

Package ‘ShinyItemAnalysis’

June 1, 2021

Type Package

Title Test and Item Analysis via Shiny

Version 1.3.7

Date 2021-05-28

Description Package including functions and interactive shiny application for the psychometric analysis of educational tests, psychological assessments, health-related and other types of multi-item measurements, or ratings from multiple raters.

License GPL-3

URL <http://www.ShinyItemAnalysis.org>,
<https://CRAN.R-project.org/package=ShinyItemAnalysis>

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

Depends R (>= 3.5.0)

Imports data.table, deltaPlotR, difNLR (>= 1.3.2), difR (>= 5.0), dplyr, DT, ggdendro, ggplot2, gridExtra, knitr, latticeExtra, lme4, ltm, magrittr, mirt (>= 1.24), msm, nnet, plotly, psych, psychometric, purrr, rlang, rmarkdown, rstudioapi, scales, shiny (>= 1.0.3), shinyBS, shinydashboard, shinyjs (>= 0.9), stringr, tibble, tidyr, VGAM, xtable

Encoding UTF-8

LazyData TRUE

RoxygenNote 7.1.1

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2021-05-31 22:30:02 UTC

R topics documented:

ShinyItemAnalysis-package	3
AIBS	5
BFI2	7
CZmatura	8
CZmaturaS	9
dataMedical	10
dataMedicalgraded	11
dataMedicalkey	12
dataMedicaltest	13
DDplot	14
DistractorAnalysis	17
fa_parallel	19
gDiscrim	22
ggWrightMap	24
HCI	25
HCIdata	26
HCIgrads	27
HCIkey	28
HCIprepost	29
HCItest	30
HCItestretest	31
ICCrestricted	32
ItemAnalysis	34
LearningToLearn	37
MSclinical	39
NIH	40
plot.sia_parallel	42
plotAdjacent	42
plotCumulative	44
plotDIFirt	45
plotDIFLogistic	47
plotDistractorAnalysis	48
plotMultinomial	50
plot_corr	52
recode_nr	54
startShinyItemAnalysis	56
TestAnxietyCor	57
theme_app	58

ShinyItemAnalysis-package

ShinyItemAnalysis: Test and Item Analysis via shiny

Description

The ShinyItemAnalysis package contains an interactive shiny application for the psychometric analysis of educational tests, psychological assessments, health-related and other types of multi-item measurements, or ratings from multiple raters, which can be accessed using function `startShinyItemAnalysis()`. The shiny application covers a broad range of psychometric methods and offers data examples, model equations, parameter estimates, interpretation of results, together with a selected R code, and is therefore suitable for teaching psychometric concepts with R. It also allows the users to upload and analyze their own data and to automatically generate analysis reports in PDF or HTML.

Besides, the package provides its own functions for test and item analysis within classical test theory framework (e.g., functions `gDiscrim()`, `ItemAnalysis()`, `DistractorAnalysis()`, or `DDplot()`), using various regression models (e.g., `plotCumulative()`, `plotAdjacent()`, `plotMultinomial()`, or `plotDIFLogistic()`), and under IRT framework (e.g., `ggWrightMap()`, or `plotDIFirt()`).

Package also contains several demonstration datasets including the HCI dataset from the reference paper by Martinkova and Drabinova (2018).

Details

Package: ShinyItemAnalysis

Type: Package

Version: 1.3.7

Date: 2021-03-20

Depends: R (>= 3.5.0)

Imports: cowplot, data.table, deltaPlotR, difNLR (>= 1.3.2), difR (>= 5.0), dplyr, DT, gg dendro, ggplot2, gridExtra, knitr, latticeExtra, lme4, ltm, magrittr, mirt (>= 1.24), msm, nnet, plotly, psych, psychometric, purrr, rlang, rmarkdown, rstudioapi, scales, shiny (>= 1.0.3), shinyBS, shinydashboard, shinyjs (>= 0.9), stringr, tibble, tidyr, VGAM, xtable License: GPL-3

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

Website: <http://www.ShinyItemAnalysis.org/>

Encoding: UTF-8

Functions

- [startShinyItemAnalysis](#)
- [DDplot](#)
- [DistractorAnalysis](#)
- [plotDistractorAnalysis](#)

- `gDiscrim`
- `ggWrightMap`
- `ItemAnalysis`
- `plotAdjacent`, `plotCumulative`, `plotMultinomial`
- `plotDIFirt`, `plotDIFLogistic`
- `plot_corr`
- `recode_nr`
- `ICCrestricted`
- `fa_parallel`

Datasets

- `AIBS`
- `CZmatura`
- `CZmaturaS`
- `dataMedical`
- `dataMedicalgraded`
- `dataMedicalkey`
- `dataMedicaltest`
- `HCI`
- `HCIdata`
- `HCIgrads`
- `HCIkey`
- `HCIprepost`
- `HCItest`
- `HCItestretest`
- `LearningToLearn`
- `NIH`

Author(s)

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Education, Charles University
<martinkova@cs.cas.cz>

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
<hladka@cs.cas.cz>

References

Martinkova, P., & Drabinova, A. (2018). ShinyItemAnalysis for teaching psychometrics and to enforce routine analysis of educational tests. *The R Journal*, 10(2), 503–515, doi: [10.32614/RJ-2018074](https://doi.org/10.32614/RJ-2018074)

See Also

Useful links:

- <http://www.ShinyItemAnalysis.org>
- <https://CRAN.R-project.org/package=ShinyItemAnalysis>
- Report bugs at <https://github.com/patriciamar/ShinyItemAnalysis/issues>

AIBS

AIBS grant peer review scoring dataset

Description

The AIBS dataset (Gallo, 2020) comes from the scientific peer review facilitated by the American Institute of Biological Sciences (AIBS) of biomedical applications from and intramural collaborative biomedical research program for 2014–2017. For each proposal, three assigned individual reviewers were asked to provide scores and commentary for the following application criteria: Innovation, Approach/Feasibility, Investigator, and Significance (Impact added as scored criterion in 2014). Each of these criteria is scored on a scale from 1.0 (best) to 5.0 (worst) with a 0.1 gradation, as well as an overall score (1.0–5.0 with a 0.1 gradation). Asynchronous discussion was allowed, although few scores changed post-discussion. The data includes reviewers' self-reported expertise scores (1/2/3, 1 is high expertise) relative to each proposal reviewed, and reviewer / principal investigator demographics. A total of 72 applications ("Standard" or "Pilot") were reviewed in 3 review cycles. The success rate was 34–38%. Application scores indicate where each application falls among all practically possible applications in comparison with the ideal standard of quality from a perfect application. The dataset was used by Erosheva et al. (2021a) to demonstrate issues of inter-rater reliability in case of restricted samples. For details, see Erosheva et al. (2021b).

Usage

```
data(AIBS)
```

Format

AIBS is a `data.frame` consisting of 216 observations on 25 variables. Data describes 72 proposals with 3 ratings each.

ID Proposal ID.

Year Year of the review.

PropType Proposal type; "Standard" or "Pilot".

PIID Anonymized ID of principal investigator (PI).

PIOrgType PI's organization type.

PIGender PI's gender membership; "1" females, "2" males.

PIRank PI's rank; "3" full professor, "1" assistant professor.

PIDegree PI's degree; "1" PhD, "2" MD, "3" PhD/MD.

Innovation Innovation score.

Approach Approach score.

Investig Investigator score.

Signif Significance score.

Impact Impact score.

Score Scientific merit (overall) score.

ScoreAvg Average of the three overall scores from three different reviewers.

ScoreAvgAdj Average of the three overall scores from three different reviewers, increased by multiple of 0.001 of the worst score.

ScoreRank Project rank calculated based on ScoreAvg.

ScoreRankAdj Project rank calculated based on ScoreAvgAdj.

RevID Reviewer's ID.

RevExp Reviewer's experience.

RevInst Reviewer's institution; "1" academia, "2" government.

RevGender Reviewer's gender; "1" females, "2" males.

RevRank Reviewer's rank; "3" full professor, "1" assistant professor.

RevDegree Reviewer's degree; "1" PhD, "2" MD, "3" PhD/MD.

RevCode Reviewer code ("A", "B", "C") in the original wide dataset.

Author(s)

Stephen Gallo
American Institute of Biological Sciences

References

- Gallo, S. (2021). Grant peer review scoring data with criteria scores. doi: [10.6084/m9.figshare.12728087](https://doi.org/10.6084/m9.figshare.12728087)
- Erosheva, E., Martinkova, P., & Lee, C. (2021a). When zero may not be zero: A cautionary note on the use of inter-rater reliability in evaluating grant peer review. *Journal of the Royal Statistical Society - Series A*. doi: [10.1111/rssa.12681](https://doi.org/10.1111/rssa.12681)
- Erosheva, E., Martinkova, P., & Lee, C. (2021b). Supplementary material: When zero may not be zero: A cautionary note on the use of inter-rater reliability in evaluating grant peer review. doi: [10.17605/OSF.IO/KNPH8](https://doi.org/10.17605/OSF.IO/KNPH8)

See Also

[ICCRestricted](#)

Description

BFI2 dataset (Hřebíčková et al., 2020) consists of responses of 1,733 respondents (1,003 females, 730 males) to Big Five Inventory 2 (BFI-2). It contains 60 ordinal items, vector of age, education, and vector of gender membership.

Usage

`data(BFI2)`

Format

BFI2 is a `data.frame` consisting of 1,733 observations on 64 variables.

i1–i60 The BFI items, scored on Likert scale where 1 = Disagree strongly, 2 = Disagree a little, 3 = Neutral; no opinion, 4 = Agree a little, and 5 = Agree strongly. Some items were recoded so that all items are scored in the same direction, see Details.

Gender Gender membership, 0 = females, 1 = males.

Age Age in years.

