

Package ‘rasch’

August 24, 2026

Type Package

Title Models and Diagnostics for Rasch Measurement Theory

Version 1.12.0

Description Fits models within Rasch Measurement Theory, whose defining properties include sufficiency and invariance (Rasch, 1960 <ISBN:9780226705538>; Andrich and Marais, 2019 <doi:10.1007/978-981-13-7496-8>). Models include the dichotomous Rasch, partial credit, rating scale, explanatory, many-facet and extended frame of reference models. Explanatory models include the linear logistic test model (Fischer, 1973 <doi:10.1016/0001-6918(73)90003-6>) and linear partial credit model (Fischer and Ponocny, 1994 <doi:10.1007/BF02295182>). Comparative judgement models, including explanatory object models, are available for dichotomous (Andrich, 1978 <doi:10.1177/014662167800200319>) and ordered pairwise responses (Tutz, 1986 <doi:10.1016/0022-2496(86)90034-9>). Item parameters for item-response models are estimated by pairwise conditional maximum likelihood (Zwinderman, 1995 <doi:10.1177/014662169501900406>), comparative judgement parameters by maximum likelihood, and person locations by weighted likelihood (Warm, 1989 <doi:10.1007/BF02294627>). Functions cover fit, targeting, reliability, dimensionality, local dependence, differential item functioning, equating and simulation. A 'shiny' application provides a graphical interface to the analyses.

URL <https://drjoshmcgrane.github.io/rasch/>,
<https://github.com/drjoshmcgrane/rasch>

BugReports <https://github.com/drjoshmcgrane/rasch/issues>

License MIT + file LICENSE

Encoding UTF-8

Imports stats, graphics, grDevices, utils, parallel, Rcpp

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), shiny, bslib, DT, bsicons, htmltools,
xml2, knitr, rmarkdown, eRm, sirt, psychotools, mirt,
ShinyItemAnalysis, callr, WrightMap

VignetteBuilder knitr
RoxygenNote 7.3.3
Config/testthat/edition 3
NeedsCompilation yes
Author Josh McGrane [aut, cre]
Maintainer Josh McGrane <drjoshmcgrane@gmail.com>
Repository CRAN
Date/Publication 2026-08-24 08:40:02 UTC

Contents

btl	4
btl_dif	7
btl_dimensionality	9
btl_efrm	10
btl_equate	13
btl_explanatory	15
btl_information	17
btl_next_pairs	18
btl_transitivity	20
chisq_detail	21
combine_items	21
compare_fits	22
ctt_table	24
dependence_magnitude	25
dif_anova	26
dif_contrasts	29
dif_posthoc	31
dif_size	33
dimensionality_magnitude	35
dimensionality_test	37
distractor_analysis	38
distractor_rescore	39
drop_items	40
equate_tests	41
explanatory_diagnostics	42
explanatory_test	43
fit_summary_table	44
frame_invariance	44
guttman_table	46
item_moments	47
judge_pair_surprise	48
judge_surprise	49
lr_test	50
pcml	51

pcml_pc	52
person_extrapolated	53
person_wle	54
plot_btl	55
plot_btl_categories	56
plot_btl_dependence	56
plot_btl_dim_map	57
plot_btl_equate	58
plot_btl_icc	59
plot_btl_judge_map	60
plot_btl_scree	61
plot_btl_targeting	61
plot_btl_transitivity	62
plot_btl_units	63
plot_catfreq	63
plot_ccc	64
plot_distractors	65
plot_equate	66
plot_facets	66
plot_frames	67
plot_guttman	68
plot_icc	69
plot_icc_frames	70
plot_item_map	71
plot_kidmap	71
plot_pca	72
plot_pca_biplot	73
plot_pcc	74
plot_person_fit	74
plot_pimap	75
plot_recovery	76
plot_resid_cor	76
plot_resid_dist	77
plot_scree	78
plot_tcc	79
plot_threshold_map	80
plot_threshold_prob	80
plot_tif	81
plot_wright	82
rack_data	83
rasch	84
rasch_efrm	87
rasch_explanatory	90
rasch_mfrm	92
relax_btl_explanatory	94
relax_explanatory	95
report_document	96
report_html	97

residual_correlations	97
residual_pca	98
resolve_dif	99
resolve_frames	100
run_app	101
save_item_plots	102
save_outputs	103
save_person_plots	104
score_table	105
simulate_btl	106
simulate_btl_efrm	107
simulate_efrm	109
simulate_mfrm	110
simulate_rasch	111
sim_apply	113
sim_recovery	114
sim_replicate	114
split_items	115
spread_test	116
tailored_analysis	117
targeting_table	119
test_information	119
threshold_index	120
wright_map	121

Index 123

btl

Fit comparative judgement models to paired comparisons

Description

Fits the Bradley–Terry–Luce model to dichotomous comparisons or its ordered-response extension (Tutz 1986). Object, judge, and pair fit are reported with an object separation index and design diagnostics.

Usage

```
btl(
  data,
  object_a,
  object_b,
  winner = NULL,
  response = NULL,
  margin = NULL,
  judge = NULL,
  count = NULL,
  order = NULL,
```

```

position = FALSE,
anchors = NULL,
ties = c("drop", "half", "error"),
thresholds = c("free", "pc"),
maxit = 60,
tol = 1e-08,
.object_design = NULL
)

```

Arguments

<code>data</code>	A data frame with one comparison per row.
<code>object_a, object_b</code>	Names of the columns holding the two objects compared.
<code>winner</code>	Name of the column holding the winner of each row: its value must equal one of the two objects. "tie" and "draw" mark ties. Ignored when response is supplied.
<code>response</code>	Optional ordered response favouring <code>object_a</code> over <code>object_b</code> : an ordered factor from least to greatest preference for <code>object_a</code> , or integer scores 0 . . m.
<code>margin</code>	Optional ordered margin-of-victory column, combined with <code>winner</code> to construct an orientation-invariant response.
<code>judge</code>	Optional name of a judge column; enables the judge fit table and clusters the sandwich standard errors by judge.
<code>count</code>	Optional name of a column of replication counts (a row standing for several identical comparisons).
<code>order</code>	Optional column giving each judge's comparison sequence; requires <code>judge</code> . See Details. Incompatible with <code>ties = "half"</code> .
<code>position</code>	If TRUE, estimate a first-presentation advantage, treating <code>object_a</code> as the first object in each comparison.
<code>anchors</code>	Optional named numeric vector of fixed object locations. Anchored objects have standard error zero and must not be boundary objects.
<code>ties</code>	How to treat ties in the dichotomous analysis: "drop" (default, removed with a note), "half" (half a win each way, a common pragmatic device – flagged in the notes because the halves are not independent Bernoulli trials), or "error". With polytomous responses, code ties as a middle category instead.
<code>thresholds</code>	"free" (default) estimates every symmetric threshold; "pc" retains only the symmetric spread component.
<code>maxit, tol</code>	Newton-Raphson iteration cap and convergence tolerance.
<code>.object_design</code>	Internal object-location design used by btl_explanatory .

Details

For objects a and b , the dichotomous model is

$$P(a \succ b) = \frac{\exp(\beta_a)}{\exp(\beta_a) + \exp(\beta_b)}.$$

This is the conditional form of the dichotomous Rasch model (Andrich 1978). For an ordered response $Y = 0, \dots, m$,

$$\log\{P(Y = r)/P(Y = r - 1)\} = \beta_a - \beta_b - \tau_r,$$

with thresholds constrained to be symmetric under reversal of presentation order. Two categories reproduce the dichotomous model.

