# Package ‘ShinyItemAnalysis’

November 27, 2024

**Type** Package

**Title** Test and Item Analysis via Shiny

**Version** 1.5.2

**Date** 2024-11-27

**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** https://www.ShinyItemAnalysis.org

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

**Depends** R (>= 3.5.0)

**Imports** difR (>= 5.0), dplyr, ggplot2 (>= 3.4.0), lme4, magrittr, methods, mirt (>= 1.43), nnet, psych (>= 2.1.9), purrr, rlang, rstudioapi, tibble, tidyr

**Suggests** data.table, deltaPlotR, difNLR (>= 1.4.1), DT, ggdendro, gridExtra, knitr, latticeExtra, ltm, plotly, rmarkdown, shiny (>= 1.6.0), shinyBS, shinydashboard, shinyjs, stringr, testthat (>= 3.0.0), VGAM, xtable, yaml

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**Config/testthat/edition** 3

**NeedsCompilation** no

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

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

**Repository** CRAN

**Date/Publication** 2024-11-27 14:20:02 UTC

# Contents

ShinyItemAnalysis-package . . . . . 3
AIBS . . . . . 5
Anxiety . . . . . 7
AttitudesExpulsion . . . . . 8
BFI2 . . . . . 10
BlisClass-class . . . . . 11
CLoSEread6 . . . . . 12
coef,BlisClass-method . . . . . 13
CZmatura . . . . . 14
CZmaturaS . . . . . 15
dataMedical . . . . . 16
dataMedicalgraded . . . . . 17
dataMedicalkey . . . . . 18
dataMedicaltest . . . . . 18
DDplot . . . . . 19
DistractorAnalysis . . . . . 22
EPIA . . . . . 24
fa_parallel . . . . . 25
fit_blis . . . . . 28
gDiscrim . . . . . 36
get_orig_levels . . . . . 38
ggWrightMap . . . . . 39
GMAT . . . . . 40
HCI . . . . . 41
HCIdata . . . . . 42
HCIgrads . . . . . 44
HCIkey . . . . . 45
HCIlong . . . . . 46
HCIprepost . . . . . 47
HCItest . . . . . 48
HCItestretest . . . . . 49
HeightInventory . . . . . 50
ICCrestricted . . . . . 51
ItemAnalysis . . . . . 53
LearningToLearn . . . . . 56
MSATB . . . . . 57
MSclinical . . . . . 58
NIH . . . . . 59
nominal_to_int . . . . . 60
obtain_nrm_def . . . . . 61
plot.sia_parallel . . . . . 62

plotAdjacent . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
plotCumulative . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
plotDIFirt . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
plotDIFLogistic . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
plotDistractorAnalysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
plotMultinomial . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71
plot_corr . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
print.blis_coefs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
recode_nr . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76
ShinyItemAnalysis_options . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77
startShinyItemAnalysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
TestAnxietyCor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80
theme_app . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81

**Index** **83**

---

`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 book by Martinkova and Hladka (2023), and from paper by Martinkova and Drabinova (2018).

## Functions

- `startShinyItemAnalysis()`
- `DDplot()`
- `DistractorAnalysis()`
- `plotDistractorAnalysis()`
- `fa_parallel()`
- `gDiscrim()`

- ggWrightMap()
- ICCrestricted()
- ItemAnalysis()
- blis()
- plotAdjacent(), plotCumulative(), plotMultinomial()
- plotDIFirt(), plotDIFLogistic()
- plot_corr()
- recode_nr()

**Datasets**

- AIBS()
- Anxiety()
- AttitudesExpulsion()
- BFI2()
- CLoSEread6()
- CZmatura()
- CZmaturaS()
- dataMedical()
- dataMedicalgraded()
- dataMedicalkey()
- dataMedicaltest()
- HCI()
- HCIdata()
- HCIgrads()
- HCIkey()
- HCIlong()
- HCIprepost()
- HCItest()
- HCItestretest()
- HeightInventory()
- LearningToLearn()
- MSATB()
- MSclinical()
- NIH()
- TestAnxietyCor()

## 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

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

## References

Martinkova, P., & Hladka, A. (2023). Computational Aspects of Psychometric Methods: With R. Chapman and Hall/CRC. doi:10.1201/9781003054313

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

## See Also

Useful links:

- `https://www.ShinyItemAnalysis.org`
- 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

```
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

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

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

### See Also

`ICCrestricted()`

---

## `Anxiety` *PROMIS Anxiety Scale Dataset*

---

### Description

The data contains responses from 766 people sampled from a general population to the PROMIS Anxiety scale (http://www.nihpromis.org) composed of 29 Likert-type questions with a common rating scale (1 = Never, 2 = Rarely, 3 = Sometimes, 4 = Often, and 5 = Always).

### Usage

```
Anxiety
```

### Format

A data frame with 766 observations on the following 32 variables.

`age` 0 = younger than 65 and 1 = 65 and older

`gender` 0 = Male and 1 = Female

`education` 0 = some college or higher and 1 = high school or lower

`R1` I felt fearful

`R2` I felt frightened

`R3` It scared me when I felt nervous

`R4` I felt anxious

`R5` I felt like I needed help for my anxiety

`R6` I was concerned about my mental health

`R7` I felt upset

`R8` I had a racing or pounding heart

`R9` I was anxious if my normal routine was disturbed

`R10` I had sudden feelings of panic

`R11` I was easily startled

`R12` I had trouble paying attention

`R13` I avoided public places or activities

`R14` I felt fidgety

`R15` I felt something awful would happen

`R16` I felt worried

`R17` I felt terrified

`R18` I worried about other people's reactions to me

`R19` I found it hard to focus on anything other than my anxiety

`R20` My worries overwhelmed me

`R21` I had twitching or trembling muscles

`R22` I felt nervous

`R23` I felt indecisive

`R24` Many situations made me worry

`R25` I had difficulty sleeping

`R26` I had trouble relaxing

`R27` I felt uneasy

`R28` I felt tense

`R29` I had difficulty calming down

`score` Total score.

`zscore` Standardized total score.

### Source

Reexport from `lordif` package; http://www.nihpromis.org

### References

PROMIS Cooperative Group. Unpublished Manual for the Patient-Reported Outcomes Measurement Information System (PROMIS) Version 1.1. October, 2008: http://www.nihpromis.org

## `AttitudesExpulsion` *Attitudes towards the Expulsion of the Sudeten Germans (dataset)*

### Description

Dataset from Kolek et al. (2021) study investigating a video game's effects on implicit and explicit attitudes towards depicted historical events in the short- and long-term. As an intervention tool, a serious game *Czechoslovakia 38–89: Borderlands* was utilized that deals with the expulsion of the Sudeten Germans from the former Czechoslovakia after the WWII. Data consists responses from 145 adults from two groups (experimental and control group) on number of multi-item measurements.

## Usage

```
AttitudesExpulsion
```

## Format

A *data.frame* with 145 rows and 239 variables:

**ID** anonymous identifier

**Group** C = control or E = experimental group

**Gender** *factor*, `male` or `female`

**GenderF** *integer*, 1 = female

**Merkel** effect of Merkel speech between the posttest and the delayed posttest, range 0–5, where 0 stands for no effect, 5 for very significant effect

**Sudety** *factor*, N = not originally from Czech Borderlands; Y = originally from Czech Borderlands

**Education** *factor*, V = university; S = high school; Z = elementary school

**Education123** *integer*, same as above, but coded as 3= university; 2= high school; 1= elementary school, meaning higher the number, higher the education

***PANASpn** total PANAS score of positive and negative affect scales

***PANASp** total PANAS score of positive affect scale

***PANASn** total PANAS score of negative affect scale

***Macro** Macro attitude measurement

***Micro** Micro attitude measurement

***IATeffect** Single-Category Implicit association test score

Items beginning with an asterisk have following prefixes in the actual dataset:

**pre** pretest

**post** immediate posttest

**del** one month delayed posttest

**Post_Pre** difference between posttest_pretest

**Del_Post** difference between delayed posttest and posttest

## Source

Kolek, L., Šisler, V., Martinková, P., & Brom, C. (2021). Can video games change attitudes towards history? Results from a laboratory experiment measuring short- and long-term effects. *Journal of Computer Assisted Learning, 1–22*. doi:10.1111/jcal.12575

BFI2 *BFI2 Dataset*

## 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

```
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

```
colnames(BFI2)[1:60] <- c(
  "iEscb01", "iAcmp02", "iCorg03r", "iNanx04r", "iOaes05r", "iEasr06",
  "iArsp07", "iCprd08r", "iNdep09r", "iOint10", "iEenl11r", "iAtrs12r", "iCrsp13", "iNemt14",
  "iOcrt15", "iEscb16r", "iAcmp17r", "iCorg18", "iNanx19", "iOaes20", "iEasr21", "iArsp22r",
  "iCprd23r", "iNdep24r", "iOint25r", "iEenl26r", "iAtrs27", "iCrsp28r", "iNemt29r",
  "iOcrt30r", "iEscb31r", "iAcmp32", "iCorg33", "iNanx34", "iOaes35", "iEasr36r", "iArsp37r",
  "iCprd38", "iNdep39", "iOint40", "iEenl41", "iAtrs42r", "iCrsp43", "iNemt44r", "iOcrt45r",
  "iEscb46", "iAcmp47r", "iCorg48r", "iNanx49r", "iOaes50r", "iEasr51r", "iArsp52", "iCprd53",
  "iNdep54", "iOint55r", "iEenl56", "iAtrs57", "iCrsp58r", "iNemt59", "iOcrt60"
)
```

# `BlisClass-class` *BLIS S4 class*

## Description

Extends `mirt`'s `SingleGroupClass` directly (meaning all `mirt` methods that work with that class will work with BlisClass too; make sure `mirt` is loaded).