Educ Education, 1 = Basic school, 2 = Secondary technical school, 3 = Secondary general school, 4 = Other secondary school, 5 = Tertiary professional school, 6 = Bachelor degree, 7 = Masters degree, 8 = PhD

Details

The items prefixed with *i* are item scores. Items are indicators of 5 latent personality factors/dimensions/domains, which are further broken down into so-called facets. The 5 personality domains are: N = Negative Emotionality, E = Extraversion, O = Open-Mindedness, C = Consciousness and A = Agreeability. These are further broken down into so-called facets, as shown in the following table:

Domain	Facet	Item numbers
E	Sociability (scb)	1, 16, 31, 46
E	Assertiveness (asr)	6, 21, 36, 51
E	Energy Level (enl)	11, 26, 41, 56
A	Compassion (cmp)	2, 17, 32, 47
A	Respectfulness (rsp)	7, 22, 37, 52
A	Trust (trs)	12, 27, 42, 57
C	Organization (org)	3, 18, 33, 48
C	Productiveness (prd)	8, 23, 38, 53
C	Responsibility (rsp)	13, 28, 43, 58
N	Anxiety (anx)	4, 19, 34, 49
N	Depression (dep)	9, 24, 39, 54
N	Emotional Volatility (emt)	14, 29, 44, 59
O	Intellectual Curiosity (int)	10, 25, 40, 55

O	Aesthetic Sensitivity (aes)	5, 20, 35, 50
O	Creative Imagination (crt)	15, 30, 45, 60

In the original instrument, some items are inversely oriented, i.e., the higher score means the lower latent trait. This was the case of items number 3, 4, 5, 8, 9, 11, 12, 16, 17, 22, 23, 24, 25, 26, 28, 29, 30, 31, 36, 37, 42, 44, 45, 47, 48, 49, 50, 51, 55, and 58. These **items have been recoded** for you, i.e., displayed is value of 6 - original score.

In the sample code, alternative item names are provided. These item names can be used to decode the item domain, facet, item number, and whether it was recoded or not. For example, iCorg03r stands for recoded 3rd item (out of 60) from Consciousness domain and Organization facet.

Note

Thanks to Martina Hřebíčková for sharing this dataset.

References

Hřebíčková, M., Jelínek, M., Květon, P., Benkovič, A., Botek, M., Sudzina, F. Soto, Ch., John, O. (2020). Big Five Inventory 2 (BFI-2): Hierarchický model s 15 subškálami [Big Five Inventory 2 (BFI-2): Hierarchical model with 15 subscales, in Czech]. *Československá psychologie*, 64, 437–460.

Soto, C. J., & John, O. P. (2017). The next Big Five Inventory (BFI-2): Developing and assessing a hierarchical model with 15 facets to enhance bandwidth, fidelity, and predictive power. *Journal of Personality and Social Psychology*, 113, 117–143.

Examples

```
data(BFI2)
colNames(BFI2)[1:60] <- c("iEscb01", "iAcmp02", "iCorg03r", "iNanx04r", "iOaes05r", "iEasr06",
"iArsp07", "iCprd08r", "iNdep09r", "iOint10", "iEen11r", "iAtrs12r", "iCrsp13", "iNemt14",
"iOcrt15", "iEscb16r", "iAcmp17r", "iCorg18", "iNanx19", "iOaes20", "iEasr21", "iArsp22r",
"iCprd23r", "iNdep24r", "iOint25r", "iEen126r", "iAtrs27", "iCrsp28r", "iNemt29r",
"iOcrt30r", "iEscb31r", "iAcmp32", "iCorg33", "iNanx34", "iOaes35", "iEasr36r", "iArsp37r",
"iCprd38", "iNdep39", "iOint40", "iEen141", "iAtrs42r", "iCrsp43", "iNemt44r", "iOcrt45r",
"iEscb46", "iAcmp47r", "iCorg48r", "iNanx49r", "iOaes50r", "iEasr51r", "iArsp52", "iCprd53",
"iNdep54", "iOint55r", "iEen156", "iAtrs57", "iCrsp58r", "iNemt59", "iOcrt60")
```

CZmatura

CZmatura dataset

Description

The CZmatura dataset comes from a matura exam in mathematics. The exam was assigned to students from Grade 13, at the end of their secondary education.

Usage

```
data(CZmatura)
```

Format

CZmatura is a data.frame consisting of 15,702 observations on 75 variables.

SchType School type code.

FirstAtt First attempt; "1" yes, "0" no.

SchTypeGY School type gymnasium; "1" yes, "0" no.

o1 – o26.2 Item answers.

b1 – b26 Scored item answers.

Total Total score, calculated as sum of item scores (0 - 50).

IRTscore Score estimated from GPCM/2PL model.

IRTscoreSE SE of score estimated from GPCM/2PL model.

See Also

[CZmaturaS](#)

CZmaturaS

CZmatura dataset - sample

Description

The CZmaturaSdataset comes from a matura exam in mathematics. The exam was assigned to students from Grade 13, at the end of their secondary education. This is a random sample of 2,000 students from a total of 15,702.

Usage

```
data(CZmaturaS)
```

Format

CZmatura is a data.frame consisting of 2,000 observations on 75 variables.

SchType School type code.

FirstAtt First attempt; "1" yes, "0" no.

SchTypeGY School type gymnasium; "1" yes, "0" no.

o1 – o26.2 Item answers.

b1 – b26 Scored item answers.

Total Total score, calculated as sum of item scores (0 - 50).

IRTscore Score estimated from GPCM/2PL model.

IRTscoreSE SE of score estimated from GPCM/2PL model.

See Also[CZmatura](#)

`dataMedical`*Dichotomous dataset of admission test to medical school*

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.

X The first 100 columns represent dichotomously scored items of the test.

gender Variable describing gender; values "0" and "1" refer to males and females.

StudySuccess 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

Martin Vejrazka
First Faculty of Medicine, Charles University

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Stuka, C., Vejrazka, M., Martinkova, P., Komenda, M., & Stepanek, L. (2016). The use of test and item analysis for improvement of tests. Workshop held at conference MEFANET, 2016, Brno, Czech Republic.

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.

X The first 100 columns represent ordinal item scores of the test.

gender Variable describing gender; values "0" and "1" refer to males and females.

StudySuccess 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

Martin Vejrazka
First Faculty of Medicine, Charles University

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Stuka, C., Vejrazka, M., Martinkova, P., Komenda, M., & Stepanek, L. (2016). The use of test and item analysis for improvement of tests. Workshop held at conference MEFANET, 2016, Brno, Czech Republic.

See Also

[dataMedical](#), [dataMedicaltest](#), [dataMedicalkey](#)

dataMedicalkey *Key of correct answers for dataset of admission test to medical school*

Description

The dataMedicalkey is a vector of factors representing correct answers of dataMedicaltest dataset.

Usage

```
data(dataMedicalkey)
```

Format

A vector 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

Martin Vejrazka
First Faculty of Medicine, Charles University

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Stuka, C., Vejrazka, M., Martinkova, P., Komenda, M., & Stepanek, L. (2016). The use of test and item analysis for improvement of tests. Workshop held at conference MEFANET, 2016, Brno, Czech Republic.

See Also

[dataMedical](#), [dataMedicaltest](#), [dataMedicalgraded](#)

dataMedicaltest	<i>Dataset of admission test to medical school</i>
-----------------	--

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

A ‘dataMedicaltest’ is a ‘data.frame’ consisting of 2,392 observations on the following 102 variables.

X The first 100 columns represent items answers.

gender Variable describing gender; values “0” and “1” refer to males and females.

StudySuccess 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

Martin Vejrazka
First Faculty of Medicine, Charles University

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Stuka, C., Vejrazka, M., Martinkova, P., Komenda, M., & Stepanek, L. (2016). The use of test and item analysis for improvement of tests. Workshop held at conference MEFANET, 2016, Brno, Czech Republic.

See Also

[dataMedical](#), [dataMedicalkey](#), [dataMedicalgraded](#)

DDplot

*Plot difficulties and discriminations/item validity***Description**

Plots difficulty and (generalized) discrimination or criterion validity for items of the multi-item measurement test using the **ggplot2** package. Difficulty and discrimination/validity indices are plotted for each item, items are ordered by their difficulty.

Usage

```
DDplot(
  Data,
  item.names,
  discrim = "ULI",
  k = 3,
  l = 1,
  u = 3,
  maxscore,
  minscore,
  bin = FALSE,
  cutscore,
  average.score = FALSE,
  thr = 0.2,
  criterion = "none",
  val_type = "simple",
  data
)
```

Arguments

Data	numeric: binary or ordinal data matrix or data.frame which rows represent examinee answers (1 correct, 0 incorrect, or ordinal item scores) and columns correspond to the items.
item.names	character: the names of items. If not specified, the names of Data columns are used.
discrim	character: type of discrimination index to be calculated. Possible values are "ULI" (default), "RIT", "RIR", and "none". See Details .
k	numeric: number of groups to which data may be divided by the total score to estimate discrimination using discrim = "ULI". 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 scores of items. If single number is provided, the same maximal score is used for all items. If missing, vector of achieved maximal scores is calculated and used in calculations.

minscore	numeric: minimal scores of items. If single number is provided, the same maximal score is used for all items. If missing, vector of achieved maximal scores is calculated and used in calculations.
bin	logical: should the ordinal data be binarized? Default value is FALSE. In case that bin = TRUE, all values of Data equal or greater than cutscore are marked as 1 and all values lower than cutscore are marked as 0.
cutscore	numeric: cut-score used to binarize Data. If numeric, the same cut-score is used for all items. If missing, vector of maximal scores is used in calculations.
average.score	logical: should average score of the item be displayed instead of difficulty? Default value is FALSE. See Details .
thr	numeric: value of discrimination threshold. Default value is 0.2. With thr = NULL, no horizontal line is displayed in the plot.
criterion	numeric or logical vector: values of criterion. If supplied, discrim argument is ignored and item-criterion correlation (validity) is displayed instead. Default value is "none".
val_type	character: criterion validity measure. Possible values are "simple" (correlation between item score and validity criterion; default) and "index" (item validity index calculated as $\text{cor}(\text{item}, \text{criterion}) * \sqrt{((N-1) / N) * \text{var}(\text{item})}$), where N is number of respondents, see Allen & Yen, 1979, Ch. 6.4, for details). The argument is ignored if user does not supply any criterion.
data	deprecated. Use argument Data instead.