Locations are identified by a sum-zero constraint unless anchors are supplied. The comparison graph must be connected, and the directed win graph must be strongly connected for all free locations to be finite (Ford 1957). Boundary objects are removed when this leaves an identified model; otherwise fitting stops.

Standard errors use the Godambe sandwich covariance. When judge is supplied, the covariance is clustered by judge. Clustered inference is withheld when there are fewer than ten judges, fewer than eight effective judges, or no residual cluster degrees of freedom. A caution is attached when the effective count is below 9.5 or one judge supplies more than 20 per cent of the comparisons.

Dichotomous data may be supplied as a winner, with ties dropped or divided equally between the two outcomes. Ordered data may instead be supplied directly as scores from 0 to m , or assembled from the winner and an ordered margin of victory. Plain factors are refused because alphabetical ordering can reverse the response scale. The "pc" threshold option retains the symmetric spread component, which can stabilise thin categories.

If comparison order is supplied, exposure and carry-over effects are estimated from each judge's preceding comparisons. The position term estimates a first-presentation effect. These coefficients enter the model jointly with the object locations and are reported in logits. The carry-over estimate and clustered SE remain descriptive below 30 judges; its probability is withheld because null calibration is mildly anti-conservative at smaller judge counts. Anchors fix nominated object locations and replace the sum-zero origin.

Value

A "rasch_btl" object. Principal components are objects, pairs, judges, the total pair-fit test, `osi`, `loglik`, composite-likelihood information `cl`, convergence details, and notes. Ordered-response fits also contain thresholds, `m`, and categories. Fits using order contain dependence and `dependence_data`.

References

- Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39, 324–345.
- Luce, R. D. (1959). *Individual Choice Behavior*. Wiley.
- Andrich, D. (1978). Relationships between the Thurstone and Rasch approaches to item scaling. *Applied Psychological Measurement*, 2, 451–462.
- Tutz, G. (1986). Bradley-Terry-Luce models with an ordered response. *Journal of Mathematical Psychology*, 30(3), 306–316.
- Agresti, A. (1992). Analysis of ordinal paired comparison data. *Journal of the Royal Statistical Society C*, 41(2), 287–297.
- Davidson, R. R. (1970). On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329), 317–328.

Ford, L. R. (1957). Solution of a ranking problem from binary comparisons. *American Mathematical Monthly*, 64(8), 28–33.

Davidson, R. R. and Beaver, R. J. (1977). On extending the Bradley-Terry model to incorporate within-pair order effects. *Biometrics*, 33(4), 693–702.

See Also

[btl_dif](#), [btl_efrm](#), [btl_information](#), [btl_transitivity](#), and [simulate_btl](#).

Examples

```
set.seed(1)
beta <- c(A = -1, B = -0.3, C = 0.4, D = 0.9)
pairs <- t(combn(names(beta), 2))
d <- data.frame(a = rep(pairs[, 1], each = 30),
               b = rep(pairs[, 2], each = 30))
p <- plogis(beta[d$a] - beta[d$b])
d$win <- ifelse(runif(nrow(d)) < p, d$a, d$b)
btl(d, object_a = "a", object_b = "b", winner = "win")
```

btl_dif

DIF analysis for paired comparisons

Description

Tests whether object locations differ across groups of judges. Several judge factors can be fitted jointly, with optional factor-by-factor interactions. Uniform DIF is a judge-factor effect; non-uniform DIF is its interaction with opponent-strength band.

Usage

```
btl_dif(
  fit,
  factors,
  objects = NULL,
  effects = c("main", "factorial"),
  p_adjust = "holm",
  alpha = 0.05,
  flag_logits = 0.5,
  min_n = 20,
  maxit = 60,
  tol = 1e-08
)
```

Arguments

<code>fit</code>	An ordinary paired-comparison fit from <code>btl</code> .
<code>factors</code>	One judge factor, or a named list containing several. Each factor may have one value per comparison row or be a vector named by judge.
<code>objects</code>	Objects to test; all by default.
<code>effects</code>	"main" (default) models several factors additively (each factor's main effect and its band interaction); "factorial" also crosses the factors with one another.
<code>p_adjust</code>	Multiplicity adjustment over all object-by-term tests; the resolved-size probabilities are adjusted separately in one pool over all objects, terms, and cell pairs.
<code>alpha</code>	Significance level for adjusted probabilities.
<code>flag_logits</code>	Absolute resolved difference flagged as practically significant.
<code>min_n</code>	Term cells with fewer comparisons involving the object are dropped from its resolution, with a note.
<code>maxit, tol</code>	Newton controls for the resolution refits.

Details

Judges are the independent units. For each object, oriented residuals are aggregated to one weighted mean per judge and opponent band. A split-plot analysis then tests judge factors between judges and band effects within judges. Each factor level requires at least two judges. Confirmatory Wald tests are available only when the base fit supplies a valid judge-clustered covariance. The base paired-comparison calibration must have converged. BTL-EFRM fits are not accepted: the ordinary residual and resolution models do not contain the fitted panel and set units.

A significant uniform term is followed by a joint refit in which the object has one location per factor cell. Differences between these locations are reported in logits. A cell needs at least eight effective judges for pairwise inference; otherwise its location and differences remain descriptive. Pairwise tests use degrees of freedom based on the effective judges in the two cells. Higher-order terms supersede their component terms. Models fitted with order retain the exposure and carry-over effects in both the residual analysis and refit. Between-judge tests use HC3 covariance so unequal comparison workloads do not impose equal precision on judge means. Omnibus probabilities require at least eight judges and eight effective judges in every factor cell. Holm adjustment is the default; "BH" remains available for false-discovery-rate screening.

Objects are resolved one at a time against the common locations of the remaining objects. With DIF in several objects, this can induce compensating apparent DIF in invariant objects (Andrich and Hagquist 2012, 2015). An externally anchored object is not resolved: fixing each of its copies at the same anchor would define their difference as zero. Anchors on the other objects are retained in the joint refit.

Value

A list of class "rasch_btl_dif": summary (one row per object and group term with the uniform F, adjusted p and partial eta-squared – the term itself – the non-uniform ones – the term crossed with the opponent band – plus uniform_DIF, nonuniform_DIF and superseded flags); terms (the full per-object analysis-of-variance table, including its raw and effective judge support); levels (resolved location, SE, comparison count, judge count and effective judge count per object, term

and cell); sizes (per object, term and cell pair: difference in logits, judge support for both cells, SE, t, degrees of freedom, adjusted p, significance and practical flags); effects, factors, alpha, p_adjust, flag_logits, and notes. `summary_factors` retains the factor membership of each displayed term.

References

- Andrich, D., & Hagquist, C. (2012). Real and artificial differential item functioning. *Journal of Educational and Behavioral Statistics*, 37(3), 387-416.
- Dittrich, R., Hatzinger, R., & Katzenbeisser, W. (1998). Modelling the effect of subject-specific covariates in paired comparison studies with an application to university rankings. *Journal of the Royal Statistical Society C*, 47(4), 511-525.
- MacKinnon, J. G., & White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, 29(3), 305-325.

Examples

```
set.seed(1)
beta <- c(A = -1, B = -0.3, C = 0.4, D = 0.9)
pr <- t(combn(names(beta), 2))
d <- data.frame(a = rep(pr[, 1], each = 100), b = rep(pr[, 2], each = 100),
               judge = sample(sprintf("J%02d", 1:20), 600, TRUE))
shift <- ifelse(d$judge %in% sprintf("J%02d", 1:10) & d$a == "C", 0.9,
               ifelse(d$judge %in% sprintf("J%02d", 1:10) & d$b == "C", -0.9, 0))
p <- plogis(beta[d$a] - beta[d$b] + shift)
d$win <- ifelse(runif(nrow(d)) < p, d$a, d$b)
f <- btl(d, "a", "b", winner = "win", judge = "judge")
grp <- setNames(rep(c("g1", "g2"), each = 10), sprintf("J%02d", 1:20))
btl_dif(f, grp, objects = "C")
```

btl_dimensionality	<i>Experimental residual dimensionality of paired comparisons</i>
--------------------	---

Description

Decomposes the skew-symmetric matrix of observed-minus-expected pair log-odds into Gower's (1977) rotational planes, or bimensions. A large leading bimension indicates a structured cycle in the residual comparisons. Its strength is compared with simulations from the fitted one-dimensional model using the observed comparison counts.

