not run:
# loading 100-item medical admission test data
data(dataMedicaltest, dataMedicalkey)
data <- dataMedicaltest[, 1:100]
dataBin <- dataMedical[, 1:100]
key <- unlist(dataMedicalkey)

# distractor plot for items 48, 57 and 32 displaying distractors only
plotDistractorAnalysis(data, key, item = 48, multiple.answers = F)
# correct answer B does not function well
plotDistractorAnalysis(data, key, item = 57, multiple.answers = F)
# all options function well, thus the whole item discriminates well
plotDistractorAnalysis(data, key, item = 32, multiple.answers = F)
# functions well, thus the whole item discriminates well

# distractor plot for items 48, 57 and 32 displaying all combinations
plotDistractorAnalysis(data, key, item = 48)
plotDistractorAnalysis(data, key, item = 57)
plotDistractorAnalysis(data, key, item = 32)

# distractor plot for item 57 with all combinations and 6 groups
plotDistractorAnalysis(data, key, item = 57, num.group = 6)

# distractor plot for item 57 using specified matching
matching <- round(rowSums(dataBin), -1)
plotDistractorAnalysis(data, key, item = 57, matching = matching)

# distractor plot for item 57 using discrete matching
plotDistractorAnalysis(data, key, item = 57, matching = matching, match.discrete = T)

# distractor plot for item 57 using groups specified by cut.points
plotDistractorAnalysis(data, key, item = 57, cut.points = seq(10, 100, 10))

# End(Not run)
```

---

**plotMultinomial**

Function for plotting category probabilities of multinomial log-linear regression model

**Description**

Plots category probabilities functions estimated by `multinom()` from `nnet` package.

**Usage**

```
plotMultinomial(x, matching, matching.name = "matching")
```
Arguments

- `x`: object of class `multinom`
- `matching`: numeric: vector of matching criterion used for estimation in `x`.
- `matching.name`: character: name of matching criterion used for estimation in `x`.

Author(s)

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

Tomas Jurica
Faculty of Mathematics and Physics, Charles University

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
martinkova@cs.cas.cz

See Also

- `multinom`

Examples

```r
## Not run:
# loading data
data(GMAT, GMATtest, GMATkey, package = "difNLR")

matching <- scale(apply(GMAT[, 1:20], 1, sum)) # Z-score
data <- GMATtest[, 1:20]
key <- GMATkey

# multinomial model for item 1
fit <- nnet::multinom(relevel(data[, 1], ref = paste(key[1])) ~ matching)

# plotting category probabilities
plotMultinomial(fit, matching, matching.name = "Z-score")

## End(Not run)
```
startShinyItemAnalysis

This function will start ShinyItemAnalysis application.

Description

An interactive shiny application for running test and item analysis.

Usage

startShinyItemAnalysis()

Author(s)

Patricia Martinkova
Institute of Computer Science, The Czech Academy of Sciences
martinkova@cs.cas.cz

Adela Hladka
Institute of Computer Science, The Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
hladka@cs.cas.cz

Examples

```r
## Not run:
rm(list = ls())
startShinyItemAnalysis()

## End(Not run)
```

theme_app

Complete theme for ShinyItemAnalysis graphics

Description

This complete theme is based on theme_bw and it was modified for purposes of ShinyItemAnalysis.

Usage

```r
theme_app(base_size = 15, base_family = "")
```
Arguments

- base_size: base font size
- base_family: base font family

See Also

ggtheme

Examples

```r
# Not run:
data(GMAT)
data <- GMAT[, 1:20]
# total score calculation
df <- data.frame(score = apply(data, 1, sum))
# histogram
g <- ggplot(df, aes(score)) +
geom_histogram(binwidth = 1) +
xlab("Total score") +
ylab("Number of respondents")

g
g + theme_app()

# End(Not run)
```
# Index

*Topic **datasets**
  - dataMedical, 2
  - dataMedicalgraded, 3
  - dataMedicalkey, 4
  - dataMedicaltest, 6
  - HCI, 15
  - HCIkey, 16
  - HCItest, 16
  - dataMedical, 2, 4, 5, 7
  - dataMedicalgraded, 3, 3, 5, 7
  - dataMedicalkey, 3, 4, 4, 7
  - dataMedicaltest, 3-5, 6
  - DDplot, 7, 12, 13, 19
  - difLord, 23
  - difRaju, 23
  - discrim, 9
  - distractor.analysis, 10, 11, 26
  - DistractorAnalysis, 10, 26
  - gDiscrim, 8, 9, 11, 19
  - ggtheme, 30
  - ggWrightMap, 13
  - HCI, 15, 16, 17
  - HCIkey, 15, 16, 17
  - HCItest, 15, 16, 16
  - ItemAnalysis, 17
  - itemParEst, 23
  - multinom, 28
  - plotAdjacent, 20
  - plotCumulative, 21
  - plotDIFirt, 22
  - plotDIFLogistic, 24
  - plotDistractorAnalysis, 25
  - plotMultinomial, 27
  - startShinyItemAnalysis, 29

  theme_app, 29
  vglm, 20, 22
  wrightMap, 14

31