Details

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 = TRUE. Note that for binary data difficulty estimate is the same as average score of the item.

Note that all correlations are estimated using Pearson correlation coefficient.

Author(s)

Adela Hladka
 Institute of Computer Science of the Czech Academy of Sciences
 <hladka@cs.cas.cz>

Lubomir Stepanek
Charles University

Jana Vorlickova
Institute of Computer Science of the Czech Academy of Sciences

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

- Allen, M. J., & Yen, W. M. (1979). Introduction to measurement theory. Monterey, CA: Brooks/Cole.
- Martinkova, P., Stepanek, L., Drabinova, A., Houdek, J., Vejrazka, M., & Stuka, C. (2017). Semi-real-time analyses of item characteristics for medical school admission tests. In: Proceedings of the 2017 Federated Conference on Computer Science and Information Systems.

See Also

[discrim](#) for calculation of discrimination
[gDiscrim](#) for calculation of generalized ULI
[ggplot](#) for general function to plot a "ggplot" object

Examples

```
# loading 100-item medical admission test datasets
data(dataMedical, dataMedicalgraded)
# binary dataset
dataBin <- dataMedical[, 1:100]
# ordinal dataset
dataOrd <- dataMedicalgraded[, 1:100]

# DDplot of binary dataset
DDplot(dataBin)
## Not run:
# DDplot of binary dataset without threshold
DDplot(dataBin, thr = NULL)
# compared to DDplot using ordinal dataset and 'bin = TRUE'
DDplot(dataOrd, bin = TRUE)
# compared to binarized dataset using bin = TRUE and cut-score 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
# threshold lowered to 0.1
DDplot(dataBin, k = 5, l = 4, u = 5, thr = 0.1)

# DDplot of ordinal dataset using ULI
DDplot(dataOrd)
```



```

# DDplot of ordinal dataset using generalized ULI
# discrimination based on 5 groups, comparing 4th and 5th
# threshold lowered to 0.1
DDplot(dataOrd, k = 5, l = 4, u = 5, thr = 0.1)
# DDplot of ordinal dataset using RIT
DDplot(dataOrd, discrim = "RIT")
# DDplot of ordinal dataset using RIR
DDplot(dataOrd, discrim = "RIR")
# DDplot of ordinal dataset displaying only difficulty
DDplot(dataBin, discrim = "none")

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

# item difficulty / criterion validity plot for data with criterion
data(GMAT, package = "difNLR")
DDplot(GMAT[, 1:20], criterion = GMAT$criterion, val_type = "simple")

## End(Not run)

```

DistractorAnalysis *Distractor analysis*

Description

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

Usage

```

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

```

Arguments

Data character: data matrix or data.frame with rows representing unscored item responses from a multiple-choice test and columns corresponding to the items.

key	character: answer key for the items. The key must be a vector of the same length as <code>ncol(Data)</code> . In case it is not provided, criterion needs to be specified.
p.table	logical: should the function return the proportions? If FALSE (default), the counts are returned.
num.groups	numeric: number of groups to which are the respondents split.
criterion	numeric: numeric vector. If not provided, total score is calculated and distractor analysis is performed based on it.
crit.discrete	logical: is criterion discrete? Default value is FALSE. See details.
cut.points	numeric: numeric vector specifying cut points of criterion. See details.
data	deprecated. Use argument <code>Data</code> instead.
matching	deprecated. Use argument <code>criterion</code> instead.
match.discrete	deprecated. Use argument <code>crit.discrete</code> instead.

Details

This function is an adapted version of the `distractor.analysis` function from **CTT** package. In case that no `criterion` is provided, the scores are calculated using the item `Data` and `key`. The respondents are by default split 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 `criterion` is discrete (`crit.discrete = TRUE`), `criterion` is split based on its unique levels. Other cut points can be specified via `cut.points` argument.

Author(s)

Adela Hladka
 Institute of Computer Science of the Czech Academy of Sciences
 <hladka@cs.cas.cz>

Patricia Martinkova
 Institute of Computer Science of the Czech Academy of Sciences
 <martinkova@cs.cas.cz>

Examples

```
# loading 100-item medical admission test dataset
data(dataMedicaltest, dataMedicalkey)
data <- dataMedicaltest[, 1:100]
dataBin <- dataMedical[, 1:100]
key <- unlist(dataMedicalkey)

# distractor analysis for dataMedicaltest dataset
DistractorAnalysis(data, key)
## Not run:
# distractor analysis for dataMedicaltest dataset with proportions
DistractorAnalysis(data, key, p.table = TRUE)
```

```

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

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

# distractor analysis for dataMedicaltest using discrete criterion
DistractorAnalysis(data, key, criterion = criterion, crit.discrete = TRUE)

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

## End(Not run)

```

fa_parallel

Conduct Parallel Analysis

Description

Computes the eigenvalues of the sample correlation matrix and the eigenvalues obtained from a random correlation matrix for which no factors/components are assumed. By default, the function utilizes a modified Horn's (1965) method, which – instead of mean – uses 95th percentile of each item eigenvalues sampling distribution as a threshold to find the optimal number of factors/components.

Usage

```

fa_parallel(
  Data,
  cor = "pearson",
  n_obs = NULL,
  method = "pca",
  threshold = "quantile",
  p = 0.95,
  n_iter = 20,
  plot = TRUE,
  show_kaiser = TRUE,
  fm = "minres",
  use = "pairwise",
  ...
)

```

Arguments

Data *data.frame* or *matrix*, dataset (where rows are observations and columns items) or correlation matrix (recognized automatically).

cor	<i>character</i> , how to calculate the correlation matrix of the real data. Can be either pearson (default), tetrachoric or polychoric. Unambiguous abbreviations accepted.
n_obs	<i>integer</i> , in case you provided the correlation matrix directly as the input, you have to provide the number of observations in the original dataset.
method	<i>character</i> , either fa, pca, or both (the default). Which method to use for the eigenvalues simulation and computation.
threshold	<i>character</i> , whether to use traditional Horn's method or more recent and well-performing quantile one. Either mean or quantile (default). Can be abbreviated.
p	<i>numeric</i> (0–1), probability for which the sample quantile is produced. Defaults to .95. Ignored if threshold = "mean".
n_iter	<i>integer</i> , number of iterations, i.e. the number of zero-factor multivariate normal distributions to sample. Defaults to 20.
plot	<i>logical</i> , if TRUE (the default), show the plot along with the function results. To create the plot from the resulting object afterwards, call plot().
show_kaiser	<i>logical</i> , whether to show Kaiser boundary in the plot (the default) or not.
fm	<i>character</i> , factoring method. See fa from the package psych .
use	an optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings "everything", "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs".
...	Arguments passed on to <code>psych::polychoric</code>
	correct Correction value to use to correct for continuity in the case of zero entry cell for tetrachoric, polychoric, polybi, and mixed.cor. See the examples for the effect of correcting versus not correcting for continuity.
	smooth if TRUE and if the tetrachoric/polychoric matrix is not positive definite, then apply a simple smoothing algorithm using cor.smooth
	global When finding pairwise correlations, should we use the global values of the tau parameter (which is somewhat faster), or the local values (global=FALSE)? The local option is equivalent to the polycor solution, or to doing one correlation at a time. global=TRUE borrows information for one item pair from the other pairs using those item's frequencies. This will make a difference in the presence of lots of missing data. With very small sample sizes with global=FALSE and correct=TRUE, the function will fail (for as yet undetermined reasons).
	weight A vector of length of the number of observations that specifies the weights to apply to each case. The NULL case is equivalent of weights of 1 for all cases.
	progress Show the progress bar (if not doing multicores)
	ML ML=FALSE do a quick two step procedure, ML=TRUE, do longer maximum likelihood — very slow! Deprecated
	delete Cases with no variance are deleted with a warning before proceeding.
	max.cat The maximum number of categories to bother with for polychoric.

Details

Horn proposed a solution to the problem of optimal factor number identification using an approach based on a Monte Carlo simulation.

First, several (20 by default) zero-factor p -variate normal distributions (where p is the number of columns) are obtained, and $p \times p$ correlation matrices are computed for them. Eigenvalues of each matrix is then calculated in order to get an eigenvalues sampling distribution for each simulated variable.

Traditionally, Horn obtains an average of each sampling distribution and these averages are used as a threshold which is compared with eigenvalues of the original, real data. However, *usage of the mean was later disputed* by Buja & Eyuboglu (1992), and 95th percentile of eigenvalues sampling distribution was suggested as a more accurate threshold. This, more recent method is used by default in the function.

Value

An object of class `data.frame` and `sia_parallel`. Can be plotted using `plot()`.

Author(s)

Jan Netik
Institute of Computer Science of the Czech Academy of Sciences
Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

- Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30, 179–185. doi: [10.1007/BF02289447](https://doi.org/10.1007/BF02289447)
- Buja, A., & Eyuboglu, N. (1992). Remarks on parallel analysis. *Multivariate Behavioral Research*, 27, 509–540. doi: [10.1207/s15327906mbr2704_2](https://doi.org/10.1207/s15327906mbr2704_2)

Examples

```
data("TestAnxietyCor", package = "ShinyItemAnalysis")
fa_parallel(TestAnxietyCor, n_obs = 335, method = "pca")

## Not run:
data("bfi", package = "psych")
items <- bfi[, 1:25]

fa_parallel(items)
fa_parallel(items, threshold = "mean") # traditional Horn's method

## End(Not run)
```

gDiscrim

*Compute 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 datasets can be specified by user.