Usage

```
btl_dimensionality(fit, reps = 200L)
```

Arguments

fit	A paired-comparison fit from btl .
reps	Model-simulated replicates for the noise reference.

Details

This is an experimental diagnostic. The reference is conditional on the fitted point estimates because the model is not re-estimated in each replicate. Ordered-response fits use the same points-proportion residual in the data and simulations. Fits with exposure or carry-over effects simulate those effects through each judge's observed sequence. The fitted model must have converged.

A categorical result is withheld if any object pair is unobserved. It is also withheld when every judge receives essentially the same comparison sequence and an order effect is fitted, because order and residual structure are then confounded.

Value

A list of class "rasch_btl_dim": dimensions (per dimension: strength and share of residual size; the reference mean, 95th percentile, and the clears-the-reference flag are reported for the leading dimension and NA for the rest); coords (each object's position in the leading dimension plane, for the residual map); leading_structured (whether dimension 1 clears its reference); residual_matrix; and notes.

References

Gower, J. C. (1977). The analysis of asymmetry and orthogonality. In J. R. Barra et al. (Eds.), *Recent Developments in Statistics* (pp. 109-123). North-Holland.

Examples

```
set.seed(1); objs <- LETTERS[1:6]; beta <- setNames(seq(-1.5, 1.5, len = 6), objs)
pr <- t(utils::combn(objs, 2))
d <- data.frame(a = rep(pr[, 1], each = 30), b = rep(pr[, 2], each = 30))
d$win <- ifelse(runif(nrow(d)) < plogis(beta[d$a] - beta[d$b]), d$a, d$b)
btl_dimensionality(btl(d, "a", "b", "win"), reps = 20)
```

btl_efrm

Fit the extended frame of reference model for paired comparisons

Description

Fits paired comparisons when judges belong to panels and objects belong to linked sets whose units or origins can differ. It combines the Bradley–Terry–Luce model with Humphry's extended frame of reference structure.

Usage

```
btl_efrm(
  data,
  object_a,
  object_b,
  winner,
  judge,
```

```

panels,
object_sets,
response = NULL,
ties = c("drop", "error"),
min_link = 20,
se_method = c("judge_bootstrap", "bootstrap", "conditional"),
boot_reps = 200,
workers = 4L,
seed = NULL,
progress = NULL,
cancel = NULL,
maxit = 60,
tol = 1e-08
)

```

Arguments

<code>data</code>	A data frame with one comparison per row.
<code>object_a</code> , <code>object_b</code>	Names of the columns holding the two compared objects.
<code>winner</code>	Name of the winner column. A value must match one of the two objects in that row. "tie" and "draw" mark ties; other values are treated as missing.
<code>judge</code>	Name of the judge column (clusters the stage-one standard errors and defines the panels when <code>panels</code> is a judge attribute).
<code>panels</code>	Either the name of a judge-attribute column in <code>data</code> or a named vector mapping judge to panel.
<code>object_sets</code>	A named list mapping set names to character vectors of object names; every compared object must belong to exactly one set.
<code>response</code>	Not supported: this first implementation fits dichotomous winner data only. Supplying it raises an informative error.
<code>ties</code>	"drop" (default, removed with a note) or "error".
<code>min_link</code>	Minimum number of cross-set comparisons a set pair must supply to be used for linking; sets not reachable from the reference set through sufficient cross-set pairs raise an error.
<code>se_method</code>	Method used for standard errors. The default, "judge_bootstrap", resamples judges within panels and retains dependence among a judge's comparisons. "bootstrap" instead draws independent outcomes from fitted probabilities. Both stages are refitted. "conditional" uses analytic stage-one standard errors for beta and phi, and inverse observed information for alpha and kappa conditional on the stage-one estimates. It is faster, but does not propagate stage-one uncertainty into the linking parameters; unit probabilities and omnibus tests are therefore withheld.
<code>boot_reps</code>	Number of replicates for <code>se_method = "bootstrap"</code> or <code>"judge_bootstrap"</code> ; at least 30 are required.
<code>workers</code>	Number of judge-bootstrap workers. The default is four, reduced when the system limit is lower. The parametric bootstrap remains serial because its refits are inexpensive.

seed	Optional bootstrap seed. The caller's random-number state is restored when estimation finishes.
progress	Optional function called as <code>progress(stage, current, total)</code> during estimation.
cancel	Optional zero-argument function checked between bootstrap batches. Returning TRUE stops with a <code>rasch_cancelled</code> condition.
maxit, tol	Newton iteration cap and convergence tolerance.

Details

For object k in set s , let

$$v_k = \alpha_s \beta_k + \kappa_s,$$

where β_k is its within-set location, $\alpha_s > 0$ is the set unit, and κ_s is the set origin. A comparison in panel g has logit

$$\phi_g(\beta_a - \beta_b)$$

for objects in the same set, and

$$\phi_g(v_a - v_b)$$

for objects in different sets. Cross-set comparisons identify the common scale. The first set fixes $\alpha = 1$ and $\kappa = 0$; panel units have geometric mean one.

Estimation has two stages. Within-set comparisons estimate object locations and panel-unit ratios. Weighted least squares reconciles the ratios over the panel-by-set linking graph. Cross-set comparisons then estimate the set units and origins. Unlike the person-by-item EFRM, this linking step uses only comparison outcomes and does not require a distribution of persons. The paired-comparison form is an extension of Humphry's model implemented in this package.

The default judge bootstrap resamples judges within panels and refits both stages. The parametric bootstrap draws independent outcomes from the fitted probabilities and uses normal and chi-square reference distributions. `se_method = "conditional"` uses analytic stage-one errors and conditions the linking errors on stage one; it is intended for preliminary inspection. Its unit probabilities and omnibus tests are withheld because it does not propagate stage-one uncertainty. Bootstrap failures and boundary estimates are reported in notes.

With one set, the model contains panel units only. With one set and one panel, it reduces to `btl`. Omnibus Wald tests provide inference for the unit families; individual contrasts are Holm-adjusted follow-ups. Judge-bootstrap probabilities require at least six judges and 5.5 effective judges in every contributing panel, and eight of each on a set link. The support is returned in `unit_support`; estimates remain descriptive when a probability is withheld.

Value

An object of class `"rasch_btl_efrm"`. It contains the object estimates, group- and set-unit tables, origin shifts, omnibus unit tests, unit-specific judge support, frame definitions, convergence information, and analysis notes.

References

- Andrich, D. (1978). Relationships between the Thurstone and Rasch approaches to item scaling. *Applied Psychological Measurement*, 2(3), 451–462.
- Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39, 324–345.
- David, H. A. (1988). *The Method of Paired Comparisons* (2nd ed.). Griffin.
- Humphry, S. M. (2005). Maintaining a common arbitrary unit in social measurement. PhD thesis, Murdoch University.
- Humphry, S. M. (2012). Item set discrimination and the unit in the Rasch model. *Journal of Applied Measurement*, 13(2), 165–180.
- Humphry, S. M. and Andrich, D. (2008). Understanding the unit in the Rasch model. *Journal of Applied Measurement*, 9(3), 249–264.
- Luce, R. D. (1959). *Individual Choice Behavior: A Theoretical Analysis*. Wiley.
- Thurstone, L. L. (1927). A law of comparative judgment. *Psychological Review*, 34, 273–286.