## Details

The purpose of the class is to have a custom `coef` method (see coef,BlisClass-method) dispatched and the original levels with correct response (as a `key` attribute) stored in the resulting fitted model.

## Slots

`orig_levels` *list* of original levels with logical attribute `key`, which stores the information on which response (level) has been considered as correct. Note that levels not used in the original data are dropped.

## See Also

Other BLIS/BLIRT related: `coef,BlisClass-method`, `fit_blis()`, `get_orig_levels()`, `nominal_to_int()`, `obtain_nrm_def()`, `print.blis_coefs()`

---

# `CLoSEread6` *Czech Longitudinal Study in Education (CLoSE) - reading in 6th grade*

---

## Description

`CLoSEread6` dataset consists of the dichotomously scored responses of 2,634 students (1,324 boys, 1,310 girls) on 19 multiple-choice items in a test of reading skills, version B, taken in the 6th grade. Item responses were dichotomized: 1 point was awarded only if the answer was fully correct and 0 if it was not (Greger, Straková, & Martinková, 2022; Martinková, Hladká, & Potužníková, 2020; Hladká, Martinková, & Magis, 2023)

## Usage

```
CLoSEread6
```

## Format

`CLoSEread6` is a `data.frame` consisting of 2,634 observations on the 20 variables.

**Q6B_1-Q6B_19** Dichotomously scored items of the test on reading skills.

**gender** Gender membership, `"0"` boys, `"1"` girls.

## Source

Hladká, A., Martinková, P., & Magis, D. (2023). Combining item purification and multiple comparison adjustment methods in detection of differential item functioning. *Multivariate Behavioral Research*, In Press.

## References

Greger, D., Straková, J., & Martinková, P. (2022). Extending the ILSA study design to a longitudinal design. TIMSS & PIRLS extension in the Czech Republic: CLoSE study. In T. Nilsen, A. Stancel-Piatak, & J.-E. Gustafsson (Eds.), *Springer international handbooks of education. International handbook of comparative large-scale studies in education: Perspectives, methods and findings*. Springer. doi:10.1007/9783030382988_311

Martinková, P., Hladká, A., & Potužníková, 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

---

# `coef,BlisClass-method` *Get Coefficients from a fitted BLIS model*

---

## Description

Extracts item parameters from fitted BLIS model. For BLIRT parametrization, use `IRTpars = TRUE` in your function call. Contrary to mirt::coef,SingleGroupClass-method, response category labels can be displayed in the output using `labels = TRUE`. On top of that, as BLIS/BLIRT parametrizations utilize the information of correct response category, you can denote these in the output with `mark_correct = TRUE`.

## Usage

```
## S4 method for signature 'BlisClass'
coef(
  object,
  ...,
  CI = 0.95,
  printSE = FALSE,
  IRTpars = FALSE,
  simplify = FALSE,
  labels = FALSE,
  mark_correct = labels
)
```

## Arguments

| | |
|---|---|
| `object` | *object of class BlisClass*, model fitted via `fit_blis()` or `blis()`. |
| `...` | Additional arguments. Not utilized at the moment. |
| `CI` | *numeric*, a width of the confidence intervals. |
| `printSE` | *logical*, print standard errors instead of CI? Defaults to `FALSE`. |
| `IRTpars` | *logical*, convert slope intercept parameters into IRT parameters (i.e. BLIRT)? Defaults to `FALSE`. |

simplify — *logical*, return coefficients as a matrix, instead of list? Defaults to FALSE. *Not implemented yet.*

labels — *logical*, if TRUE, show response labels (e.g. "A", "B", "C") instead of response numeric indices (e.g. 0, 1, 2). Defaults to FALSE.

mark_correct — *logical*, mark the correct response with an asterisk symbol. Applicable only if labels is TRUE (in which case, mark_correct defaults to TRUE).

### Value

List of item coefficients of S3 class blis_coefs, so the resulting output of coef() call is formatted to display only first 3 digits (you can opt for different rounding via the print.blis_coefs method, see the examples). Note that the list-object returned invisibly has the raw coefficients stored in it.

### See Also

Other BLIS/BLIRT related: BlisClass-class, fit_blis(), get_orig_levels(), nominal_to_int(), obtain_nrm_def(), print.blis_coefs()

### Examples

```
fitted_blis <- fit_blis(HCItest[, 1:20], HCIkey)

# BLIS coefs
coef(fitted_blis)

# BLIRT coefs
coef(fitted_blis, IRTpars = TRUE)

# store raw coefs
blis_coefs <- coef(fitted_blis)

# print coefs rounded to 2 digits
print(blis_coefs, digits = 2)
```

## CZmatura *CZmatura dataset*

### Description

The CZmatura dataset comes from matura exam in mathematics. The exam was assigned in 2019 to students from Grade 13, at the end of their secondary education. Original data available from https://cermat.cz/.

### Usage

```
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 `CZmaturaS` dataset comes from a matura exam in mathematics. The exam was assigned in 2019 to students in Grade 13, at the end of their secondary education. This is a random sample of 2,000 students from a total of 15,702. Original data available from `https://cermat.cz/`.

### Usage

```
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

```
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).

### Source

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.

### 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

### 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

```
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).

### Source

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.

### 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

### 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

```
dataMedicalkey
```

### Format

A vector with 100 values representing correct answers to items of dataMedicaltest dataset. For more details see dataMedicaltest().

### Source

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.

### 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

### See Also

dataMedical(), dataMedicaltest(), dataMedicalgraded()

## dataMedicaltest *Dataset of admission test to medical school*

### Description

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

### Usage

```
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).

## Source

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.

## 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

## 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, `disrim` 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 `cor(item, criterion) * sqrt(((N - 1) / N) * var(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 a 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

`gDiscrim()` for calculation of generalized ULI
`ggplot2::ggplot()` for general function to plot a `"ggplot"` object

## Examples

```
# 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,
  item = "all",
  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 `ncol(Data)`. In case it is not provided, `criterion` needs to be specified.

`item` numeric or character: either character `"all"` to apply for all items (default), or a vector of item names (column names of `Data`), or item identifiers (integers specifying the column number).

`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 `Data` instead.

`matching` deprecated. Use argument `criterion` instead.

`match.discrete` deprecated. Use argument `crit.discrete` 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

```
Data <- dataMedicaltest[, 1:100]
Databin <- dataMedical[, 1:100]
key <- dataMedicalkey

# distractor analysis for all items
DistractorAnalysis(Data, key)

# distractor analysis for item 1
DistractorAnalysis(Data, key, item = 1)
## Not run:
# distractor analysis with proportions
DistractorAnalysis(Data, key, p.table = TRUE)

# distractor analysis for 6 groups
DistractorAnalysis(Data, key, num.group = 6)

# distractor analysis using specified criterion
criterion <- round(rowSums(Databin), -1)
DistractorAnalysis(Data, key, criterion = criterion)

# distractor analysis using discrete criterion
DistractorAnalysis(Data, key, criterion = criterion, crit.discrete = TRUE)

# distractor analysis using groups specified by cut.points
DistractorAnalysis(Data, key, cut.points = seq(10, 96, 10))

## End(Not run)
```

---

EPIA *The Eysenck Personality Inventory Impulsivity Subscale*

---

## Description

The data came from a published study and was kindly provided by Dr. Ferrando. A group of 1,033 undergraduate students were asked to check on a 112 mm line segment with two end points (almost never, almost always) using their own judgement for the five items taken from the Spanish version of the EPI-A impulsivity subscale. The direct item score was the distance in mm of the check mark from the left end point (Ferrando, 2002).

### Usage

```
EPIA
```

### Format

A data frame with 1033 observations on the following 5 variables. The sixth variable is a total score (i.e. the sum of the items).