Usage

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

Arguments

Data	matrix or data.frame of items to be examined. Rows represent respondents, columns represent items.
k	numeric: number of groups to which may be Data 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 .
x	deprecated. Use argument Data instead.

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 [DDplot](#) function.

Author(s)

Adela Hladka
 Institute of Computer Science of the Czech Academy of Sciences
 <hladka@cs.cas.cz>

Lubomir Stepanek
 Charles University

Jana Vorlickova
 Institute of Computer Science of the Czech Academy of Sciences

Patricia Martinkova
 Institute of Computer Science of the Czech Academy of Sciences
 <martinkova@cs.cas.cz>

References

Martinkova, P., Stepanek, L., Drabinova, A., Houdek, J., Vejrazka, M., & Stuka, C. (2017). Semi-real-time analyses of item characteristics for medical school admission tests. In: Proceedings of the 2017 Federated Conference on Computer Science and Information Systems. <https://doi.org/10.15439/2017F380>

See Also

[DDplot](#)

Examples

```
# loading 100-item medical admission test datasets
data(dataMedical, dataMedicalgraded)
# binary dataset
dataBin <- dataMedical[, 1:100]
# ordinal dataset
dataOrd <- dataMedicalgraded[, 1:100]

# ULI for the first 5 items of binary dataset
# compare to psychometric::discrim(dataBin)
gDiscrim(dataBin)[1:5]
# generalized ULI using 5 groups, compare 4th and 5th for binary dataset
gDiscrim(dataBin, k = 5, l = 4, u = 5)[1:5]

# ULI for first 5 items for ordinal dataset
gDiscrim(dataOrd)[1:5]
# generalized ULI using 5 groups, compare 4th and 5th for binary dataset
gDiscrim(dataOrd, k = 5, l = 4, u = 5)[1:5]
# maximum (4) and minimum (0) score are same for all items
gDiscrim(dataOrd, k = 5, l = 4, u = 5, maxscore = 4, minscore = 0)[1:5]
```

`ggWrightMap`*Plot Wright map using ggplot2*

Description

This function allows to generate Wright map (also called item-person map) using `ggplot()` function from the **ggplot2** package. 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,  
  ylab.theta = "Respondent latent trait",  
  ylab.b = "Item difficulty",  
  rel_widths = c(1, 1)  
)
```

Arguments

<code>theta</code>	numeric: vector of ability estimates.
<code>b</code>	numeric: vector of difficulty estimates.
<code>binwidth</code>	numeric: the width of the bins of histogram.
<code>color</code>	character: color of histogram.
<code>size</code>	text size in pts.
<code>item.names</code>	names of items to be displayed.
<code>ylab.theta</code>	character: description of y-axis for the histogram.
<code>ylab.b</code>	character: description of y-axis for the plot of difficulty estimates.
<code>rel_widths</code>	numeric: vector of length 2 specifying ratio of "facet's" widths.

Author(s)

Adela Hladka
Institute of Computer Science of the Czech Academy of Sciences
<hladka@cs.cas.cz>

Jan Netik
Institute of Computer Science of the Czech Academy of Sciences

Patricia Martinkova
 Institute of Computer Science of the Czech Academy of Sciences
 <martinkova@cs.cas.cz>

References

Wright, B. & Stone, M. (1979). Best test design. MESA Press: Chicago, IL

Examples

```
library(mirt)

data(HCI)

# fit Rasch model with the mirt package
fit <- mirt(HCI[, 1:20], model = 1, itemtype = "Rasch")
# factor scores
theta <- as.vector(fscores(fit))
# difficulty estimates using IRT parametrization
b <- coef(fit, simplify = TRUE, IRTpars = TRUE)$items[, "b"]

# Wright map
ggWrightMap(theta, b)

# Wright map with modified item names
item.names <- paste("Item", 1:20)
ggWrightMap(theta, b, item.names = item.names)

# Wright map with modified descriptions of y-axis and relative widths of plots
ggWrightMap(theta, b,
  ylab.theta = "Latent trait", ylab.b = "Difficulty estimates",
  rel_widths = c(2, 1)
)
```

HCI

Homeostasis concept inventory dichotomous dataset

Description

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

Usage

```
data(HCI)
```

Format

HCI is a data.frame consisting of 651 observations on the 22 variables.

Item1-Item20 Dichotomously scored items of the HCI test.

gender Gender membership, "0" males, "1" females.

major Identifier whether student plans to major in the life sciences.

Author(s)

Jenny L. McFarland
Biology Department, Edmonds Community College

References

McFarland, J. L., Price, R. M., Wenderoth, M. P., Martinkova, P., Cliff, W., Michael, J., ... & Wright, A. (2017). Development and validation of the homeostasis concept inventory. *CBE-Life Sciences Education*, 16(2), ar35. doi: [10.1187/cbe.16100305](https://doi.org/10.1187/cbe.16100305)

See Also

[HCItest](#) for HCI multiple-choice dataset
[HCIkey](#) for key of correct answers for HCI
[HCIdata](#) for HCI full dataset
[HCIgrads](#) for HCI dataset of graduate students
[HCIprepost](#) for HCI pretest and posttest scores
[HCItestretest](#) for HCI test-retest dataset

HCIdata

Homeostasis concept inventory full dataset

Description

HCIdata dataset consists of the responses of 669 students (405 males, 246 females, 18 without gender specification) to Homeostasis Concept Inventory (HCI) multiple-choice test. It contains answers to 20 multiple-choice items, scored items, total score, gender membership, identifier whether students plan to major in science, study year, minority membership, identifier whether English is the student's first language, and type of school.

Usage

data(HCIdata)

Format

HCIdata is a data.frame consisting of 669 observations on the 47 variables.

A1-A20 Multiple-choice items of the HCI test.

QR1-QR20 Scored items of the HCI test, "0" incorrect, "1" correct.

total Total test score.

gender Gender membership, "M" males, "F" females, "none" undisclosed.

major Identifier whether students plans to major in the life sciences.

years5 Study year.

minority Minority membership, "maj" majority, "min" minority, "none" undisclosed.

EnglishF Identifier whether English is the student's first language.

types Course type, "allied" allied health, "majors" physiology courses for science majors, "mixed majors" courses for non-majors.

typesCH Type of school, "AC" associate's college, "BCAS" baccalaureate college: arts and sciences focus, "R1" research university, "MCU" master's college and university.

Author(s)

Jenny L. McFarland
Biology Department, Edmonds Community College

References

McFarland, J. L., Price, R. M., Wenderoth, M. P., Martinkova, P., Cliff, W., Michael, J., ... & Wright, A. (2017). Development and validation of the homeostasis concept inventory. *CBE-Life Sciences Education*, 16(2), ar35. doi: [10.1187/cbe.16100305](https://doi.org/10.1187/cbe.16100305)

See Also

[HCI](#) for HCI dichotomous dataset
[HCItest](#) for HCI multiple-choice dataset
[HCIkey](#) for key of correct answers for HCI
[HCIgrads](#) for HCI dataset of graduate students
[HCIprepost](#) for HCI pretest and posttest scores
[HCItestretest](#) for HCI test-retest dataset

HCIgrads

Homeostasis concept inventory dataset of graduate students

Description

HCIgrads dataset consists of the responses of 10 graduate students to Homeostasis Concept Inventory (HCI) multiple-choice test. It contains answers to 20 multiple-choice items, scored items, and total test score.

Usage

```
data(HCIgrads)
```

Format

HCIgrads is a `data.frame` consisting of 10 observations on the 42 variables.

A1-A20 Multiple-choice items of the HCI test.

QR1-QR20 Scored items of the HCI test, "0" incorrect, "1" correct.

total Total test score.

Author(s)

Jenny L. McFarland
Biology Department, Edmonds Community College

References

McFarland, J. L., Price, R. M., Wenderoth, M. P., Martinkova, P., Cliff, W., Michael, J., ... & Wright, A. (2017). Development and validation of the homeostasis concept inventory. *CBE-Life Sciences Education*, 16(2), ar35. doi: [10.1187/cbe.16100305](https://doi.org/10.1187/cbe.16100305)

See Also

[HCIdata](#) for HCI full dataset
[HCI](#) for HCI dichotomous dataset
[HCItest](#) for HCI multiple-choice dataset
[HCIkey](#) for key of correct answers for HCI
[HCIprepost](#) for HCI pretest and posttest scores
[HCItestretest](#) for HCI test-retest dataset

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 nominal vector 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

McFarland, J. L., Price, R. M., Wenderoth, M. P., Martinkova, P., Cliff, W., Michael, J., ... & Wright, A. (2017). Development and validation of the homeostasis concept inventory. *CBE-Life Sciences Education*, 16(2), ar35. doi: [10.1187/cbe.16100305](https://doi.org/10.1187/cbe.16100305)

See Also

[HCItest](#) for HCI multiple-choice dataset
[HCI](#) for HCI dichotomous dataset
[HCIdata](#) for HCI full dataset
[HCIgrads](#) for HCI dataset of graduate students
[HCIprepost](#) for HCI pretest and posttest scores
[HCItestretest](#) for HCI test-retest dataset

HCIprepost

Homeostasis concept inventory pretest and posttest scores

Description

HCIprepost dataset consists of the pretest and posttest score of 16 students to Homeostasis Concept Inventory (HCI). Between the pre-test and post-test, the students received instruction on homeostasis within a physiology course.

Usage

```
data(HCIprepost)
```

Format

HCIprepost is a data.frame consisting of 16 observations on the 2 variables.

id Anonymized respondent ID.

score.pre Pretest score.

score.post Posttest score.

Author(s)

Jenny L. McFarland
Biology Department, Edmonds Community College

References

McFarland, J. L., Price, R. M., Wenderoth, M. P., Martinkova, P., Cliff, W., Michael, J., ... & Wright, A. (2017). Development and validation of the homeostasis concept inventory. *CBE-Life Sciences Education*, 16(2), ar35. doi: [10.1187/cbe.16100305](https://doi.org/10.1187/cbe.16100305)

See Also

[HCIdata](#) for HCI full dataset

[HCI](#) for HCI dichotomous dataset

[HCItest](#) for HCI multiple-choice dataset

[HCIkey](#) for key of correct answers for HCI

[HCIgrads](#) for HCI dataset of graduate students

[HCItestretest](#) for HCI test-retest dataset

HCItest

Homeostasis concept inventory multiple-choice dataset

Description

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

Usage