See Also

[btl](#), [rasch_efrm](#), [plot_btl_units](#), and [simulate_btl_efrm](#).

Examples

```
d <- simulate_btl_efrm(n_objects_per_set = 6, n_sets = 2, n_panels = 2,
                      set_units = c(1, 1.4), set_origins = c(0, 0.8),
                      seed = 1)
fit <- btl_efrm(d, "object_a", "object_b", winner = "winner",
               judge = "judge", panels = "panel",
               object_sets = attr(d, "truth")$object_sets,
               se_method = "conditional")
fit$alpha_table
```

btl_equate	<i>Equate two paired-comparison calibrations through their common objects</i>
------------	---

Description

Places two Bradley–Terry–Luce calibrations on a common origin using their shared objects, then tests the shared objects for drift. The second calibration may be a fitted model or an object bank.

Usage

```
btl_equate(fit1, fit2, alpha = 0.05, p_adjust = "holm", independent = NULL)
```

Arguments

fit1	A fitted object from <code>btl</code> : the calibration whose scale (origin) the equating targets.
fit2	A second <code>btl</code> fit, or a bank: a data frame with columns <code>object</code> , <code>location</code> , and optionally <code>se</code> ; object names must be unique and locations finite. Bank-based drift inference requires the joint location covariance as a square matrix in <code>attr(fit2, "cov_location")</code> , ordered like the bank rows (or named by object), unless the bank is treated as fixed with zero SEs. A bank whose covariance was estimated from a finite number of independent sampling units may carry their residual degrees of freedom in <code>attr(fit2, "df_location")</code> . For a polytomous fit the bank must carry <code>attr(bank, "m")</code> matching the number of fitted score steps.
alpha	Significance level for the (multiplicity-adjusted) drift tests.
p_adjust	Adjustment for the common-object tests, passed to <code>stats::p.adjust</code> . The default is "holm".
independent	Whether the calibrations have independent judges and comparisons. For two fitted objects the default NULL withholds drift tests until independence is stated explicitly. Bank tables are treated as independent unless FALSE is supplied. Dependent calibrations require a joint or paired bootstrap for inference.

Details

Let d_j be the location difference for common object j and v_j its marginal variance. The origin shift is the precision-weighted mean

$$\hat{s} = \frac{\sum_j d_j / v_j}{\sum_j 1 / v_j}.$$

Each object is tested using its shifted difference $d_j - \hat{s}$. The covariance calculation retains the dependence induced by the sum-zero constraints. Drift tests require independent calibrations and at least three common objects with usable covariance information.

The common-object set should contain a stable majority. If most common objects move in the same direction, the estimated shift follows them and stable objects can appear to drift. In that case, repeat the equating with a substantively justified anchor set.

Value

A list of class "rasch_btl_equate": the comparison table (per common object: object, both locations and standard errors, their difference, the shifted_difference against the estimated origin, the pooled se_diff, t, raw and adjusted p, and the drifting flag); the estimated shift and its shift_se; equated, the second calibration's full object table re-expressed on fit1's scale; the number of common objects n_common; the number usable for inference n_inference; whether inference was available inferential; alpha; p_adjust; and notes.

References

Bramley, T. (2007). Paired comparison methods. In P. Newton, J. Baird, H. Goldstein, H. Patrick, & P. Tymms (Eds.), *Techniques for monitoring the comparability of examination standards* (pp. 246-294). London: Qualifications and Curriculum Authority.

Examples

```

set.seed(1)
beta <- setNames(seq(-2, 2, length.out = 8), paste0("0", 1:8))
sim <- function(objs) {
  pr <- t(utils::combn(objs, 2))
  d <- data.frame(a = rep(pr[, 1], each = 40), b = rep(pr[, 2], each = 40))
  d$win <- ifelse(runif(nrow(d)) < plogis(beta[d$a] - beta[d$b]), d$a, d$b)
  btl(d, "a", "b", "win")
}
eq <- btl_equate(sim(paste0("0", 1:7)), sim(paste0("0", 2:8)),
  independent = TRUE)
eq$table

```

btl_explanatory

*Fit an explanatory comparative judgement model***Description**

Constrains Bradley–Terry–Luce object locations to linear functions of observed object characteristics. The formulation applies to dichotomous and ordered comparative judgements; ordered-response thresholds retain the structure selected in thresholds.

Usage

```

btl_explanatory(
  data,
  predictors,
  formula,
  object_a,
  object_b,
  winner = NULL,
  response = NULL,
  margin = NULL,
  judge = NULL,
  count = NULL,
  order = NULL,
  position = FALSE,
  ties = c("drop", "half", "error"),
  thresholds = c("free", "pc"),
  maxit = 60,
  tol = 1e-08
)

```

Arguments

data	A data frame with one comparison per row.
predictors	Data frame with one row per object, an object column, and the predictors named in formula.

formula	One-sided explanatory formula, including selected interactions if required.
object_a, object_b	Names of the columns holding the two objects compared.
winner	Name of the column holding the winner of each row: its value must equal one of the two objects. "tie" and "draw" mark ties. Ignored when response is supplied.
response	Optional ordered response favouring object_a over object_b: an ordered factor from least to greatest preference for object_a, or integer scores 0 . . m.
margin	Optional ordered margin-of-victory column, combined with winner to construct an orientation-invariant response.
judge	Optional name of a judge column; enables the judge fit table and clusters the sandwich standard errors by judge.
count	Optional name of a column of replication counts (a row standing for several identical comparisons).
order	Optional column giving each judge's comparison sequence; requires judge. See Details. Incompatible with ties = "half".
position	If TRUE, estimate a first-presentation advantage, treating object_a as the first object in each comparison.
ties	How to treat ties in the dichotomous analysis: "drop" (default, removed with a note), "half" (half a win each way, a common pragmatic device – flagged in the notes because the halves are not independent Bernoulli trials), or "error". With polytomous responses, code ties as a middle category instead.
thresholds	"free" (default) estimates every symmetric threshold; "pc" retains only the symmetric spread component.
maxit, tol	Newton-Raphson iteration cap and convergence tolerance.

Details

For objects a and b ,

$$\log\{P(a \succ b)/P(b \succ a)\} = \beta_a - \beta_b, \quad \beta_i = \mathbf{z}_i^T \boldsymbol{\gamma}.$$

The scale origin is fixed at mean object location zero. Numeric predictors are continuous, unordered factors are categorical, and ordered factors use successive contrasts between adjacent levels. Character predictors are converted to unordered factors. Selected interactions may be included in formula. A free calibration is retained for `explanatory_test()`. Standard errors use the same sandwich covariance as `btl()`; when judges are identified, coefficient tests use the judge-clustered covariance and a t reference with judge-cluster degrees of freedom. Holm adjustment covers the coefficient family.

Value

An object of class "rasch_btl_explanatory", inheriting from "rasch_btl".

References

- Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39, 324–345.
- Fischer, G. H. (1973). The linear logistic test model as an instrument in educational research. *Acta Psychologica*, 37, 359–374.

See Also

[btl](#), [explanatory_test](#), [explanatory_diagnostics](#), and [relax_btl_explanatory](#).