`Item 1` Longs for excitement

`Item 2` Does not stop and think things over before doing anything

`Item 3` Often shouts back when shouted at

`Item 4` Likes doing things in which he/she has to act quickly

`Item 5` Tends to do many things at the same time

`score` Total score for the aforementioned items

### Source

Reexport from `EstCRM` package with added total scores.

### References

Ferrando, P. J. (2002). Theoretical and Empirical Comparison between Two Models for Continuous Item Responses. *Multivariate Behavioral Research*, *37*(4), 521–542.

Zopluoglu C (2022). *EstCRM: Calibrating Parameters for the Samejima's Continuous IRT Model*. R package version 1.5, `https://CRAN.R-project.org/package=EstCRM`.

## `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` *character*, how to calculate the correlation matrix of the real data. Can be either `pearson` (default), `tetrachoric` or `polychoric`. Unambiguous abbreviations accepted.

`n_obs` *integer*, in case you provided the correlation matrix directly as the input, you have to provide the number of observations in the original dataset.

`method` *character*, either `fa`, `pca`, or `both` (the default). Which method to use for the eigenvalues simulation and computation.

`threshold` *character*, whether to use traditionall Horn's method or more recent and well-performing quantile one. Either `mean` or `quantile` (default). Can be abbreviated.

`p` *numeric* (0–1), probability for which the sample quantile is produced. Defaults to `.95`. Ignored if `threshold = "mean"`.

`n_iter` *integer*, number of iterations, i.e. the number of zero-factor multivariate normal distributions to sample. Defaults to `20`.

`plot` *logical*, 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` *logical*, whether to show Kaiser boundary in the plot (the default) or not.

`fm` *character*, factoring method. See `psych::fa()` from the package `psych::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 `psych::polychoric`

`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 underdetermined 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` × `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
`<netik@cs.cas.cz>`

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

Buja, A., & Eyuboglu, N. (1992). Remarks on parallel analysis. Multivariate Behavioral Research, 27, 509–540. doi:10.1207/s15327906mbr2704_2

## Examples

```
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)
```

---

# fit_blis — *Fit Baseline-category Logit Intercept-Slope (BLIS) model on nominal data*

---

## Description

`blis` fits the IRT Nominal Response Model to data from multiple-choice tests, while accounting for the correct answer and treating this option as a baseline in this baseline-category logit model. The intercept-slope parametrization in BLIS can be converted to IRT (difficulty-discrimination) parametrization (BLIRT).

## Usage

```
fit_blis(Data, key, ...)

blis(Data, key, ...)
```

## Arguments

`Data` *data.frame* or *tibble* with all columns being factors. Support for *matrix* is limited and behavior not guaranteed.

`key` A single-column `data.frame`, (**not** matrix) `tibble` or - preferably - a factor vector of levels considered as correct responses.

`...` Arguments passed on to `mirt::mirt`

`SE` logical; estimate the standard errors by computing the parameter information matrix? See `SE.type` for the type of estimates available covdata a data.frame of data used for latent regression models

formula an R formula (or list of formulas) indicating how the latent traits can be regressed using external covariates in covdata. If a named list of formulas is supplied (where the names correspond to the latent trait names in model) then specific regression effects can be estimated for each factor. Supplying a single formula will estimate the regression parameters for all latent traits by default

itemdesign a data.frame with rows equal to the number of items and columns containing any item-design effects. If items should be included in the design structure (i.e., should be left in their canonical structure) then fewer rows can be used, however the rownames must be defined and matched with colnames in the data input. The item design matrix is constructed with the use of item.formula. Providing this input will fix the associated 'd' intercepts to 0, where applicable

item.formula an R formula used to specify any intercept decomposition (e.g., the LLTM; Fischer, 1983). Note that only the right-hand side of the formula is required for compensatory models.

For non-compensatory itemtypes (e.g., 'PC1PL') the formula must include the name of the latent trait in the left hand side of the expression to indicate which of the trait specification should have their intercepts decomposed (see MLTM; Embretson, 1984)

SE.type type of estimation method to use for calculating the parameter information matrix for computing standard errors and wald tests. Can be:

- 'Richardson', 'forward', or 'central' for the numerical Richardson, forward difference, and central difference evaluation of observed Hessian matrix
- 'crossprod' and 'Louis' for standard error computations based on the variance of the Fisher scores as well as Louis' (1982) exact computation of the observed information matrix. Note that Louis' estimates can take a long time to obtain for large sample sizes and long tests
- 'sandwich' for the sandwich covariance estimate based on the 'crossprod' and 'Oakes' estimates (see Chalmers, 2018, for details)
- 'sandwich.Louis' for the sandwich covariance estimate based on the 'crossprod' and 'Louis' estimates
- 'Oakes' for Oakes' (1999) method using a central difference approximation (see Chalmers, 2018, for details)
- 'SEM' for the supplemented EM (disables the accelerate option automatically; EM only)
- 'Fisher' for the expected information, 'complete' for information based on the complete-data Hessian used in EM algorithm
- 'MHRM' and 'FMHRM' for stochastic approximations of observed information matrix based on the Robbins-Monro filter or a fixed number of MHRM draws without the RM filter. These are the only options supported when method = 'MHRM'
- 'numerical' to obtain the numerical estimate from a call to optim when method = 'BL'

Note that both the 'SEM' method becomes very sensitive if the ML solution has has not been reached with sufficient precision, and may be further sensitive if the history of the EM cycles is not stable/sufficient for convergence of the respective estimates. Increasing the number of iterations (increasing NCYCLES and decreasing TOL, see below) will help to improve the accuracy, and can be run in parallel if a mirtCluster object has been defined (this will be used for Oakes' method as well). Additionally, inspecting the symmetry of the ACOV matrix for convergence issues by passing technical = list(symmetric = FALSE) can be helpful to determine if a sufficient solution has been reached

method a character object specifying the estimation algorithm to be used. The default is 'EM', for the standard EM algorithm with fixed quadrature, 'QMCEM' for quasi-Monte Carlo EM estimation, or 'MCEM' for Monte Carlo EM estimation. The option 'MHRM' may also be passed to use the MH-RM algorithm, 'SEM' for the Stochastic EM algorithm (first two stages of the MH-RM stage using an optimizer other than a single Newton-Raphson iteration), and 'BL' for the Bock and Lieberman approach (generally not recommended for longer tests).

The 'EM' is generally effective with 1-3 factors, but methods such as the 'QMCEM', 'MCEM', 'SEM', or 'MHRM' should be used when the dimensions are 3 or more. Note that when the optimizer is stochastic the associated SE.type is automatically changed to SE.type = 'MHRM' by default to avoid the use of quadrature

optimizer a character indicating which numerical optimizer to use. By default, the EM algorithm will use the 'BFGS' when there are no upper and lower bounds box-constraints and 'nlminb' when there are.

Other options include the Newton-Raphson ('NR'), which can be more efficient than the 'BFGS' but not as stable for more complex IRT models (such as the nominal or nested logit models) and the related 'NR1' which is also the Newton-Raphson but consists of only 1 update that has been coupled with RM Hessian (only applicable when the MH-RM algorithm is used). The MH-RM algorithm uses the 'NR1' by default, though currently the 'BFGS', 'L-BFGS-B', and 'NR' are also supported with this method (with fewer iterations by default) to emulate stochastic EM updates. As well, the 'Nelder-Mead' and 'SANN' estimators are available, but their routine use generally is not required or recommended.

Additionally, estimation subroutines from the Rsolnp and nloptr packages are available by passing the arguments 'solnp' and 'nloptr', respectively. This should be used in conjunction with the solnp_args and nloptr_args specified below. If equality constraints were specified in the model definition only the parameter with the lowest parnum in the pars = 'values' data.frame is used in the estimation vector passed to the objective function, and group hyper-parameters are omitted. Equality an inequality functions should be of the form function(p, optim_args), where optim_args is a list of internally parameters that largely can be ignored when defining constraints (though use of browser() here may be helpful)

dentype type of density form to use for the latent trait parameters. Current options include

- 'Gaussian' (default) assumes a multivariate Gaussian distribution with an associated mean vector and variance-covariance matrix
- 'empiricalhist' or 'EH' estimates latent distribution using an empirical histogram described by Bock and Aitkin (1981). Only applicable for unidimensional models estimated with the EM algorithm. For this option, the number of cycles, TOL, and quadpts are adjusted accommodate for less precision during estimation (namely: TOL = 3e-5, NCYCLES = 2000, quadpts = 121)
- 'empiricalhist_Woods' or 'EHW' estimates latent distribution using an empirical histogram described by Bock and Aitkin (1981), with the same specifications as in dentype = 'empiricalhist', but with the extrapolation-interpolation method described by Woods (2007). NOTE: to improve stability in the presence of extreme response styles (i.e., all highest or lowest in each item) the technical option zeroExtreme = TRUE may be required to down-weight the contribution of these problematic patterns
- 'Davidian-#' estimates semi-parametric Davidian curves described by Woods and Lin (2009), where the # placeholder represents the number of Davidian parameters to estimate (e.g., 'Davidian-6' will estimate 6 smoothing parameters). By default, the number of quadpts is increased to 121, and this method is only applicable for unidimensional models estimated with the EM algorithm