```
data(HCItest)
```

Format

HCItest is a data.frame consisting of 651 observations on the 22 variables.

Item1-Item20 Multiple-choice items of the HCI test.

gender Gender membership, "0" males, "1" females.

major Identifier whether student plans to major in the life sciences.

Author(s)

Jenny L. McFarland
Biology Department, Edmonds Community College

References

McFarland, J. L., Price, R. M., Wenderoth, M. P., Martinkova, P., Cliff, W., Michael, J., ... & Wright, A. (2017). Development and validation of the homeostasis concept inventory. *CBE-Life Sciences Education*, 16(2), ar35. doi: [10.1187/cbe.16100305](https://doi.org/10.1187/cbe.16100305)

See Also

[HCIkey](#) for key of correct answers for HCI
[HCI](#) for HCI dichotomous dataset
[HCIdata](#) for HCI full dataset
[HCIgrads](#) for HCI dataset of graduate students
[HCIprepost](#) for HCI pretest and posttest scores
[HCItestretest](#) for HCI test-retest dataset

HCItestretest

Homeostasis concept inventory test-retest dataset

Description

HCItestretest) dataset consists of the responses of 45 students to Homeostasis Concept Inventory (HCI). It contains answers to 20 multiple-choice items, scored items, identifier of test/retest, total score, gender membership and identifier whether students plan to major in life sciences. The data are organized so that each pair of subsequent rows belongs to one student. Students took no courses on homeostasis between the test and retest.

Usage

```
data(HCItestretest)
```

Format

HCItestretest is a data.frame consisting of 90 observations on the 44 variables.

A1-A20 Multiple-choice items of the HCI test.

QR1-QR20 Scored items of the HCI test, "0" incorrect, "1" correct.

test Identifier of test vs retest, "test" test, "retest" retest after.

total Total test score.

gender Gender membership, "M" male, "F" female.

major Identifier whether student plans to major in the life sciences.

Author(s)

Jenny L. McFarland
Biology Department, Edmonds Community College

References

McFarland, J. L., Price, R. M., Wenderoth, M. P., Martinkova, P., Cliff, W., Michael, J., ... & Wright, A. (2017). Development and validation of the homeostasis concept inventory. *CBE-Life Sciences Education*, 16(2), ar35. doi: [10.1187/cbe.16100305](https://doi.org/10.1187/cbe.16100305)

See Also

[HCIdata](#) for HCI full dataset
[HCI](#) for HCI dichotomous dataset
[HCItest](#) for HCI multiple-choice dataset
[HCIkey](#) for key of correct answers for HCI
[HCIgrads](#) for HCI dataset of graduate students
[HCIprepost](#) for HCI pretest and posttest scores

 ICCRestricted

Range-restricted reliability with intra-class correlation

Description

Function estimating reliability with intra-class correlation for the complete or for the range-restricted sample.

Usage

```

ICCRestricted(
  Data,
  case,
  var,
  rank = NULL,
  dir = "top",
  sel = 1,
  nsim = 100,
  ci = 0.95,
  seed = NULL
)
  
```

Arguments

Data	matrix or data.frame which includes variables describing ID of ratees (specified in case), ratings (specified in var), and (optionally) rank of ratees (specified in rank).
case	character: name of the variable in Data with ID of the ratee (subject or object being evaluated, such as a respondent, proposal, patient, applicant etc.)
var	character: name of the variable in Data with the ratings/scores.
rank	numeric: vector of ranks of ratees. If not provided, rank of ratee is calculated based on average rating based on var variable.
dir	character: direction of range-restriction, available options are "top" (default) or "bottom". Can be an unambiguous abbreviation (i.e., "t" or "b").
sel	numeric: selected number (given > 1) or percentage (given <= 1) of ratees. Default value is 1 (complete dataset).

nsim	numeric: number of simulations for bootstrap confidence interval. Default value is 100.
ci	numeric: confidence interval. Default value is 0.95.
seed	seed for simulations. Default value is NULL, random seed. See lme4::bootMer for more detail.

Value

A data.frame with the following columns:

n_sel	number of ratees selected/subsetted.
prop_sel	proportion of ratees selected.
dir	direction of range-restriction. NA if range is effectively not restricted (100 ratee, "true variance", between-group variance).
VarResid	residual variance.
VarTotal	total variance.
ICC1	single-rater inter-rater reliability.
ICC1_LCI	lower bound of the confidence interval for ICC1.
ICC1_UCI	upper bound of the confidence interval for ICC1.
ICC3	multiple-rater inter-rater reliability.
ICC3_LCI	lower bound of the confidence interval for ICC3.
ICC3_UCI	upper bound of the confidence interval for ICC3.

Author(s)

Patricia Martinkova
 Institute of Computer Science of the Czech Academy of Sciences
 <martinkova@cs.cas.cz>

Jan Netik
 Institute of Computer Science of the Czech Academy of Sciences

References

- Erosheva, E., Martinkova, P., & Lee, C. (2021a). When zero may not be zero: A cautionary note on the use of inter-rater reliability in evaluating grant peer review. *Journal of the Royal Statistical Society - Series A*. Accepted.
- Erosheva, E., Martinkova, P., & Lee, C. (2021b). Supplementary material for When zero may not be zero: A cautionary note on the use of inter-rater reliability in evaluating grant peer review.

Examples

```
# loading AIBS dataset
data(AIBS, package = "ShinyItemAnalysis")

# ICC for the whole sample
ICCrestricted(Data = AIBS, case = "ID", var = "Score", rank = "ScoreRankAdj")

# ICC for the range-restricted sample considering 80% of top rates
ICCrestricted(Data = AIBS, case = "ID", var = "Score", rank = "ScoreRankAdj",
              sel = 0.8)
```

ItemAnalysis

Compute traditional item analysis indices

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

```
ItemAnalysis(
  Data,
  criterion = "none",
  k = 3,
  l = 1,
  u = 3,
  maxscore = NULL,
  minscore = NULL,
  cutscore = NULL,
  bin = FALSE,
  data,
  y,
  add.bin
)
```

Arguments

Data	matrix or data.frame of items to be examined. Rows represent respondents, columns represent items.
criterion	vector of criterion values.
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 or vector: maximal score in ordinal items. If missing, vector of obtained maximal scores is imputed. See Details .
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 .
bin	logical: If TRUE, indices are printed also for binarized data. See Details .
data	deprecated. Use argument Data instead.
y	deprecated. Use argument criterion instead.
add.bin	deprecated. Use argument bin instead.

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 `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.
Mean	average item score.
SD	standard deviation of the item score.
SD.bin	standard deviation of the item score for binarized data.
Prop.max.score	proportion of maximal scores.
Min.score	minimal score specified in <code>minscore</code> ; if not provided, observed minimal score.
Max.score	maximal score specified in <code>maxscore</code> ; if not provided, observed maximal score.
obs.min	observed minimal score.
obs.max	observed maximal score.
Cut.Score	cut-score specified in <code>cutscore</code> .
gULI	generalized ULI.
gULI.bin	generalized ULI for binarized data.
ULI	discrimination with ULI using the usual parameters (3 groups, comparing 1st and 3rd).

ULI.bin	discrimination with ULI using the usual parameters for binarized data (3 groups, comparing 1st and 3rd).
RIT	item-total correlation (correlation between item score and overall test score).
RIT.bin	item-total correlation for binarized data.
RIR	item-rest correlation (correlation between item score and overall test score without the given item).
RIR.bin	item-rest correlation for binarized data.
Corr.criterion	correlation between item score and criterion criterion.
Corr.criterion.bin	correlation between item score and criterion criterion for binarized data.
Index.val	item validity index calculated as $\text{cor}(\text{item}, \text{criterion}) * \sqrt{((N-1) / N) * \text{var}(\text{item})}$, see Allen and Yen (1979, Ch.6.4).
Index.val.bin	item validity index for binarized data.
Index.rel	item reliability index calculated as $\text{cor}(\text{item}, \text{test}) * \sqrt{((N-1) / N) * \text{var}(\text{item})}$, see Allen and Yen (1979, Ch.6.4).
Index.rel.bin	item reliability index for binarized data.
Index.rel.drop	item reliability index 'drop' (scored without item).
Index.rel.drop.bin	item reliability index 'drop' (scored without item) for binarized data.
Alpha.drop	Cronbach's alpha without given item. In case of two-item dataset, NAs are returned.
Alpha.drop.bin	Cronbach's alpha without given item, for binarized data. In case of two-item dataset, NAs are returned.
Perc.miss	Percentage of missed responses on the particular item.
Perc.nr	Percentage of respondents that did not reached the item nor the subsequent ones, see recode_nr function for further details.

With `bin = TRUE`, indices based on binarized dataset are also provided and marked with `bin` suffix.

Author(s)

Patricia Martinkova
 Institute of Computer Science of the Czech Academy of Sciences
 <martinkova@cs.cas.cz>

Jan Netik
 Institute of Computer Science of the Czech Academy of Sciences
 Charles University

Jana Vorlickova
 Institute of Computer Science of the Czech Academy of Sciences

Adela Hladka
 Institute of Computer Science of the Czech Academy of Sciences
 <hladka@cs.cas.cz>

References

- Martinkova, P., Stepanek, L., Drabinova, A., Houdek, J., Vejrazka, M., & Stuka, C. (2017). Semi-real-time analyses of item characteristics for medical school admission tests. In: Proceedings of the 2017 Federated Conference on Computer Science and Information Systems. <https://doi.org/10.15439/2017F380>
- Allen, M. J. & Yen, W. M. (1979). Introduction to measurement theory. Monterey, CA: Brooks/Cole.

See Also

[DDplot](#), [gDiscrim](#), [recode_nr](#)

Examples

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

# item analysis for binary data
head(ItemAnalysis(dataBin))
# item analysis for binary data using also study success
head(ItemAnalysis(dataBin, criterion = StudySuccess))

# item analysis for binary data
head(ItemAnalysis(dataOrd))
# item analysis for binary data using also study success
head(ItemAnalysis(dataOrd, criterion = StudySuccess))
# including also item analysis for binarized data
head(ItemAnalysis(dataOrd,
  criterion = StudySuccess, k = 5, l = 4, u = 5,
  maxscore = 4, minscore = 0, cutscore = 4, bin = TRUE
))

## End(Not run)
```