Examples

```
set.seed(1)
q <- data.frame(object = LETTERS[1:6],
                 domain = rep(0:1, each = 3))
beta <- setNames(0.8 * q$domain, q$object)
pr <- t(combn(q$object, 2))
d <- data.frame(a = rep(pr[, 1], each = 20),
               b = rep(pr[, 2], each = 20))
p <- plogis(beta[d$a] - beta[d$b])
d$winner <- ifelse(runif(nrow(d)) < p, d$a, d$b)
fit <- btl_explanatory(d, q, ~ domain, "a", "b", winner = "winner")
fit$object_coefficients
explanatory_test(fit)
```

btl_information

Information and targeting of a paired-comparison design

Description

Calculates the Fisher information supplied by the observed comparison design. For location difference $d = \beta_a - \beta_b$, one dichotomous comparison contributes

$$I(d) = P(a \succ b)\{1 - P(a \succ b)\}.$$

For an ordered comparison, the contribution is the variance of the response score. Information is summed over the comparisons involving each object, including replication counts.

Usage

```
btl_information(fit)
```

Arguments

`fit` A paired-comparison fit from [btl](#).

Details

se_naive = 1/sqrt(information) treats each object’s comparisons in isolation. It is a description of the design, not the fitted standard error or a bound on it. The fitted standard error also reflects joint estimation, the identifying constraint, and judge clustering. The fitted model must have converged.

Value

A list of class "rasch_btl_info": objects (per object: location, the fit’s se, n_comparisons, the design information, and se_naive); pairs (per observed pair: n, the mean location gap, and the pair’s information); comparisons (per comparison: the signed gap, weight, and the single-comparison information); the scalar total information; m; the clustered flag; and notes.

References

Pollitt, A. (2012). The method of adaptive comparative judgement. *Assessment in Education*, 19(3), 281-300.

See Also

[plot_btl_targeting](#), [btl_next_pairs](#)

Examples

```
set.seed(1)
beta <- c(A = -1, B = -0.3, C = 0.4, D = 0.9)
pr <- t(combn(names(beta), 2))
d <- data.frame(a = rep(pr[, 1], each = 30), b = rep(pr[, 2], each = 30))
d$win <- ifelse(runif(nrow(d)) < plogis(beta[d$a] - beta[d$b]), d$a, d$b)
btl_information(btl(d, "a", "b", "win"))
```

btl_next_pairs	<i>Recommend the next informative comparisons (adaptive step)</i>
----------------	---

Description

Ranks candidate object pairs by the information expected from one additional comparison at the current estimates (Pollitt 2012). By default, priority is the one-step reduction in total location variance from a rank-one covariance update. This favours close pairs and objects measured with less precision.

Usage

```
btl_next_pairs(fit, n = 10, weight_se = TRUE)
```

Arguments

fit	A paired-comparison fit from btl .
n	Number of pairs to return.
weight_se	If TRUE (the default), rank pairs by their one-step reduction in total location variance. When the fit has no covariance, the fallback priority is expected information multiplied by the sum of the two squared standard errors. If FALSE, rank pairs by expected information alone.

Details

The procedure is a greedy, one-step ranking rather than a jointly optimal design. Applied to a sandwich covariance, the update ranks pairs but does not give an exact variance reduction. Adaptive selection can also inflate a separation reliability calculated from the same comparisons (Bramley 2015). The fitted model must have converged.

Value

A data frame of the top *n* candidate pairs, each oriented to its stronger object: *object_a*, *object_b*, the location gap, *n_existing* (replications already observed for the pair), *expected_information* (of one new comparison), and *priority*. Sorted by *priority* (or by *expected_information* when *weight_se* = FALSE).

References

Pollitt, A. (2012). The method of adaptive comparative judgement. *Assessment in Education*, 19(3), 281-300. Bramley, T. (2015). Investigating the reliability of Adaptive Comparative Judgment. *Cambridge Assessment Research Report*.

See Also

[btl_information](#), [plot_btl_targeting](#)

Examples

```
set.seed(1)
beta <- c(A = -1, B = -0.3, C = 0.4, D = 0.9)
pr <- t(combn(names(beta), 2))
d <- data.frame(a = rep(pr[, 1], each = 30), b = rep(pr[, 2], each = 30))
d$win <- ifelse(runif(nrow(d)) < plogis(beta[d$a] - beta[d$b]), d$a, d$b)
btl_next_pairs(btl(d, "a", "b", "win"), n = 5)
```

btl_transitivity	<i>Transitivity of paired comparisons</i>
------------------	---

Description

Summarises circular triads in the observed paired comparisons. A triad is circular when A is preferred to B, B to C, and C to A. For a complete tournament, the function reports Kendall's coefficient of consistency (Kendall and Babington Smith 1940). Judge-specific summaries are returned when judges are available.

Usage

```
btl_transitivity(fit, min_triples = 5L)
```

Arguments

fit	A paired-comparison fit from btl .
min_triples	A judge is reported only if this many complete triples (all three pairs judged) are available.

Details

A circular-triad rate of one quarter is the benchmark for a random tournament. It is not the expected rate under a fitted BTL model with unequal object locations, so this function is a descriptive consistency measure rather than a calibrated goodness-of-fit test.

Value

A list of class "rasch_btl_transitivity": summary (one row: objects, pairs compared, complete triples, circular triads, the circular rate, the chance rate 0.25, the consistency index $1 - \text{rate}/0.25$, and Kendall's zeta when the design is a complete round-robin with no exactly-tied pair – NA otherwise); objects (each object's circular-triad involvement); judges (per-judge consistency, when judges exist); and notes.

References

Kendall, M. G., & Babington Smith, B. (1940). On the method of paired comparisons. *Biometrika*, 31(3/4), 324-345.

Examples

```
set.seed(1); objs <- LETTERS[1:6]; beta <- setNames(seq(-1.5, 1.5, len = 6), objs)
pr <- t(utils::combn(objs, 2))
d <- data.frame(a = rep(pr[, 1], each = 20), b = rep(pr[, 2], each = 20))
d$win <- ifelse(runif(nrow(d)) < plogis(beta[d$a] - beta[d$b]), d$a, d$b)
btl_transitivity(btl(d, "a", "b", "win"))
```

chisq_detail	<i>Class-interval detail for one item's chi-square test of fit</i>
--------------	--

Description

The per-class-interval breakdown behind an item's item-trait chi-square, as dissected in Andrich and Marais (2019, ch. 13): for every class interval the size, the maximum and mean person location, the standardised residual between observed and expected interval means, its squared chi-square component, the observed and expected means (OM, EV), the sample-size-free effect size $ES = (OM - EV)/\sqrt{\text{mean } V}$, and per response category the observed proportion (OBS.P), the mean model probability (EST.P), and the observed conditional threshold proportion (OBS.T), the proportion scoring k among those scoring $k - 1$ or k .

Usage

```
chisq_detail(fit, item)
```

Arguments

- fit A fitted object from [rasch](#).
- item Item name or index.

Value

A list with item, location, the intervals data frame, the categories data frame, the whole-sample observed mean ave, and the item's total chisq, df, and p. Intervals with fewer than 2 responders are shown but carry no chi-square contribution (used = FALSE), matching the item-trait computation.

Examples

```
set.seed(1)
d <- seq(-1.5, 1.5, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
chisq_detail(rasch(X), "I3")$intervals
```

combine_items	<i>Combine items into subtests and re-analyse</i>
---------------	---

Description

Replaces each nominated item group by a polytomous super-item whose score is the sum of its members, then refits the model. The function is commonly used to examine item groups identified by [residual_correlations](#).

Usage

```
combine_items(fit, groups, model = "PCM")
```

Arguments

fit	A fitted object from rasch .
groups	A list of character vectors, each naming two or more items to combine; a single vector is also accepted.
model	Model for the re-analysis; defaults to "PCM", which is almost always required because subtests change the maximum scores.