Note that when itemtype = 'ULL' then a log-normal(0,1) density is used to support the unipolar scaling

constrain a list of user declared equality constraints. To see how to define the parameters correctly use pars = 'values' initially to see how the parameters are labeled. To constrain parameters to be equal create a list with separate concatenated vectors signifying which parameters to constrain. For example, to set parameters 1 and 5 equal, and also set parameters 2, 6, and 10 equal use constrain = list(c(1,5), c(2,6,10)). Constraints can also be specified using the mirt.model syntax (recommended)

calcNull logical; calculate the Null model for additional fit statistics (e.g., TLI)? Only applicable if the data contains no NA's and the data is not overly sparse

draws the number of Monte Carlo draws to estimate the log-likelihood for the MH-RM algorithm. Default is 5000

survey.weights a optional numeric vector of survey weights to apply for each case in the data (EM estimation only). If not specified, all cases are weighted equally (the standard IRT approach). The sum of the survey.weights must equal the total sample size for proper weighting to be applied

quadpts number of quadrature points per dimension (must be larger than 2). By default the number of quadrature uses the following scheme: switch(as.character(nfact), '1'=61, '2'=31, '3'=15, '4'=9, '5'=7, 3). However, if the method input is set to 'QMCEM' and this argument is left blank then the default number of quasi-Monte Carlo integration nodes will be set to 5000 in total

TOL convergence threshold for EM or MH-RM; defaults are .0001 and .001. If SE.type = 'SEM' and this value is not specified, the default is set to 1e-5.

To evaluate the model using only the starting values pass TOL = NaN, and to evaluate the starting values without the log-likelihood pass TOL = NA

gpcm_mats a list of matrices specifying how the scoring coefficients in the (generalized) partial credit model should be constructed. If omitted, the standard gpcm format will be used (i.e., seq(0, k, by = 1) for each trait). This input should be used if traits should be scored different for each category (e.g., matrix(c(0:3, 1,0,0,0), 4, 2) for a two-dimensional model where the first trait is scored like a gpcm, but the second trait is only positively indicated when the first category is selected). Can be used when itemtypes are 'gpcm' or 'Rasch', but only when the respective element in gpcm_mats is not NULL

grsm.block an optional numeric vector indicating where the blocking should occur when using the grsm, NA represents items that do not belong to the grsm block (other items that may be estimated in the test data). For example, to specify two blocks of 3 with a 2PL item for the last item: grsm.block = c(rep(1,3), rep(2,3), NA). If NULL the all items are assumed to be within the same group and therefore have the same number of item categories

rsm.block same as grsm.block, but for 'rsm' blocks

monopoly.k a vector of values (or a single value to repeated for each item) which indicate the degree of the monotone polynomial fitted, where the monotone polynomial corresponds to monopoly.k * 2 + 1 (e.g., monopoly.k = 2 fits a 5th degree polynomial). Default is monopoly.k = 1, which fits a 3rd degree polynomial

large a logical indicating whether unique response patterns should be obtained prior to performing the estimation so as to avoid repeating computations on identical patterns. The default TRUE provides the correct degrees of freedom for the model since all unique patterns are tallied (typically only affects goodness of fit statistics such as G2, but also will influence nested model comparison methods such as anova(mod1, mod2)), while FALSE will use the number of rows in data as a placeholder for the total degrees of freedom. As such, model objects should only be compared if all flags were set to TRUE or all were set to FALSE

Alternatively, if the collapse table of frequencies is desired for the purpose of saving computations (i.e., only computing the collapsed frequencies for the data onte-time) then a character vector can be passed with the arguement large = 'return' to return a list of all the desired table information used by mirt. This list object can then be reused by passing it back into the large argument to avoid re-tallying the data again (again, useful when the dataset are very large and computing the tabulated data is computationally burdensome). This strategy is shown below:

**Compute organized data** e.g., internaldat <- mirt(Science, 1, large = 'return')

**Pass the organized data to all estimation functions** e.g., mod <- mirt(Science, 1, large = internaldat)

GenRandomPars logical; generate random starting values prior to optimization instead of using the fixed internal starting values?

`accelerate` a character vector indicating the type of acceleration to use. Default is `'Ramsay'`, but may also be `'squarem'` for the SQUAREM procedure (specifically, the gSqS3 approach) described in Varadhan and Roldand (2008). To disable the acceleration, pass `'none'`

`verbose` logical; print observed- (EM) or complete-data (MHRM) log-likelihood after each iteration cycle? Default is TRUE

`solnp_args` a list of arguments to be passed to the `solnp::solnp()` function for equality constraints, inequality constraints, etc

`nloptr_args` a list of arguments to be passed to the `nloptr::nloptr()` function for equality constraints, inequality constraints, etc

`spline_args` a named list of lists containing information to be passed to the `bs` (default) and `ns` for each spline itemtype. Each element must refer to the name of the itemtype with the spline, while the internal list names refer to the arguments which are passed. For example, if item 2 were called 'read2', and item 5 were called 'read5', both of which were of itemtype 'spline' but item 5 should use the `ns` form, then a modified list for each input might be of the form:

`spline_args = list(read2 = list(degree = 4), read5 = list(fun = 'ns', knots = c(-2, 2)))`

This code input changes the `bs()` splines function to have a `degree = 4` input, while the second element changes to the `ns()` function with knots set a `c(-2, 2)`

`control` a list passed to the respective optimizers (i.e., `optim()`, `nlminb()`, etc). Additional arguments have been included for the `'NR'` optimizer: `'tol'` for the convergence tolerance in the M-step (default is `TOL/1000`), while the default number of iterations for the Newton-Raphson optimizer is 50 (modified with the `'maxit'` control input)

`technical` a list containing lower level technical parameters for estimation. May be:

**NCYCLES** maximum number of EM or MH-RM cycles; defaults are 500 and 2000

**MAXQUAD** maximum number of quadratures, which you can increase if you have more than 4GB or RAM on your PC; default 20000

**theta_lim** range of integration grid for each dimension; default is `c(-6, 6)`. Note that when `itemtype = 'ULL'` a log-normal distribution is used and the range is change to `c(.01, and 6^2)`, where the second term is the square of the `theta_lim` input instead

**set.seed** seed number used during estimation. Default is 12345

**SEtol** standard error tolerance criteria for the S-EM and MHRM computation of the information matrix. Default is 1e-3

**symmetric** logical; force S-EM/Oakes information matrix estimates to be symmetric? Default is TRUE so that computation of standard errors are more stable. Setting this to FALSE can help to detect solutions that have not reached the ML estimate

**SEM_window** ratio of values used to define the S-EM window based on the observed likelihood differences across EM iterations. The default is `c(0, 1 - SEtol)`, which provides nearly the very full S-EM window (i.e., nearly all EM cycles used). To use the a smaller SEM window change the window to to something like `c(.9, .999)` to start at a point farther into the EM history

**warn** logical; include warning messages during estimation? Default is TRUE

**message** logical; include general messages during estimation? Default is TRUE

**customK** a numeric vector used to explicitly declare the number of response categories for each item. This should only be used when constructing mirt model for reasons other than parameter estimation (such as to obtain factor scores), and requires that the input data all have 0 as the lowest category. The format is the same as the `extract.mirt(mod, 'K')` slot in all converged models

**customPriorFun** a custom function used to determine the normalized density for integration in the EM algorithm. Must be of the form `function(Theta, Etable){...}`, and return a numeric vector with the same length as number of rows in `Theta`. The `Etable` input contains the aggregated table generated from the current E-step computations. For proper integration, the returned vector should sum to 1 (i.e., normalized). Note that if using the `Etable` it will be NULL on the first call, therefore the prior will have to deal with this issue accordingly

**zeroExtreme** logical; assign extreme response patterns a `survey.weight` of 0 (formally equivalent to removing these data vectors during estimation)? When `dentype = 'EHW'`, where Woods' extrapolation is utilized, this option may be required if the extrapolation causes expected densities to tend towards positive or negative infinity. The default is `FALSE`

**customTheta** a custom `Theta` grid, in matrix form, used for integration. If not defined, the grid is determined internally based on the number of `quadpts`

**nconstrain** same specification as the `constrain` list argument, however imposes a negative equality constraint instead (e.g., $a12 = -a21$, which is specified as `nconstrain = list(c(12, 21))`). Note that each specification in the list must be of length 2, where the second element is taken to be -1 times the first element

**delta** the deviation term used in numerical estimates when computing the ACOV matrix with the 'forward' or 'central' numerical approaches, as well as Oakes' method with the Richardson extrapolation. Default is 1e-5