Description

LearningToLearn is a real longitudinal dataset used in Martinkova et al (2020) study, demonstrating differential item functioning in change (DIF-C) on Learning to Learn (LtL) test. Among other variables, it primarily contains binary-coded responses of 782 subjects to (mostly) multiple-choice test consisting of 41 items within 7 subscales (see **Format** for details). Each respondent was tested

twice in total – the first time in Grade 6 and the second time in Grade 9. Most importantly, school track (variable `track_01` or `track`) is available, with 391 students attending basic school (BS) and 391 pursuing selective academic school (AS). This dataset was created using propensity score matching algorithm to achieve similar characteristics in both tracks (see **References** for details). To further simplify the work with L_{tL} dataset, we provide computed total scores as well as 7 subscores, both for Grade 6 and Grade 9. The dataset also includes *change* variables for each item (see **Format** for details) for more detailed DIF-C analysis using multinomial regression model.

Usage

```
data(LearningToLearn)
```

Format

A LearningToLearn data frame consists of 782 observations on the following 141 variables:

track_01 Dichotomously scored school track, where "1" denotes the selective academic school one.

track School track, where "AS" represents the selective academic school track, and "BS" stands for basic school track.

score_6 & score_9 Total test score value obtained by summing all 41 items of L_{tL}, the number denotes the Grade which the respondent was taking at the time of testing.

score_6_subtest1–score_6_subtest7 Scores of respective cognitive subtest (1–7) of L_{tL} in Grade 6.

score_9_subtest1–score_9_subtest7 Scores of respective cognitive subtest (1–7) of L_{tL} in Grade 9.

Item1A_6–Item7F_6 Dichotomously coded 41 individual items obtained at Grade 6, "1" represents the correct answer to the particular item.

Item1A_9–Item7F_9 Dichotomously coded 41 individual items obtained at Grade 9, "1" represents the correct answer to the particular item.

Item1A_changes–Item7F_changes Change patterns with those possible values:

- a student responded correctly in neither Grade 6 nor in Grade 9 (did not improve, "00")
- a student responded correctly in Grade 6 but not in Grade 9 (deteriorated, "10")
- a student did not respond correctly in Grade 6 but responded correctly in Grade 9 (improved, "01"), and
- a student responded correctly in both grades (did not deteriorate, "11")

Author(s)

Patricia Martinkova
 Faculty of Education, Charles University
 Institute of Computer Science of the Czech Academy of Sciences
 <martinkova@cs.cas.cz>

Adela Hladka
 Institute of Computer Science of the Czech Academy of Sciences

Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Eva Potuznikova
Faculty of Education, Charles University

References

Martinkova, P., Hladka, A., & Potuznikova, E. (2020). Is academic tracking related to gains in learning competence? Using propensity score matching and differential item change functioning analysis for better understanding of tracking implications. *Learning and Instruction, 66*, 101286. doi: [10.1016/j.learninstruc.2019.101286](https://doi.org/10.1016/j.learninstruc.2019.101286)

MSclinical

Clinical outcomes in multiple sclerosis patients dataset

Description

The MSclinical dataset contains clinical measures on multiple sclerosis patients.

Usage

```
data(MScclinical)
```

Format

MSclinical is a data.frame consisting of 17 observations on 13 variables.

LCLA Low-Contrast Letter Acuity test.

MI Motricity Index.

MAS Modified Ashworth Scale.

BBS Berg Balance Scale.

T Tremor.

DD Dysdiadochokinesia.

DM Dysmetria.

PRs Postural reactions.

KH Knee Hyperextension.

NHPT Nine-Hole Peg Test.

T25FW Timed 25-Foot Walk.

PASAT3 3-minute version of the Paced Auditory Serial Addition Test.

EDSS Kurtzke Expanded Disability Status Scale.

References

Rasova, K., Martinkova, P., Vyskotova, J., & Sedova, M. (2012). Assessment set for evaluation of clinical outcomes in multiple sclerosis: Psychometric properties. *Patient related outcome measures*, 3, 59. doi: [10.2147/PROM.S32241](https://doi.org/10.2147/PROM.S32241)

 NIH

NIH grant peer review scoring dataset

Description

The NIH dataset (Erosheva et al., 2020a) was sampled from a full set of 54,740 R01 applications submitted by black and white principal investigators (PIs) and reviewed by NIH's Center for Scientific Review (CSR) during council years 2014–2016.

It contains the original random sample of white applicants as generated by Erosheva et al. (2020b) and a sample of 46 black applicants generated to obtain the same ratio of white and black applicants as in the original sample (for details, see Erosheva et al., 2021a). The dataset was used by Erosheva et al. (2021b) to demonstrate issues of inter-rater reliability in case of restricted samples.

The available variables include preliminary criterion scores on Significance, Investigator, Innovation, Approach, Environment and a preliminary Overall Impact Score. Each of these criteria and the overall score is scored on an integer scale from 1 (best) to 9 (worst). Besides the preliminary criteria and Overall Impact Scores, the data include applicant race, the structural covariates (PI ID, application ID, reviewer ID, administering institute, IRG, and SRG), the matching variables – gender, ethnicity (Hispanic/Latino or not), career stage, type of academic degree, institution prestige (as reflected by the NIH funding bin), area of science (as reflected by the IRG handling the application), application type (new or renewal) and status (amended or not) – as well as the final overall score. In addition, the file includes a study group ID variable that refers to the Matched and Random subsets used in the original study.

Usage

data(NIH)

Format

NIH is a data.frame consisting of 5802 observations on 27 variables.

ID Proposal ID.

Score Preliminary Overall Impact score (1-9 integer scale, 1 best).

Significance, Investigator, Innovation, Approach, Environment Preliminary Criterion Scores (1-9 integer scale, 1 best).

PIRace Principal investigator's self-identified race; "White" or "Black".

PIID Anonymized ID of principal investigator (PI).

PIGender PI's gender membership; "Male" or "Female".

PIEthn PI's ethnicity; "Hispanic/Latino" or "Non-Hispanic".

PICareerStage PI's career stage; "ESI" Early Stage Investigator, "Experienced" Experienced Investigator, or "Non-ES NI" Non-Early Stage New Investigator.

PIDegree PI's degree; "PhD", "MD", "MD/PhD", or "Others".

PIInst Lead PI's institution's FY 2014 total institution NIH funding; 5 bins with 1 being most-funded.

GroupID Group ID.

RevID Reviewer's ID.

IRG IRG (Integrated Research Group) id.

AdminOrg Administering Organization id.

SRG SRG (Scientific Research Group) id.

PropType Application type, "New" or "Renewal".

Ammend Ammend. Logical.

ScoreAvg Average of the three overall scores from different reviewers.

ScoreAvgAdj Average of the three overall scores from different reviewers, increased by multiple of 0.001 of the worst score.

ScoreRank Project rank calculated based on ScoreAvg.

ScoreRankAdj Project rank calculated based on ScoreAvgAdj.

ScoreFinalChar Final Overall Impact score (1-9 integer scale, 1 best; "ND" refers to "not discussed")

ScoreFinal Final Overall Impact score (1-9 integer scale, 1 best).

References

Erosheva, E. A., Grant, S., Chen, M.-C., Lindner, M. D., Nakamura, R. K., & Lee, C. J. (2020a). NIH peer review: Criterion scores completely account for racial disparities in overall impact scores. *Science Advances* 6(23), eaaz4868, doi: [10.1126/sciadv.aaz4868](https://doi.org/10.1126/sciadv.aaz4868)

Erosheva, E. A., Grant, S., Chen, M.-C., Lindner, M. D., Nakamura, R. K., & Lee, C. J. (2020b). Supplementary material: NIH peer review: Criterion scores completely account for racial disparities in overall impact scores. *Science Advances* 6(23), eaaz4868, doi: [10.17605/OSF.IO/4D6RX](https://doi.org/10.17605/OSF.IO/4D6RX)

Erosheva, E., Martinkova, P., & Lee, C. J. (2021a). Supplementary material: When zero may not be zero: A cautionary note on the use of inter-rater reliability in evaluating grant peer review.

Erosheva, E., Martinkova, P., & Lee, C. J. (2021b). When zero may not be zero: A cautionary note on the use of inter-rater reliability in evaluating grant peer review. *Journal of the Royal Statistical Society – Series A*. Accepted.

See Also

[ICCRestricted](#)

plot.sia_parallel *Plot Method for Parallel Analysis Output*

Description

You can call this method to plot an existing object resulting from `fa_parallel()` function, which behaves as a standard `data.frame`, but can be automatically recognized and processed with a dedicated plot method. Also, you can *post-hoc* disable the Kaiser boundaries shown by default.

Usage

```
## S3 method for class 'sia_parallel'
plot(x, y, ...)
```

Arguments

`x` object of class `sia_parallel` to plot.
`y` *ignored*
`...` additional argument:
 `show_kaiser` *logical*, whether to show horizontal lines denoting Kaiser boundaries (eigenvalue 0 and/or 1 for FA and/or PCA, respectively). Defaults to TRUE.