Value

A new [rasch](#) fit on the combined structure, with the combinations recorded in its notes. Person and item estimates are recalculated. Fit grouping, external anchors on unchanged items, keyed scoring, PCM constraints and optimisation controls are retained. Anchored items cannot be combined because the resulting superitem has no corresponding external anchor.

See Also

[drop_items](#) to remove an item rather than combine it, and [residual_correlations](#) for the dependence that motivates combining.

Examples

```
set.seed(1); Np <- 500; L <- 8
d <- seq(-2, 2, length.out = L)
X <- matrix(rbinom(Np * L, 1, plogis(outer(rnorm(Np), d, "-"))), Np, L)
colnames(X) <- paste0("I", 1:L)
X[, 5] <- ifelse(runif(Np) < 0.9, X[, 4], X[, 5]) # dependent pair
fit <- rasch(X)
fit2 <- combine_items(fit, list(c("I4", "I5")))
fit2$items$item
```

compare_fits

Compare fitted Rasch models

Description

Builds a comparison table for two or more fits from [rasch](#), [rasch_mfrm](#), [rasch_efrm](#), or (all together) [btl](#). For fits of the same response data (identical item columns, maximum scores, and number of persons) the pairwise conditional log-likelihoods share their conditional information, and twice the difference from the reference fit is reported with the difference in parameter counts; this is descriptive (composite likelihood), and most meaningful for nested structures such as RSM inside PCM.

Usage

```
compare_fits(..., reference = 1)
```

Arguments

...	Two or more fitted objects, preferably named. Supply either all Rasch-family fits or all btl fits. For btl, fits of the same comparison data (same objects, comparisons, and judges) support the likelihood columns – e.g. free versus principal-component thresholds, with and without a position effect or within-judge dependence.
reference	Index or name of the reference fit for the log-likelihood difference; defaults to the first.

Details

The calibrated comparison is carried by the composite-likelihood information criteria `cl_aic` (Varin and Vidoni 2005) and `cl_bic` (Gao and Song 2010): $-2cl + c \cdot \text{tr}(H^{-1}J)$, with $c = 2$ or $\log n$. Because every response enters every pair its item forms, the pairwise log-likelihood over-counts the data; the effective parameter count $\text{tr}(H^{-1}J)$ from the Godambe matrices – the same quantity whose eigenvalues calibrate `lr_test` – absorbs exactly that over-counting, where the nominal parameter count would not. n counts independent units: persons contributing at least one informative pair, or judges for paired-comparison fits (count-weighted comparisons when unclustered). Smaller is better; the criteria are valid across models of the same data whether or not they nest, and are NA (with the reason in the printed note) for MFRM and EFRM fits, which do not carry their Godambe matrices.

Across different data preparations (subtests, splits, facet or frame structures) the likelihoods are not comparable. The table retains descriptive context: total trait chi-square per degree of freedom, calibration and person fit-residual SDs (ideal 1), PSI, and alpha where applicable (OSI for paired comparisons). Alpha is NA when an MFRM or EFRM item is represented by several response cells. These columns do not provide a formal selection test across different response data.

Value

A data frame with one row per fit: label, model, persons, items (judges, objects, comparisons for btl), parameters, log-likelihood, `eff_params`, `cl_aic`, `cl_bic`, comparability with the reference, `two_delta_ll` and `delta_parameters` (same-data fits only), chi-square per df, fit residual SDs, PSI, and alpha (OSI for btl).

Examples

```
set.seed(1)
simP <- function(th, tau) {
  x <- 0:length(tau)
  p <- exp(x * th - c(0, cumsum(tau)))
  p / sum(p)
}
th <- rnorm(400)
X <- sapply(seq(-1, 1, length.out = 6), function(b)
  sapply(th, function(t)
```

```

sample(0:3, 1, prob = simP(t, b + c(-0.8, 0, 0.8))))
colnames(X) <- paste0("R", 1:6)
compare_fits(PCM = rasch(X, model = "PCM"),
             RSM = rasch(X, model = "RSM"))

```

ctt_table

Traditional (classical test theory) statistics

Description

The classical companion table conventionally reported alongside a Rasch analysis (Andrich and Marais 2019, chs. 3-5), on complete cases only: per item the facility (mean score over maximum), the item-total and corrected item-rest correlations, the discrimination index $DI = PRU - PRL$ (mean proportion-of-maximum in the upper third of total scores minus the lower third). Equal total scores remain in the same third, so the group sizes can differ and DI is withheld when three distinct score groups cannot be formed. The table also gives alpha if the item is deleted; the summary gives coefficient alpha, the raw-score mean, SD, and the classical standard error of measurement $s\sqrt{1 - \alpha}$, which unlike the Rasch SE is one value for all persons.

Usage

```
ctt_table(fit, missing = c("complete", "available"))
```

Arguments

fit	A fitted object from rasch whose columns form one administered item set. Expanded EFRM and MFRM response-cell matrices are not accepted; one-cell-per-item reductions are.
missing	"complete" (default) computes the classical table on respondents who answered every item, matching the textbook total-score definitions. "available" retains itemwise and pairwise available cases as an explicitly exploratory summary; respondents answering different item sets need not be comparable.

Value

A list of class "rasch_ctt": the per-item table (item, n, min, max, facility, item_total, item_rest, di, alpha_drop), and the scalars alpha, n (complete cases), mean, sd, and sem.

References

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, 16, 297–334.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 8)
X <- matrix(rbinom(400 * 8, 1, plogis(outer(rnorm(400), d, "-"))), 400, 8)
colnames(X) <- paste0("I", 1:8)
ctt_table(rasch(X))
```

dependence_magnitude *Estimate the magnitude of response dependence between two items*

Description

Quantifies how strongly a dependent item's response follows an independent item's response, in logits, by the resolution method of Andrich and Kreiner (2010; polytomous generalisation Andrich, Humphry and Marais 2012), by resolution of the dependent item. The dependent item is resolved into one item per category of the independent item (each carrying the responses of the persons who gave that category), both original items are removed, and the model refitted. Under dependence of magnitude d , threshold k of the resolved item for category x_i is shifted by $-d$ when $k \leq x_i$ and $+d$ otherwise, so each threshold yields $\hat{d}_k = (\hat{\delta}_{ji(k)}(x_i = k - 1) - \hat{\delta}_{ji(k)}(x_i = k))/2$ and $\hat{d}$ is their mean (eq. 24.7 of Andrich and Marais 2019). The resolved threshold estimates share the calibration of the remaining items and are therefore correlated. The standard error of $\hat{d}$ is calculated from their full sandwich covariance.

Usage

```
dependence_magnitude(fit, dependent, independent)
```

Arguments

fit A fitted object from [rasch](#).

dependent, independent Item names or indices: the item hypothesised to depend, and the item it depends on. Both must share the same maximum score (the formalisation requires it).

Details

Polytomous resolution requires an unconstrained partial credit model so that each resolved threshold can move independently. A rating scale or principal-component threshold constraint is therefore refused. The refit otherwise retains the original fit grouping, keyed scoring, anchors on items that remain, and optimisation controls. Both calibrations must converge before the magnitude and its standard error are reported. MFRM virtual items are resolved through this unconstrained PCM. EFRM virtual frames are mutually exclusive and must first be reduced to an observable frame or linked design block. For an explanatory fit, the remaining items retain their explanatory restrictions and the resolved copies receive free fixed departures.

Value

A list of class "rasch_dependence": the estimate d , its se , z and p for the hypothesis $d = 0$, the per-threshold table thresholds (columns k , δ_{lo} , δ_{hi} , d_k , se_k), and the resolved refit.

References

Andrich, D. and Kreiner, S. (2010). Quantifying response dependence between two dichotomous items using the Rasch model. *Applied Psychological Measurement*, 34, 181-192. Andrich, D., Humphry, S. M. and Marais, I. (2012). Quantifying local, response dependence between two polytomous items using the Rasch model. *Applied Psychological Measurement*, 36, 309-324.

Examples