**parallel** logical; use the parallel cluster defined by `mirtCluster`? Default is TRUE

**storeEMhistory** logical; store the iteration history when using the EM algorithm? Default is FALSE. When TRUE, use `extract.mirt` to extract

**internal_constraints** logical; include the internal constraints when using certain IRT models (e.g., 'grsm' itemtype). Disable this if you want to use special optimizers such as the solnp. Default is `TRUE`

**gain** a vector of two values specifying the numerator and exponent values for the RM gain function $(val1/cycle)^val2$. Default is `c(0.10, 0.75)`

**BURNIN** number of burn in cycles (stage 1) in MH-RM; default is 150

**SEMCYCLES** number of SEM cycles (stage 2) in MH-RM; default is 100

**MHDRAWS** number of Metropolis-Hasting draws to use in the MH-RM at each iteration; default is 5

**MHcand** a vector of values used to tune the MH sampler. Larger values will cause the acceptance ratio to decrease. One value is required for each group in unconditional item factor analysis (`mixedmirt()` requires additional values for random effect). If null, these values are determined internally, attempting to tune the acceptance of the draws to be between .1 and .4

**MHRM_SE_draws** number of fixed draws to use when `SE=TRUE` and `SE.type = 'FMHRM'` and the maximum number of draws when `SE.type = 'MHRM'`. Default is 2000

**MCEM_draws** a function used to determine the number of quadrature points to draw for the `'MCEM'` method. Must include one argument which indicates the iteration number of the EM cycle. Default is `function(cycles) 500 + (cycles - 1)*2`, which starts the number of draws at 500 and increases by 2 after each full EM iteration

**info_if_converged** logical; compute the information matrix when using the MH-RM algorithm only if the model converged within a suitable number of iterations? Default is `TRUE`

**logLik_if_converged** logical; compute the observed log-likelihood when using the MH-RM algorithm only if the model converged within a suitable number of iterations? Default is `TRUE`

**keep_vcov_PD** logical; attempt to keep the variance-covariance matrix of the latent traits positive definite during estimation in the EM algorithm? This generally improves the convergence properties when the traits are highly correlated. Default is `TRUE`

## Details

For the details on `coef` method dispatched for fitted BLIS model, see coef,BlisClass-method. To get more on the class, see BlisClass.

## Value

Fitted model of class BlisClass (extending standard `mirt`'s `SingleGroupClass`).

## Author(s)

Jan Netik
Institute of Computer Science of the Czech Academy of Sciences
`<netik@cs.cas.cz>`

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

## See Also

Other BLIS/BLIRT related: BlisClass-class, coef,BlisClass-method, get_orig_levels(), nominal_to_int(), obtain_nrm_def(), print.blis_coefs()

## Examples

```
fitted_blis <- fit_blis(HCItest[, 1:20], HCIkey, SE = TRUE)
coef(fitted_blis)
coef(fitted_blis)$`Item 12`
coef(fitted_blis, IRTpars = TRUE)
coef(fitted_blis, IRTpars = TRUE, CI = 0.90) # 90% CI instead of 95% CI
coef(fitted_blis, IRTpars = TRUE, printSE = TRUE) # SE instead of CI
```

---

# gDiscrim *Compute generalized item discrimination*

---

## Description

Generalized version of discrimination index ULI. The function enumerates the ability of an 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. You can also manually supply the maximal and minimal scores when the theoretical range of item score is known. Note that if the *observed* item range is zero NaN is returned.

## 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. |
| ... | Arguments passed on to base::findInterval<br>rightmost.closed logical; if true, the rightmost interval, vec[N-1] .. vec[N] is treated as *closed*, see below. |

`all.inside` logical; if true, the returned indices are coerced into `1,...,N-1`, i.e., `0` is mapped to `1` and `N` to `N-1`.

`left.open` logical; if true all the intervals are open at left and closed at right; in the formulas below, $\leq$ should be swapped with $<$ (and $>$ with $\geq$), and `rightmost.closed` means 'leftmost is closed'. This may be useful, e.g., in survival analysis computations.

## 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
Institute of Computer Science of the Czech Academy of Sciences

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>

Jan Netik
Institute of Computer Science of the Czech Academy of Sciences
<netik@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

```
# 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]
```

## `get_orig_levels` *Get Original Levels from a Fitted BLIS model*

### Description

Just a simple accessor to original levels and correct key stored in fitted BLIS model.

### Usage

```
get_orig_levels(object)
```

### Arguments

| | |
|---|---|
| `object` | *object of class BlisClass*, model fitted via `fit_blis()` or `blis()`. |

### Value

*list* of the original levels and correct key. Key is stored as an attribute `key` for every individual item.

### See Also

Other BLIS/BLIRT related: `BlisClass-class`, `coef,BlisClass-method`, `fit_blis()`, `nominal_to_int()`, `obtain_nrm_def()`, `print.blis_coefs()`

### Examples

```
fit <- fit_blis(HCItest[, 1:20], HCIkey)
get_orig_levels(fit)
```

## ggWrightMap *Plot person-item map (Wright map) using* ggplot2

### Description

This function allows to generate Wright map (also called person-item map) using ggplot() function from the **ggplot2** package. Wright map is used to jointly display histogram of abilities (or other measured trait) and item difficulty parameters. Function takes pre-estimated parameter estimates, such as those obtained from an 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

| | |
|---|---|
| theta | numeric: vector of ability estimates. |
| b | numeric: vector of difficulty estimates. |
| binwidth | numeric: the width of the bins of histogram. |
| color | character: color of histogram. |
| size | text size in pts. |
| item.names | names of items to be displayed. |
| ylab.theta | character: description of y-axis for the histogram. |
| ylab.b | character: description of y-axis for the plot of difficulty estimates. |
| rel_widths | 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
<netik@cs.cas.cz>

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)

# 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)
)
```

## GMAT *Dichotomous dataset based on GMAT with the same total score distribution for groups.*

### Description

The GMAT is a generated dataset based on parameters from Graduate Management Admission Test (GMAT, Kingston et al., 1985). First two items were considered to function differently in uniform and non-uniform way respectively. The dataset represents responses of 2,000 subjects to multiple-choice test of 20 items. A correct answer is coded as 1 and incorrect answer as 0. The column group represents group membership, where 0 indicates reference group and 1 indicates focal group. Groups are the same size (i.e. 1,000 per group). The distributions of total scores (sum of correct answers) are the same for both reference and focal group (Martinkova et al., 2017). The column criterion represents generated continuous variable which is intended to be predicted by test.

### Usage

```
GMAT
```

## Format

A `GMAT` data frame consists of 2,000 observations on the following 22 variables:

**Item1-Item20** dichotomously scored items of the test

**group** group membership vector, `"0"` reference group, `"1"` focal group

**criterion** continuous critetion intended to be predicted by test

## Author(s)

Adela Hladka (nee Drabinova)
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>`

## Source

Reexport from `difNLR` package.

## References

Kingston, N., Leary, L., & Wightman, L. (1985). An exploratory study of the applicability of item response theory methods to the Graduate Management Admission Test. ETS Research Report Series, 1985(2): 1–64.

Martinkova, P., Drabinova, A., Liaw, Y. L., Sanders, E. A., McFarland, J. L., & Price, R. M. (2017). Checking equity: Why differential item functioning analysis should be a routine part of developing conceptual assessments. CBE–Life Sciences Education, 16(2), rm2, doi:10.1187/cbe.16100307.

---

# `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 identificator whether students plan to major in life sciences.

## Usage

```
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** Identificator whether student plans to major in the life sciences.

**total** Total 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

### See Also

HCItest for HCI multiple-choice dataset
HCIkey for key of correct answers for HCI
HCIdata for HCI full dataset
HCIlong for HCI in a long format
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

```
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.

**yearc5** Study year.

**minority** Minority membership, `"maj"` majority, `"min"` Black/Hispanic 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

## See Also

HCI for HCI dichotomous dataset
HCItest for HCI multiple-choice dataset
HCIkey for key of correct answers for HCI
HCIlong for HCI in a long format
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

```
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

### See Also

HCI for HCI dichotomous dataset
HCItest for HCI multiple-choice dataset
HCIkey for key of correct answers for HCI
HCIdata for HCI full dataset
HCIlong for HCI in a long format
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

```
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

### See Also

HCI for HCI dichotomous dataset
HCItest for HCI multiple-choice dataset
HCIdata for HCI full dataset
HCIlong for HCI in a long format
HCIgrads for HCI dataset of graduate students
HCIprepost for HCI pretest and posttest scores
HCItestretest for HCI test-retest dataset

HCIlong *Homeostasis Concept Inventory in a long format*

## Description

HCIlong 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 (**in a long format**), vector of gender membership and identificator whether students plan to major in life sciences.

## Usage

```
HCIlong
```

## Format

HCIlong is a data.frame consisting of 13,020 rows and 5 variables.

**id** Row number of the original observation in a wide format.

**item** Name of the item the rating is for.

**rating** Response to the item.

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

**major** Identificator whether student plans to major in the life sciences.

**total** Total score

**zscore** Standardized total score (Z-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

## See Also

HCI for HCI dichotomous dataset (in a wide format)
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

## 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

```
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

### See Also

HCI for HCI dichotomous dataset
HCItest for HCI multiple-choice dataset
HCIkey for key of correct answers for HCI
HCIdata for HCI full dataset
HCIlong for HCI in a long format
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 containts 20 items, vector of gender membership and identificator whether students plan to major in life sciences.