Examples

```
## Not run:
fa_parallel_result <- BFI2[, 1:60] %>% fa_parallel(plot = FALSE) # without plot
fa_parallel_result %>% plot # generate plot from "fitted" object
fa_parallel_result %>% plot(show_kaiser = FALSE) # hide Kaiser boundaries

## End(Not run)
```

plotAdjacent *Plot category probabilities of adjacent category logit model*

Description

Function for plotting category probabilities function estimated by `vglm()` function from the VGAM package using the **ggplot2** 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`.

Value

An object of class `ggplot` and/or `gg`.

Author(s)

Tomas Jurica
Institute of Computer Science of the Czech Academy of Sciences

Adela Hladka
Institute of Computer Science of the Czech Academy of Sciences
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

[vglm](#)

Examples

```
# loading packages
library(VGAM)

# loading data
data(Science, package = "mirt")

# total score calculation
score <- rowSums(Science)
Science[, 1] <- factor(Science[, 1], levels = sort(unique(Science[, 1])), ordered = TRUE)

# adjacent category logit model for item 1
fit <- vglm(Science[, 1] ~ score, family = acat(reverse = FALSE, parallel = TRUE))
# coefficients for item 1
coef(fit)

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

plotCumulative	<i>Plot cumulative and category probabilities of cumulative logit model</i>
----------------	---

Description

Function for plotting cumulative and category probabilities function estimated by `vglm()` function from the VGAM package using the **ggplot2** package.

Usage

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

Arguments

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

Value

An object of class `ggplot` and/or `gg`.

Author(s)

Tomas Jurica
Institute of Computer Science of the Czech Academy of Sciences

Adela Hladka
Institute of Computer Science of the Czech Academy of Sciences
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

[vglm](#)

Examples

```
# loading packages
library(VGAM)

# loading data
data(Science, package = "mirt")

# total score calculation
score <- rowSums(Science)
Science[, 1] <- factor(Science[, 1], levels = sort(unique(Science[, 1])), ordered = TRUE)

# cumulative logit model for item 1
fit <- vglm(Science[, 1] ~ score, family = cumulative(reverse = TRUE, parallel = TRUE))
# coefficients for item 1
coef(fit)

plotCumulative(fit, type = "cumulative", matching.name = "Total score")
plotCumulative(fit, type = "category", matching.name = "Total score")
```

plotDIFirt

Plot item characteristic curve of DIF IRT model

Description

Plots characteristic curve of IRT model.

Usage

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

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 .
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 the `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 of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

[itemParEst](#), [difLord](#), [difRaju](#)

Examples

```
# 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")
# plot of item 1 and Lord's statistic
plotDIFirt(fitLord$itemParInit, item = 1)

# Estimation of 2PL IRT model and Raju's statistic
# by difR package
fitRaju <- difRaju(GMAT2, group = 21, focal.name = 1, model = "2PL")
# plot of item 1 and Lord's statistic
plotDIFirt(fitRaju$itemParInit, test = "Raju", item = 1)
```

plotDIFLogistic *Function for characteristic curve of 2PL logistic DIF model*

Description

Plots characteristic curve of 2PL logistic DIF model

Usage

```
plotDIFLogistic(x, item = 1, item.name, group.names = c("Reference", "Focal"),  
Data, group, match, draw.empirical = TRUE)
```

Arguments

x	an object of "Logistic" class. See Details .
item	numeric: number of item to be plotted
item.name	character: the name of item to be used as title of plot.
group.names	character: names of reference and focal group.
Data	numeric: the data matrix. See Details .
group	numeric: the vector of group membership. See Details .
match	character or numeric: specifies observed score used for matching. Can be either "score", or numeric vector of the same length as number of observations in Data. See Details .
draw.empirical	logical: whether empirical probabilities should be calculated and plotted. Default value is TRUE.

Details

This function plots characteristic curves of 2PL logistic DIF model fitted by difLogistic() function from difR package using ggplot2.

Data and group are used to calculate empirical probabilities for reference and focal group. match should be the same as in x\$match. In case that an observed score is used as a matching variable instead of the total score or the standardized score, match needs to be a numeric vector of the same length as the number of observations in Data.

Author(s)

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

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also[difLogistic](#), [ggplot](#)**Examples**

```
# loading libraries
library(difR)

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

# DIF detection using difLogistic() function
x <- difLogistic(Data, group, focal.name = 1)
# Characteristic curve by logistic regression model
plotDIFLogistic(x, item = 1, Data = Data, group = group)

# Using name of column as item identifier
plotDIFLogistic(x, item = "Item1", Data = Data, group = group)

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

# Not plotting empirical probabilities
plotDIFLogistic(x, item = 1, draw.empirical = FALSE)
```

`plotDistractorAnalysis`*Plot item distractor analysis*

Description

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

Usage

```
plotDistractorAnalysis(
  Data,
  key,
  num.groups = 3,
  item = 1,
  item.name,
  multiple.answers = TRUE,
  criterion = NULL,
  crit.discrete = FALSE,
  cut.points,
```



```

    data,
    matching,
    match.discrete
  )

```

Arguments

Data	character: data matrix or data.frame with rows representing unscored item response from a multiple-choice test and columns corresponding to the items.
key	character: answer key for the items. The key must be a vector of the same length as ncol(Data). In case it is not provided, criterion needs to be specified.
num.groups	numeric: number of groups to which are the respondents splitted.
item	numeric: the number of the item to be plotted.
item.name	character: the name of the item.
multiple.answers	logical: should be all combinations plotted (default) or should be answers splitted into distractors. See Details .
criterion	numeric: numeric vector. If not provided, total score is calculated and distractor analysis is performed based on it.
crit.discrete	logical: is criterion discrete? Default value is FALSE.
cut.points	numeric: numeric vector specifying cut points of criterion.
data	deprecated. Use argument Data instead.
matching	deprecated. Use argument criterion instead.
match.discrete	deprecated. Use argument crit.discrete instead.

Details

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

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

Author(s)

Adela Hladka
 Institute of Computer Science of the Czech Academy of Sciences
 <hladka@cs.cas.cz>

Patricia Martinkova
 Institute of Computer Science of the Czech Academy of Sciences
 <martinkova@cs.cas.cz>

See Also

[DistractorAnalysis](#), [distractor.analysis](#)

Examples

```
# loading 100-item medical admission test datasets
data(dataMedical, 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 = FALSE)
# correct answer B does not function well
plotDistractorAnalysis(data, key, item = 57, multiple.answers = FALSE)
# all options function well, thus the whole item discriminates well
plotDistractorAnalysis(data, key, item = 32, multiple.answers = FALSE)
# functions well, thus the whole item discriminates well
## Not run:
# 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 criterion and key option
criterion <- round(rowSums(dataBin), -1)
plotDistractorAnalysis(data, key, item = 57, criterion = criterion)
# distractor plot for item 57 using specified criterion without key option
plotDistractorAnalysis(data, item = 57, criterion = criterion)

# distractor plot for item 57 using discrete criterion
plotDistractorAnalysis(data, key,
  item = 57, criterion = criterion,
  crit.discrete = TRUE
)

# 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)
```

Description

Plots category probabilities functions estimated by `multinom()` from the `nnet` package using the `ggplot2` package.

Usage

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

Arguments

<code>x</code>	object of class <code>multinom</code>
<code>matching</code>	numeric: vector of matching criterion used for estimation in <code>x</code> .
<code>matching.name</code>	character: name of matching criterion used for estimation in <code>x</code> .

Value

An object of class `ggplot` and/or `gg`.

Author(s)

Adela Hladka
Institute of Computer Science of the Czech Academy of Sciences
<hladka@cs.cas.cz>

Tomas Jurica
Institute of Computer Science of the Czech Academy of Sciences

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

[multinom](#)

Examples

```
# loading data
data(GMAT, GMATtest, GMATkey, package = "difNLR")

matching <- scale(rowSums(GMAT[, 1:20])) # Z-score

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

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

plot_corr

*Compute and plot an item correlation matrix***Description**

Computes and visualizes an item correlation matrix (also known as a heatmap), offering several correlation "types" and optional clustering (with possible cluster outlining). The function relies on [ggplot2](#) package, providing a high customisability using "the grammar of graphics" (see the examples below).

Usage

```
plot_corr(
  Data,
  cor = "pearson",
  clust_method = "none",
  n_clust = 0,
  shape = "circle",
  labels = FALSE,
  labels_size = 3,
  line_size = 0.5,
  line_col = "black",
  line_alpha = 1,
  fill = NA,
  fill_alpha = NA,
  ...
)
```

Arguments

Data	matrix, data.frame or tibble: either a data.frame with scored items (as columns, one observation per row), or a correlation matrix.
cor	character: correlation "type" used to correlation matrix computation; available options are "poly", "tetra", "pearson", "spearman", or "none" (in case you provide the correlation matrix directly instead). You can use an unambiguous abbreviation.
clust_method	character: optional clustering method, available options are: "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC), "centroid" (= UPGMC) or "none" (clustering disabled). See hclust for a detailed description of available options.
n_clust	integer: the number of clusters you want to be outlined. When set to zero, clustering is disabled, ignoring the clust_method argument.
shape	character: tile appearance; either circle (default) to map the correlation coefficient to circle size and color, or square to draw square-shaped tiles with only shade denoting the coefficient magnitude. You can use an unambiguous abbreviation of the two.

labels	logical: when TRUE, the correlation coefficients are plotted onto tiles.
labels_size	numeric: label size in points (pts).
line_size	numeric: cluster outline width.
line_col	character: color of the outline, either a HEX code (e.g. "#123456"), or one of R's standard colors (see the colors).
line_alpha	numeric 0-1: the opacity of the outline.
fill	character: the color used to fill the outlined clusters.
fill_alpha	numeric 0-1: the opacity of the fill color.
...	Arguments passed on to <code>psych::polychoric</code>
correct	Correction value to use to correct for continuity in the case of zero entry cell for tetrachoric, polychoric, polybi, and mixed.cor. See the examples for the effect of correcting versus not correcting for continuity.
smooth	if TRUE and if the tetrachoric/polychoric matrix is not positive definite, then apply a simple smoothing algorithm using <code>cor.smooth</code>
global	When finding pairwise correlations, should we use the global values of the tau parameter (which is somewhat faster), or the local values (<code>global=FALSE</code>)? The local option is equivalent to the <code>polycor</code> solution, or to doing one correlation at a time. <code>global=TRUE</code> borrows information for one item pair from the other pairs using those item's frequencies. This will make a difference in the presence of lots of missing data. With very small sample sizes with <code>global=FALSE</code> and <code>correct=TRUE</code> , the function will fail (for as yet undetermined reasons).
polycor	A no longer used option, kept to stop other packages from breaking.
weight	A vector of length of the number of observations that specifies the weights to apply to each case. The NULL case is equivalent of weights of 1 for all cases.
std.err	<code>std.err=FALSE</code> does not report the standard errors (faster) deprecated
progress	Show the progress bar (if not doing multicores)
ML	<code>ML=FALSE</code> do a quick two step procedure, <code>ML=TRUE</code> , do longer maximum likelihood — very slow! Deprecated
delete	Cases with no variance are deleted with a warning before proceeding.
max.cat	The maximum number of categories to bother with for polychoric.