```
set.seed(1); N <- 700
d0 <- seq(-1.5, 1.5, length.out = 8)
X <- matrix(rbinom(N * 8, 1, plogis(outer(rnorm(N), d0, "-"))), N, 8)
X[, 5] <- ifelse(runif(N) < 0.75, X[, 4], X[, 5]) # I5 follows I4
colnames(X) <- paste0("I", 1:8)
dependence_magnitude(rasch(X), dependent = "I5", independent = "I4")
```

dif_anova

Differential item functioning by residual analysis of variance

Description

Tests uniform and non-uniform DIF by analysing each item's standardised residuals over person factors and trait class intervals (Andrich and Marais 2019, ch. 16). Several person factors are fitted jointly. The function also supports designs containing both between-person and within-person factors.

Usage

```
dif_anova(
  fit,
  factors = NULL,
  n_groups = NULL,
  p_adjust = "holm",
  alpha = 0.05,
  effects = c("main", "factorial"),
  sizes = FALSE,
  id = NULL,
  within = NULL,
  pool_facets = TRUE
)
```

Arguments

<code>fit</code>	A fitted object from rasch , rasch_mfrm , or rasch_efrm .
<code>factors</code>	A vector (one factor), a data frame of person factors, or a character vector naming factor columns nominated in the fit. Defaults to every factor stored in the fit.
<code>n_groups</code>	Number of trait class intervals. The default uses the smallest joint factor cell to retain about 30 expected responses per interval and cell, with between 2 and 10 intervals. The selected value is returned in <code>n_groups</code> .
<code>p_adjust</code>	Multiplicity adjustment over all item-by-term tests; default "holm". Use "BH" only for false-discovery-rate screening rather than familywise control.
<code>alpha</code>	Significance level applied to the adjusted probabilities.
<code>effects</code>	"main" (default) models several factors additively (each factor's main effect and its class-interval interaction, but no factor-by-factor terms); "factorial" also crosses the factors with each other. Immaterial with a single factor.
<code>sizes</code>	If TRUE, refit each flagged item-term and calculate pairwise DIF differences in logits using dif_size .
<code>id</code>	Person identifier for stacked or repeated-measures data. It may be a column name stored in the fit or a vector with one value per row; by default the identifier carried by the fit is used.
<code>within</code>	Names of within-person factors. With repeated identifiers, varying factors are detected automatically when this is omitted. See Details for the mixed-design analysis.
<code>pool_facets</code>	For MFRM fits: pool residuals to the underlying items (the default), so DIF is tested per item rather than per item-by-facet cell; FALSE tests each cell as its own item. EFRM response cells are always pooled by item, so this argument does not alter EFRM fits. Ignored for ordinary fits.

Details

With one factor G and class interval C , the residual model is

$$z = \mu + G + C + G : C + \varepsilon.$$

The factor term tests uniform DIF and its interaction with class interval tests non-uniform DIF. With several factors, `effects = "main"` fits $(f_1 + f_2 + \dots) * ci$; `effects = "factorial"` also includes factor-by-factor interactions. Type II sums of squares are used. The multiplicity adjustment covers all item-by-DIF-term tests, including both uniform and non-uniform DIF; the class-interval main effect is a nuisance term and is not included.

When identifiers repeat, the person is the unit of analysis. Between-person terms use person means and the between-person error stratum. Within-person terms use orthonormal contrasts of person-by-cell means. A Greenhouse–Geisser correction is applied to within-person factors with more than two levels. Persons missing a required cell are excluded from the corresponding within-person test. In incomplete mixed designs, within-cell effects are removed before the between-person analysis. Uniform between-person factor terms use HC3 covariance so unequal group sizes, leverage, and differing precision of person means do not impose a common residual variance. Class-interval

interactions retain the residual-ANOVA reference used to test non-uniform DIF. For between-person design matrix X , residuals e_i , and leverages h_i ,

$$\hat{V}_{HC3} = (X^T X)^{-1} X^T \text{diag} \left\{ \frac{e_i^2}{(1 - h_i)^2} \right\} X (X^T X)^{-1}.$$

A significant higher-order factor term supersedes its component terms in the summary. For EFRM fits, frame-defining factors are excluded because they define the model rather than a separate DIF contrast; testing such a factor means stepping outside the model, which is what [frame_invariance](#) does. MFRM residuals are pooled to underlying items unless `pool_facets = FALSE`. EFRM response cells are always pooled by item; the frame-defining factors remain excluded. Inference is available only from a converged calibration.

Value

A list with:

`summary` One row per item and group term, containing the uniform and non-uniform tests, partial eta-squared, adjusted probabilities, DIF flags, and supersession flag.

`terms` The complete item-wise analysis-of-variance tables.

`sizes` When requested, pairwise logit differences for the significant, non-superseded item-terms.

`posthoc` When `sizes = TRUE`, marginal pairwise differences for main effects and difference-in-differences magnitudes for interactions, calculated by [dif_posthoc](#).

`between_covariance` The covariance reference used for uniform between-person terms.

The remaining components record the factors, class intervals, adjustment, significance level, and design settings.

References

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2), 65–70.

Hagquist, C. and Andrich, D. (2017). Recent advances in analysis of differential item functioning in health research using the Rasch model. *Health and Quality of Life Outcomes*, 15, 181.

MacKinnon, J. G. and White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, 29(3), 305–325.

Maxwell, S. E. and Delaney, H. D. (2004). *Designing Experiments and Analyzing Data: A Model Comparison Perspective* (2nd ed.). Lawrence Erlbaum.

See Also

[dif_size](#), [dif_contrasts](#), and [resolve_dif](#); and [frame_invariance](#) for the frame-defining factor this function excludes.

Examples

```

set.seed(1); n <- 800
d <- seq(-1.5, 1.5, length.out = 6)
g1 <- rep(c("a", "b"), each = n / 2)
g2 <- rep(c("x", "y"), times = n / 2)
sh <- matrix(0, n, 6); sh[g1 == "b", 2] <- 0.8
X <- matrix(rbinom(n * 6, 1, plogis(outer(rnorm(n), d, "-") - sh))), n, 6)
colnames(X) <- paste0("I", 1:6)
fit <- rasch(data.frame(X, g1 = g1, g2 = g2), factors = c("g1", "g2"))
dif_anova(fit)$summary

# Mixed design: group is between persons and occasion is within persons.
N <- 320; theta <- rnorm(N); group <- rep(c("A", "B"), each = N / 2)
make_wave <- function(occasion_shift) {
  shift <- matrix(0, N, 6)
  shift[group == "B", 2] <- 0.9
  shift[, 5] <- occasion_shift
  matrix(rbinom(N * 6, 1,
    plogis(outer(theta, d, "-") - shift))), N, 6)
}
Xm <- rbind(make_wave(0), make_wave(1.0))
colnames(Xm) <- paste0("I", 1:6)
repeated <- data.frame(Xm, group = rep(group, 2),
  occasion = rep(c("T1", "T2"), each = N))
mixed_fit <- rasch(repeated, id = rep(seq_len(N), 2),
  factors = c("group", "occasion"))
mixed_dif <- dif_anova(mixed_fit, within = "occasion")
subset(mixed_dif$summary, uniform_DIF | nonuniform_DIF)

```

dif_contrasts

Planned DIF contrasts

Description

Tests a specified family of one-degree-of-freedom DIF contrasts. By default, contrasts are derived from the factor structure: differences for two-level factors, polynomial trends for ordered factors, and pairwise or level-against-rest comparisons for nominal factors. Leading contrasts are crossed to form two-factor interactions. User-supplied cell weights are also accepted.

Usage