## Usage

```
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** Identificator 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

## See Also

HCI for HCI dichotomous dataset
HCIkey for key of correct answers for HCI
HCIdata for HCI full dataset
HCIlong for HCI in a long format
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

```
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

### See Also

HCI for HCI dichotomous dataset
HCItest for HCI multiple-choice dataset
HCIkey for key of correct answers for HCI
HCIdata for HCI full dataset
HCIlong for HCI in a long format
HCIgrads for HCI dataset of graduate students
HCIprepost for HCI pretest and posttest scores

## `HeightInventory` *Height inventory dataset*

### Description

`HeightInventory` dataset consists of the responses of 4,885 respondents (1479 males, 3406 females) to a Height Inventory (Rečka, 2018). It contains 26 ordinal items of self-perceived height rated on a scale `"1"` strongly disagree, `"2"` disagree, `"3"` agree, `"4"` strongly agree, vector of self-reported heights (in centimeters), and vector of gender membership. Total score is included as the last variable, total score is NA for respondents who missed any item.

### Usage

```
HeightInventory
```

### Format

`HeightInventory` is a `data.frame` consisting of 4,885 observations on the 28 variables. First 26 variables are responses on scale `"1"` strongly disagree, `"2"` disagree, `"3"` agree, `"4"` strongly agree. Items 14 - 26 were reverse-coded, so that all items are scored in the same direction. Names of these items start with `"R-"`. Original item number and English wording is provided below.

**ShortTrousers** 1. A lot of trousers are too short for me.

**TallerThanM** 2. I am taller than men of my age.

**TallerThanF** 3. I am taller than women of my age.

**HeightForBasketball** 4. I have an appropriate height for playing basketball or volleyball.

**AskMeToReach** 5. Other people sometimes ask me to reach something for them.

**CommentsTall** 6. I am used to hearing comments about how tall I am.

**ConcertObstructs** 7. At concerts, my stature usually obstructs other people’s views.

**ShortBed** 8. Ordinary beds are too short for me.

**TopShelfEasy** 9. I can easily take wares from top shelves at a store.

**CrowdViewComf** 10. In a crowd of people, I still have a comfortable view.

**ShortBlanket** 11. Blankets and bedspreads rarely cover me completely.

**BendToHug** 12. When I want to hug someone, I usually need to bend over.

**CarefulHead** 13. I must often be careful to avoid bumping my head against a doorjamb or a low ceiling.

**R-SmallerThanM** 14. I am smaller than men of my age. (reversed)

**R-StoolNeeded** 15. I often need a stool to reach something other people could reach without one. (reversed)

**R-PlayDwarf** 16. I could play a dwarf. (reversed)

**R-SmallerThanW** 17. I am smaller than women of my age. (reversed)

**R-NoticeSmall** 18. One of the first things people notice about me is how small I am. (reversed)

**R-OnTipToes** 19. I often need to stand on the tip of my toes to get a better view. (reversed)

**R-ClothChildSize** 20. When I buy clothes, children's sizes often fit me well. (reversed)

**R-BusLegsEnoughSpace** 21. I have enough room for my legs when traveling by bus. (reversed)

**R-FasterWalk** 22. I often need to walk faster than I'm used to in order to keep pace with taller people. (reversed)

**R-AgeUnderestim** 23. Because of my smaller stature, people underestimate my age. (reversed)

**R-WishLowerChair** 24. It would be more comfortable for me if chairs were made lower. (reversed)

**R-UpwardLook** 25. When talking to other adults, I have to look upwards if I want to meet their eyes. (reversed)

**R-MirrorTooHigh** 26. Some mirrors are placed so high up that I have to crane my neck to use them. (reversed)

**gender** Gender membership, `"M"` males, `"F"` females.

**HeightCM** Self-reported height in centimeters.

**total** Total score.

## Note

Thanks to Karel Rečka and Hynek Cígler for sharing this dataset.

## References

Rečka, K. (2018). Height and Weight Inventory. Brno, Masaryk University: Unpublished Master's thesis

---

# `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% used).

`VarID` variance due to 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
<netik@cs.cas.cz>

### 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

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

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

## ItemAnalysis *Compute traditional item analysis indices*

### Description

Computes various traditional item analysis indices including difficulty, discrimination and item validity. For ordinal items, the function returns scaled values for some of the indices. See the details below.

### Usage

```
ItemAnalysis(
  Data,
  minscore = NULL,
  maxscore = NULL,
  cutscore = NULL,
  criterion = NULL,
  k = NULL,
  l = NULL,
  u = NULL,
  bin = "deprecated"
)
```

### Arguments

`Data` *matrix* or *data.frame* of items to be examined. Rows represent respondents, columns represent items.

`minscore`, `maxscore`
*integer*, theoretical minimal/maximal score. If not provided, these are computed on observed data. Automatically recycled to the number of columns of the data.

`cutscore` *integer* If provided, the input data are binarized accordingly. Automatically recycled to the number of columns of the data.

`criterion` vector of criterion values.

`k`, `l`, `u` Arguments passed on to `gDiscrim()`. Provide these if you want to compute generalized upper-lower index along with a standard ULI (using `k` = 3, `l` = 1, `u` = 3), which is provided by default.

`bin` *deprecated*, use `cutscore` instead. See the **Details**.

## Details

For calculation of generalized ULI index, it is possible to specify a custom 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 `cutscore` is provided, item analysis is conducted on binarized data; values greater or equal to cut-score are set to `1`, other values are set to `0`. Both the `minscore` and `maxscore` arguments are then ingored and set to 0 and 1, respectively.

## Value

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.

`Cut.score` cut-score specified in `cutscore`.

`obs.min` observed minimal score.

`Min.score` minimal score specified in `minscore`; if not provided, observed minimal score.

`obs.max` observed maximal score.

`Max.score` maximal score specified in `maxscore`; if not provided, observed maximal score.

`Prop.max.score` proportion of maximal scores.

`RIT` item-total correlation (correlation between item score and overall test score).

`RIR` item-rest correlation (correlation between item score and overall test score without the given item).

`ULI` upper-lower index using the standard parameters (3 groups, comparing 1st and 3rd).

`Corr.criterion` correlation between item score and criterion `criterion`.

`gULI` generalized ULI. `NA` when the arguments `k`, `l`, and `u` were not provided.

`Alpha.drop` Cronbach's alpha without given item.

`Index.rel` Gulliksen's (1950) item reliability index.

`Index.val` Gulliksen's (1950) item validity index.

`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()` for further details.

## 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
`<netik@cs.cas.cz>`

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

Gulliksen, H. (1950). *Theory of mental tests.* John Wiley & Sons Inc. https://doi.org/10.1037/13240-000

## See Also

`DDplot()`, `gDiscrim()`, `recode_nr()`

## Examples

```
## Not run:
# 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
))

## End(Not run)
```

## LearningToLearn *Dichotomous dataset of learning to learn test*

### 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 LtL 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

```
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 LtL, 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 LtL in Grade 6.

**score_9_subtest1–score_9_subtest7** Scores of respective cognitive subtest (1–7) of LtL 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")

## Source

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

---

# MSATB *Dichotomous dataset of Medical School Admission Test in Biology.*

---

## Description

The MSATB dataset consists of the responses of 1,407 subjects (484 males, 923 females) to admission test to medical school in the Czech republic. It contains 20 selected items from original test while first item was previously detected as differently functioning (Vlckova, 2014). A correct answer is coded as 1 and incorrect answer as 0. The column gender represents gender of students, where 0 indicates males (reference group) and 1 indicates females (focal group).

## Usage

```
MSATB
```

## Format

A MSATB data frame consists of 1,407 observations on the following 21 variables:

**Item** dichotomously scored items of the test

**gender** gender of respondents, "0" males, "1" females

## Author(s)

Adela Hladka (nee Drabinova)
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>

### Source

Reexport from `difNLR` package.

### References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Vlckova, K. (2014). Test and item fairness. Master's thesis. Faculty of Mathematics and Physics, Charles University.

---

## `MSclinical` *Clinical outcomes in multiple sclerosis patients dataset*

---

### Description

The `MSclinical` dataset contains clinical measures on multiple sclerosis patients.

### Usage

```
MSclinical
```

### 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

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 Center for Scientific Review (CSR) of the National Institutes of Health (NIH) 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

```
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

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

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()`

---

## `nominal_to_int` *Turn nominal (factor) data to integers, keep original levels with a key of correct responses alongside*

---

### Description

Convert a `data.frame` or `tibble` with factor variables (items) to integers, keeping the original factor levels (i.e. response categories) and correct answers (stored as an `key` attribute of each item) alongside.