Details

Correlation heatmap displays selected type of correlations between items. The color of tiles indicates how much and in which way the items are correlated - red color means positive correlation and blue color means negative correlation. Correlation heatmap can be reordered using hierarchical clustering method specified with `clust_method` argument. When the desired number of clusters (argument `n_clust`) is not zero and some clustering is demanded, the rectangles outlining the found clusters are drawn.

Value

An object of class `ggplot` and/or `gg`.

Author(s)

Jan Netik
 Charles University
 Patricia Martinkova
 Institute of Computer Science of the Czech Academy of Sciences
 <martinkova@cs.cas.cz>

Examples

```
# use first 20 columns from HCI dataset (the remainder are not items)
HCI <- HCI[, 1:20]

# use Pearson product-moment correlation coefficient for matrix computation
plot_corr(HCI, cor = "pearson")
## Not run:
# use tetrachoric correlation and reorder the resulting heatmap
# using Ward's method
HCI %>% plot_corr(cor = "tetra", clust_method = "ward.D")

# outline 3 Ward's clusters with bold yellow line and add labels
HCI %>%
  plot_corr(
    n_clust = 3, clust_method = "ward.D", line_col = "yellow",
    line_size = 1.5, labels = TRUE
  )

# add title and position the legend below the plot
library(ggplot2)
HCI %>% plot_corr(n_clust = 3) +
  ggtitle("HCI heatmap") +
  theme(legend.position = "bottom")

# mimic the look of corrplot package
plot_corr(HCI, cor = "poly", clust_method = "complete", shape = "sq") +
  scale_fill_gradient2(
    limits = c(-.1, 1),
    breaks = seq(-.1, 1, length.out = 12),
    guide = guide_colorbar(
      barheight = .8, barwidth = .0275,
      default.unit = "npc",
      title = NULL, frame.colour = "black", ticks.colour = "black"
    )
  ) + theme(axis.text = element_text(colour = "red", size = 12))

## End(Not run)
```

Description

recode_nr() function recognizes and recodes not-reached responses, i.e., missing responses to items such that all subsequent items are missed as well by the respondent.

Usage

```
recode_nr(Data, nr_code = 99, df)
```

Arguments

Data	matrix or data.frame: object to be recoded, must include only items columns and no additional information
nr_code	single character, integer or numeric: specifying how should be recognized not-reached responses coded (default is 99)
df	deprecated. Use argument Data instead.

Value

A data.frame object.

Author(s)

Jan Netik
Charles University

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

[ItemAnalysis](#)

Examples

```
data(HCI, package = "ShinyItemAnalysis")
HCImissed <- HCI[, 1:20]

# simulate skipped (missed) and not-reached items in HCI dataset
set.seed(4211)
for (i in 1:150) {
  # not-reached (minimum at 10th item, maximum at 20th)
  HCImissed[sample(1:nrow(HCImissed), 1), seq(sample(10:20, 1), 20)] <- NA

  # missed with random location
  HCImissed[sample(1:nrow(HCImissed), 1), sample(1:20, 1)] <- NA
}
```

```
summary(HCImissed)

HCImissedNR <- recode_nr(HCImissed, nr_code = 99)
head(HCImissedNR)
summary(HCImissedNR)
```

startShinyItemAnalysis

Start ShinyItemAnalysis application

Description

An interactive shiny application to run test and item analysis. By default, the function runs the application as a background process (Jobs tab in the 'RStudio'). User is then free to use the Console for other work and to try the sample R code examples. You can still run the app the usual way in the console by specifying `background = FALSE`.

Usage

```
startShinyItemAnalysis(background = TRUE)
```

Arguments

`background` logical: should the application be run as a background process (in the 'RStudio')?

Value

No return value. Called for side effects.

Author(s)

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

Adela Hladka
Institute of Computer Science of the Czech Academy of Sciences
<hladka@cs.cas.cz>

Examples

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

## End(Not run)
```

TestAnxietyCor	<i>Correlation matrix for the test anxiety dataset</i>
----------------	--

Description

The TestAnxietyCor dataset contains between-item correlations for 20 items of the Test Anxiety dataset.

Usage

```
data(TestAnxietyCor)
```

Format

TestAnxietyCor is a `data.frame` consisting of between-item correlations for 20 items.

- i1** Lack of confidence during tests.
- i2** Uneasy, upset feeling.
- i3** Thinking about grades.
- i4** Freeze up.
- i5** Thinking about getting through school.
- i6** The harder I work, the more confused I get.
- i7** Thoughts interfere with concentration.
- i8** Jittery when taking tests.
- i9** Even when prepared, get nervous.
- i10** Uneasy before getting the test back.
- i11** Tense during test.
- i12** Exams bother me.
- i13** Tense/ stomach upset.
- i14** Defeat myself during tests.
- i15** Panicky during tests.
- i16** Worry before important tests.
- i17** Think about failing.
- i18** Heart beating fast during tests.
- i19** Can't stop worrying.
- i20** Nervous during test, forget facts.

References

Bartholomew, D. J., Steele, F., & Moustaki, I. (2008). Analysis of multivariate social science data. CRC press.

`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

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

Arguments

<code>base_size</code>	base font size
<code>base_family</code>	base font family

See Also

[ggtheme](#)

Examples

```
library(ggplot2)
data(GMAT, package = "difNLR")
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()
```

Index

* datasets

- AIBS, [5](#)
 - BFI2, [7](#)
 - CZmatura, [8](#)
 - CZmaturaS, [9](#)
 - dataMedical, [10](#)
 - dataMedicalgraded, [11](#)
 - dataMedicalkey, [12](#)
 - dataMedicaltest, [13](#)
 - HCI, [25](#)
 - HCIdata, [26](#)
 - HCIgrads, [27](#)
 - HCIkey, [28](#)
 - HCIprepost, [29](#)
 - HCItest, [30](#)
 - HCItestretest, [31](#)
 - LearningToLearn, [37](#)
 - MScclinical, [39](#)
 - NIH, [40](#)
 - TestAnxietyCor, [57](#)
 - _PACKAGE (ShinyItemAnalysis-package), [3](#)
- AIBS, [4, 5](#)
- BFI2, [7](#)
- colors, [53](#)
- CZmatura, [4, 8, 10](#)
- CZmaturaS, [4, 9, 9](#)
- dataMedical, [4, 10, 11–13](#)
- dataMedicalgraded, [4, 10, 11, 12, 13](#)
- dataMedicalkey, [4, 10, 11, 12, 13](#)
- dataMedicaltest, [4, 10–12, 13](#)
- DDplot, [3, 14, 22, 23, 37](#)
- difLogistic, [48](#)
- difLord, [46](#)
- difRaju, [46](#)
- discrim, [16](#)
- distractor.analysis, [18, 50](#)
- DistractorAnalysis, [3, 17, 49, 50](#)
- fa, [20](#)
- fa_parallel, [4, 19](#)
- gDiscrim, [4, 16, 22, 37](#)
- ggplot, [16, 48](#)
- ggplot2, [52](#)
- ggtheme, [58](#)
- ggWrightMap, [4, 24](#)
- HCI, [4, 25, 27–32](#)
- HCIdata, [4, 26, 26, 28–32](#)
- HCIgrads, [4, 26, 27, 27, 29–32](#)
- HCIkey, [4, 26–28, 28, 30–32](#)
- HCIprepost, [4, 26–29, 29, 31, 32](#)
- HCItest, [4, 26–30, 30, 32](#)
- HCItestretest, [4, 26–31, 31](#)
- hclust, [52](#)
- ICCrestricted, [4, 6, 32, 41](#)
- ItemAnalysis, [4, 34, 55](#)
- itemParEst, [46](#)
- LearningToLearn, [4, 37](#)
- lme4::bootMer, [33](#)
- MScclinical, [39](#)
- multinom, [51](#)
- NIH, [4, 40](#)
- plot.sia_parallel, [42](#)
- plot_corr, [4, 52](#)
- plotAdjacent, [4, 42](#)
- plotCumulative, [4, 44](#)
- plotDIFirt, [4, 45](#)
- plotDIFlogistic, [4, 47](#)
- plotDistractorAnalysis, [3, 48](#)
- plotMultinomial, [4, 50](#)
- psych, [20](#)

`psych::polychoric`, [20](#), [53](#)

`recode_nr`, [4](#), [36](#), [37](#), [54](#)

`ShinyItemAnalysis-package`, [3](#)

`startShinyItemAnalysis`, [3](#), [56](#)

`TestAnxietyCor`, [57](#)

`theme_app`, [58](#)

`vglm`, [43](#), [44](#)