### Usage

```
nominal_to_int(Data, key)
```

### Arguments

| | |
|---|---|
| `Data` | *data.frame* or *tibble* with all columns being factors. Support for *matrix* is limited and behavior not guaranteed. |
| `key` | A single-column `data.frame`, (**not** matrix) `tibble` or - preferably - a factor vector of levels considered as correct responses. |

### Details

Fitting a nominal model using `mirt::mirt()` package requires the dataset to consist only of integers, *arbitrarily* representing the response categories. You can convert your dataset to integers on your own in that case.

On the other hand, BLIS model (and thus also the BLIRT parametrization) further requires the information of correct item response category. On top of that, the same information is leveraged when fitting a `mirt` model that conserves the "directionality" of estimated latent ability (using a model definition from `obtain_nrm_def()`). In these cases, you are recommended to use `nominal_to_int()` (note that `fit_blis()` and `blis()` does this internally). Note also that fitted BLIS model (of class BlisClass) stores the original levels with correct answer key in its `orig_levels` slot, accessible by a user via `get_orig_levels()`.

### Value

*List* of original levels with logical attribute `key`, which stores the information on which response (level) is considered correct. *Note that levels not used in the original data are dropped.*

### See Also

Other BLIS/BLIRT related: `BlisClass-class`, `coef,BlisClass-method`, `fit_blis()`, `get_orig_levels()`, `obtain_nrm_def()`, `print.blis_coefs()`

---

## `obtain_nrm_def` *Obtain model definition for `mirt`'s nominal model taking in account the key of correct answers*

---

### Description

Standard `mirt` model with `itemtype = "nominal"` puts the identification constrains on the item response category slopes such as $ak_0 = 0$ and $ak_{(K-1)} = (K-1)$, freely estimating the rest.

While nominal item responses are unordered by definition, it is often the case that one of the item response categories is correct and the respondents endorsing this category "naturally" possess a higher latent ability. Use this function to obtain model definition where the correct response category $k_c$ for item $i$ with $K$ possible response categories translates to constrains $ak_{k_c} = (K-1)$ and $ak_{k_{d1}} = 0$, with $k_{d1}$ being the first incorrect response category (i.e. the first distractor).

### Usage

```
obtain_nrm_def(data_with_key, ...)
```

### Arguments

| | |
|---|---|
| `data_with_key` | The output of `nominal_to_int()`. |
| `...` | arguments passed onto `mirt::mirt()`. No practical use for now. |

### Value

A `data.frame` with the starting values, parameter numbers, estimation constrains etc. Pass it as `pars` argument of `mirt::mirt()`.

### See Also

Other BLIS/BLIRT related: `BlisClass-class`, `coef,BlisClass-method`, `fit_blis()`, `get_orig_levels()`, `nominal_to_int()`, `print.blis_coefs()`

### Examples

```
library(mirt)

# convert nominal data to integers and the original labels with correct answers
data_with_key <- nominal_to_int(HCItest[, 1:20], HCIkey)

# build model definition for {mirt} using the returned list from above
nrm_def <- obtain_nrm_def(data_with_key)

# fit the nominal model using the obtained model definition in `pars` argument
fit <- mirt(data_with_key$Data, 1, "nominal", pars = nrm_def)
```

## plot.sia_parallel *Plot Method for Parallel Analysis Output*

### Description

You can call this method to plot an existing object resulting from `fa_paralell()` 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 horizonal 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

`VGAM::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` *Plot cumulative and category probabilities of cumulative logit model*

### 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

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

### 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

VGAM::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
`<hladka@cs.cas.cz>`

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

### See Also

`difR::itemParEst()`, `difR::difLord()`, `difR::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 the same length as the number of observations in Data.

## 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

difR::difLogistic(), ggplot2::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()

## Examples

```
Data <- dataMedicaltest[, 1:100]
DataBin <- dataMedical[, 1:100]
key <- dataMedicalkey

# distractor plot for items 48, 57 and 32 displaying distractors only
# correct answer B does not function well:
plotDistractorAnalysis(Data, key, item = 48, multiple.answers = FALSE)

# all options function well, thus the whole item discriminates well:
plotDistractorAnalysis(Data, key, item = 57, multiple.answers = FALSE)

# functions well, thus the whole item discriminates well:
plotDistractorAnalysis(Data, key, item = 32, multiple.answers = FALSE)

## Not run:
# distractor plot for items 48, 57 and 32 displaying all combinations
plotDistractorAnalysis(Data, key, item = c(48, 57, 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, 96, 10))

## End(Not run)
```

## `plotMultinomial` *Plot category probabilities of multinomial model*

### Description

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

### Usage

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

### Arguments

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

### 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

nnet::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::ggplot(), providing a high customisability using "the grammar of graphics" (see the examples below).

### Usage

```
plot_corr(
  Data,
  cor = c("polychoric", "tetrachoric", "pearson", "spearman", "none"),
  clust_method = "none",
  n_clust = 0L,
  shape = c("circle", "square"),
  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 `polychoric`, `tetrachoric`, `pearson`, `spearman`, or `none` (in case you provide the correlation matrix as `Data`).

`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 (the default), no cluster are outlined, but items still do get sorted according to `clust_method` (if not set to `none`).

`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 `psych::polychoric`

`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 underdetermined 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.

`std.err` std.err=FALSE does not report the standard errors (faster) deprecated 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

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
Institute of Computer Science of the Czech Academy of Sciences
<netik@cs.cas.cz>

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 = "tetrachoric", 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.D2", 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 = "polychoric", clust_method = "complete", shape = "square") +
  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)
```

## print.blis_coefs *Print method for BLIS coefficients*

### Description

Print method for BLIS coefficients

### Usage

```
## S3 method for class 'blis_coefs'
print(x, digits = 3, ...)
```

### Arguments

| | |
|---|---|
| x | result of `coef()`. |
| digits | *integer*, number of digits to show in the output. Note that printed object are still an original list, which does not round any value (it is returned invisibly). |
| ... | Additional arguments passed on to `print()`. |

### See Also

Other BLIS/BLIRT related: `BlisClass-class`, `coef,BlisClass-method`, `fit_blis()`, `get_orig_levels()`, `nominal_to_int()`, `obtain_nrm_def()`

## recode_nr *Recognize and recode not-reached responses*

### 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
Institute of Computer Science of the Czech Academy of Sciences
<netik@cs.cas.cz>

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

### See Also

ItemAnalysis()

### Examples

```
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)
```

---

## ShinyItemAnalysis_options

*Options consulted by ShinyItemAnalysis*

---

### Description

The package and interactive {shiny} app consult several options that you can easily set via options(). Moreover, there is some behavior that can be changed through environment variables.

### Options

Options are set with options(<option> = <value>).

- sia.disable_modules: You can completely disable SIA modules by setting this to TRUE.
- sia.modules_repo: This is the URL for a CRAN-like repository that the app uses to retrieve information about available module packages.
- sia.offer_modules: If set to TRUE (the default), calling run_app() will check for the available SIA modules at the official repository and offer to install those module packages that are not installed yet.

### Environment variables

You can set this variable system-wide or use R or project-wise .Renviron file. For more details, please navigate to the R documentation.

- SIA_MODULES_DEBUG: Setting this to TRUE provides a verbose description of SIA modules-related processes. Useful only for debugging purposes.

## 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" IDE). User is then free to use the R 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, ...)

run_app(background = TRUE, ...)
```

### Arguments

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

`...` Arguments passed on to utils::install.packages

- `lib` character vector giving the library directories where to install the packages. Recycled as needed. If missing, defaults to the first element of .libPaths().
- `repos` character vector, the base URL(s) of the repositories to use, e.g., the URL of a CRAN mirror such as "https://cloud.r-project.org". For more details on supported URL schemes see url.
  Can be NULL to install from local files, directories or URLs: this will be inferred by extension from pkgs if of length one.
- `contriburl` URL(s) of the contrib sections of the repositories. Use this argument if your repository mirror is incomplete, e.g., because you mirrored only the 'contrib' section, or only have binary packages. Overrides argument repos. Incompatible with type = "both".
- `method` download method, see download.file. Unused if a non-NULL available is supplied.
- `available` a matrix as returned by available.packages listing packages available at the repositories, or NULL when the function makes an internal call to available.packages. Incompatible with type = "both".
- `destdir` directory where downloaded packages are stored. If it is NULL (the default) a subdirectory downloaded_packages of the session temporary directory will be used (and the files will be deleted at the end of the session).
- `dependencies` logical indicating whether to also install uninstalled packages which these packages depend on/link to/import/suggest (and so on recursively). Not used if repos = NULL. Can also be a character vector, a subset of c("Depends", "Imports", "LinkingTo", "Suggests", "Enhances").

Only supported if `lib` is of length one (or missing), so it is unambiguous where to install the dependent packages. If this is not the case it is ignored, with a warning.
The default, `NA`, means `c("Depends", "Imports", "LinkingTo")`.
`TRUE` means to use `c("Depends", "Imports", "LinkingTo", "Suggests")` for `pkgs` and `c("Depends", "Imports", "LinkingTo")` for added dependencies: this installs all the packages needed to run `pkgs`, their examples, tests and vignettes (if the package author specified them correctly).
In all of these, `"LinkingTo"` is omitted for binary packages.

`type` character, indicating the type of package to download and install. Will be `"source"` except on Windows and some macOS builds: see the section on 'Binary packages' for those.

`configure.args` (Used only for source installs.) A character vector or a named list. If a character vector with no names is supplied, the elements are concatenated into a single string (separated by a space) and used as the value for the '`--configure-args`' flag in the call to `R CMD INSTALL`. If the character vector has names these are assumed to identify values for '`--configure-args`' for individual packages. This allows one to specify settings for an entire collection of packages which will be used if any of those packages are to be installed. (These settings can therefore be re-used and act as default settings.)
A named list can be used also to the same effect, and that allows multi-element character strings for each package which are concatenated to a single string to be used as the value for '`--configure-args`'.

`configure.vars` (Used only for source installs.) Analogous to `configure.args` for flag '`--configure-vars`', which is used to set environment variables for the `configure` run.

`clean` a logical value indicating whether to add the '`--clean`' flag to the call to `R CMD INSTALL`. This is sometimes used to perform additional operations at the end of the package installation in addition to removing intermediate files.

`Ncpus` the number of parallel processes to use for a parallel install of more than one source package. Values greater than one are supported if the `make` command specified by `Sys.getenv("MAKE", "make")` accepts argument `-k -j` *Ncpus*.

`verbose` a logical indicating if some "progress report" should be given.

`INSTALL_opts` an optional character vector of additional option(s) to be passed to `R CMD INSTALL` for a source package install. E.g., `c("--html", "--no-multiarch", "--no-test-load")`.
Can also be a named list of character vectors to be used as additional options, with names the respective package names.

`quiet` logical: if true, reduce the amount of output. This is *not* passed to `available.packages()` in case that is called, on purpose.

`keep_outputs` a logical: if true, keep the outputs from installing source packages in the current working directory, with the names of the output files the package names with '`.out`' appended. Alternatively, a character string giving the directory in which to save the outputs. Ignored when installing from local files.

### 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>

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

### Examples

```
## Not run:
startShinyItemAnalysis()
startShinyItemAnalysis(background = FALSE)

## End(Not run)
```

---

## TestAnxietyCor *Correlation matrix for the test anxiety dataset*

---

### Description

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

### Usage

```
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

| | |
|---|---|
| `base_size` | base font size |
| `base_family` | base font family |

### See Also

`ggplot2::theme()`

### 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

∗ **BLIS/BLIRT related**
  BlisClass-class, 11
  coef,BlisClass-method, 13
  fit_blis, 28
  get_orig_levels, 38
  nominal_to_int, 60
  obtain_nrm_def, 61
  print.blis_coefs, 75
∗ **datasets**
  AIBS, 5
  Anxiety, 7
  AttitudesExpulsion, 8
  BFI2, 10
  CLoSEread6, 12
  CZmatura, 14
  CZmaturaS, 15
  dataMedical, 16
  dataMedicalgraded, 17
  dataMedicalkey, 18
  dataMedicaltest, 18
  EPIA, 24
  GMAT, 40
  HCI, 41
  HCIdata, 42
  HCIgrads, 44
  HCIkey, 45
  HCIlong, 46
  HCIprepost, 47
  HCItest, 48
  HCItestretest, 49
  HeightInventory, 50
  LearningToLearn, 56
  MSATB, 57
  MSclinical, 58
  NIH, 59
  TestAnxietyCor, 80
.libPaths, *78*

AIBS, 5
AIBS(), *4*
Anxiety, 7
Anxiety(), *4*
AttitudesExpulsion, 8
AttitudesExpulsion(), *4*
available.packages, *78, 79*

base::findInterval, *36*
BFI2, 10
BFI2(), *4*
blis (fit_blis), 28
blis(), *4*, *61*
BlisClass, *11*, *13*, *35*, *38*, *61*
BlisClass-class, 11
bs, *33*

CLoSEread6, 12
CLoSEread6(), *4*
coef,BlisClass-method, *12*, 13, *35*
colors(), *73*
CZmatura, 14
CZmatura(), *4*, *15*
CZmaturaS, 15
CZmaturaS(), *4*, *15*

dataMedical, 16
dataMedical(), *4*, *17–19*
dataMedicalgraded, 17
dataMedicalgraded(), *4*, *16, 18, 19*
dataMedicalkey, 18
dataMedicalkey(), *4*, *16, 17*, *19*
dataMedicaltest, 18
dataMedicaltest(), *4*, *16–18*
DDplot, 19
DDplot(), *3*, *37*, *55*
difR::difLogistic(), *68*
difR::difLord(), *66*
difR::difRaju(), *66*
difR::itemParEst(), *66*
DistractorAnalysis, 22
DistractorAnalysis(), *3*, *70*

download.file, 78

EPIA, 24
extract.mirt, 34

fa_parallel, 25
fa_parallel(), 3
fit_blis, 12, 14, 28, 38, 61, 62, 75
fit_blis(), 61

gDiscrim, 36
gDiscrim(), 3, 21, 54, 55
get_orig_levels, 12, 14, 36, 38, 61, 62, 75
get_orig_levels(), 61
ggplot2::ggplot(), 21, 68, 72
ggplot2::theme(), 81
ggWrightMap, 39
ggWrightMap(), 4
GMAT, 40

HCI, 41, 43–49
HCI(), 4
HCIdata, 42, 42, 44–49
HCIdata(), 4
HCIgrads, 42, 43, 44, 45–49
HCIgrads(), 4
HCIkey, 42–44, 45, 46–49
HCIkey(), 4
HCIlong, 42–45, 46, 47–49
HCIlong(), 4
HCIprepost, 42–46, 47, 48, 49
HCIprepost(), 4
HCItest, 42–47, 48, 49
HCItest(), 4, 45
HCItestretest, 42–48, 49
HCItestretest(), 4
hclust(), 73
HeightInventory, 50
HeightInventory(), 4

ICCrestricted, 51
ICCrestricted(), 4, 7, 60
ItemAnalysis, 53
ItemAnalysis(), 4, 76

LearningToLearn, 56
LearningToLearn(), 4
lme4::bootMer(), 52

mirt.model, 31
mirt::coef,SingleGroupClass-method, 13
mirt::mirt, 28
mirt::mirt(), 61, 62
mirtCluster, 30, 34
MSATB, 57
MSATB(), 4
MSclinical, 58
MSclinical(), 4

NIH, 59
NIH(), 4
nnet::multinom(), 72
nominal_to_int, 12, 14, 36, 38, 60, 62, 75
nominal_to_int(), 61
ns, 33

obtain_nrm_def, 12, 14, 36, 38, 61, 61, 75
obtain_nrm_def(), 61
optim, 29
options(), 77

plot.sia_parallel, 62
plot_corr, 72
plot_corr(), 4
plotAdjacent, 63
plotAdjacent(), 4
plotCumulative, 64
plotCumulative(), 4
plotDIFirt, 65
plotDIFirt(), 4
plotDIFLogistic, 67
plotDIFLogistic(), 4
plotDistractorAnalysis, 69
plotDistractorAnalysis(), 3
plotMultinomial, 71
plotMultinomial(), 4
print.blis_coefs, 12, 14, 36, 38, 61, 62, 75
psych::fa(), 26
psych::polychoric, 26, 73
psych::psych(), 26

recode_nr, 76
recode_nr(), 4, 55
run_app (startShinyItemAnalysis), 78

ShinyItemAnalysis
(ShinyItemAnalysis-package), 3
ShinyItemAnalysis-package, 3
ShinyItemAnalysis_options, 77

startShinyItemAnalysis, 78
startShinyItemAnalysis(), *3*

TestAnxietyCor, 80
TestAnxietyCor(), *4*
the R documentation, *77*
theme_app, 81

url, *78*
utils::install.packages, *78*

VGAM::vglm(), *64, 65*

wald, *29*