```

dif_contrasts(
  fit,
  factors = NULL,
  items = NULL,
  within = NULL,
  id = NULL,

```

```

contrasts = "auto",
p_adjust = "holm",
alpha = 0.05,
flag_logits = 0.5,
min_n = 20
)

```

Arguments

fit	A fitted object from rasch or rasch_mfrm .
factors	A data frame of person factors, a character vector naming factors nominated in the fit, or a single grouping vector. Defaults to every factor stored in the fit.
items	Item names or indices to test; all items by default.
within	Names of factors that vary within person (for example time). Detected automatically when id is supplied and a factor varies within an id.
id	Person identifier with one entry per row, or the name of a nominated factor holding it. By default the identifier stored by the fitted model is used, so stacked designs retain their pairing.
contrasts	"auto" (derive the family from the factor structure) or a named list of numeric cell-weight vectors, each named by the design-cell labels (factor levels joined by ":"). Weights are rescaled so the positive and negative parts each sum to one.
p_adjust	Familywise adjustment across items and contrasts. The default is "holm".
alpha	Significance level for the adjusted probabilities.
flag_logits	Absolute estimate flagged as practically significant.
min_n	Cells with fewer responders to an item are dropped from that item's resolution, with a note.

Details

Each logit contrast is calculated from resolved item locations. Weights are scaled so their positive and negative parts each sum to one. With repeated persons, inference uses person-level residual contrast scores with the same cell weights as the resolved estimate. Nuisance-factor cells are averaged equally rather than in proportion to their sample sizes. Independent between-person cells are then combined with a Welch–Satterthwaite reference. The resolved logit estimate is retained, but its calibration-based standard error is withheld because it does not include repeated-person dependence.

For independent rows, a contrast with weights $\mathbf{c}$ is

$$\Delta_i = \mathbf{c}^\top \delta_i, \quad \text{SE}(\Delta_i) = \sqrt{\mathbf{c}^\top \mathbf{V}_i \mathbf{c}}.$$

In a repeated-measures design, a within-person contrast is formed from the standardised residuals,

$$s_p = \sum_l c_l z_{pl},$$

and tested over persons. The complete cell-weight vector is retained for main effects and interactions, so the residual test and resolved estimate address the same marginal contrast. The sign of

each residual test is aligned with the resolved logit contrast. Contrasts require a converged calibration. For an MFRM fit, underlying items are pooled over their facet cells by default. EFRM fits are excluded because the required split refit would discard the frame units.

Value

A list of class "rasch_dif_contrasts": table (one row per item and contrast: estimate in logits, SE, statistic, df where a t test was used, raw and adjusted p, 95 per cent interval, significant, practical, within), family (the derived questions with their cell weights), the settings, and any notes.

References

- Maxwell, S. E. and Delaney, H. D. (2004). Designing Experiments and Analyzing Data (2nd ed.). Erlbaum.
- Andrich, D. and Hagquist, C. (2015). Real and artificial differential item functioning in polytomous items. Educational and Psychological Measurement, 75(2), 185–207.
- Hagquist, C. and Andrich, D. (2017). Recent advances in analysis of differential item functioning in health research using the Rasch model. Health and Quality of Life Outcomes, 15, 181.

See Also

[dif_anova](#) and [dif_size](#).

Examples

```
set.seed(1); n <- 600
d <- seq(-2, 2, length.out = 8); g <- rep(c("a", "b"), each = n / 2)
sh <- matrix(0, n, 8); sh[g == "b", 3] <- 0.8
X <- matrix(rbinom(n * 8, 1, plogis(outer(rnorm(n), d, "-") - sh)), n, 8)
colnames(X) <- paste0("I", 1:8)
fit <- rasch(data.frame(X, grp = g), factors = "grp")
dif_contrasts(fit, items = c("I3", "I5"))
```

dif_posthoc

Pairwise follow-up comparisons for a DIF term

Description

Resolves one item's locations over the complete person-factor design and follows up a selected main effect or interaction. Main effects are pairwise marginal differences. Interactions are differences between those differences, providing a logit-scale magnitude for the interaction itself.

Usage

```

dif_posthoc(
  fit,
  item,
  term,
  factors = NULL,
  within = NULL,
  id = NULL,
  p_adjust = "holm",
  alpha = 0.05,
  flag_logits = 0.5,
  min_n = 20
)

```

Arguments

<code>fit</code>	A fitted object from rasch or rasch_mfrm . EFRM fits are excluded because resolved comparisons would discard their frame units.
<code>item</code>	Item name or index.
<code>term</code>	A factor name for a main effect, or a character vector of factor names for an interaction. A single colon-separated string is also accepted when the factor names themselves contain no colon.
<code>factors</code>	The complete person-factor design, specified as for dif_contrasts . Other factors are retained when calculating marginal comparisons.
<code>within</code>	Within-person factor names, specified as for dif_contrasts .
<code>id</code>	Person identifiers, specified as for dif_contrasts .
<code>p_adjust</code>	Familywise adjustment over this post-hoc family; default "holm".
<code>alpha</code>	Significance level for adjusted probabilities.
<code>flag_logits</code>	Absolute logit magnitude flagged as practically important.
<code>min_n</code>	Minimum responders required in a resolved design cell.

Details

For levels a, b of one factor, the comparison is

$$\Delta_{ba} = \bar{\delta}_b - \bar{\delta}_a,$$

where the bars average equally over complete cells of the other nominated factors. For a two-factor interaction, levels a, b and c, d give

$$\Delta_{ba:dc} = (\delta_{bd} - \delta_{ad}) - (\delta_{bc} - \delta_{ac}).$$

Higher-order interactions use the corresponding tensor-product contrast. Standard errors use the full covariance of the resolved locations.

This is the follow-up to a significant DIF term with more than two levels. It reports effects in Rasch logits, adjusts the chosen family of comparisons, and uses person-level scores with the same equal-cell marginal weights in repeated-measures designs.

Value

An object of class "rasch_dif_posthoc", extending the [dif_contrasts](#) result. Its table contains the pairwise marginal differences or interaction contrasts, with logit estimates, standard errors where available, confidence intervals, raw and adjusted probabilities, and statistical and practical flags.

References

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2), 65–70.

See Also

[dif_anova](#), [dif_size](#), and [dif_contrasts](#).

Examples

```
set.seed(1); n <- 800
g <- factor(rep(c("A", "B", "C", "D"), each = n / 4))
sex <- factor(rep(c("female", "male"), length.out = n))
d <- seq(-1.5, 1.5, length.out = 6)
sh <- matrix(0, n, 6); sh[g == "D", 2] <- 0.8
X <- matrix(rbinom(n * 6, 1, plogis(outer(rnorm(n), d, "-") - sh))), n, 6)
colnames(X) <- paste0("I", 1:6)
fit <- rasch(data.frame(X, group = g, sex = sex),
             factors = c("group", "sex"))
dif_posthoc(fit, "I2", term = "group")
```

dif_size

DIF differences between factor levels

Description

Resolves an item by one or more person factors and compares the resulting locations. Several factors in by give pairwise comparisons between their joint cells and can be used to quantify an interaction.

Usage

```
dif_size(
  fit,
  item,
  by,
  p_adjust = "holm",
  alpha = 0.05,
  flag_logits = 0.5,
  min_n = 20
)
```

Arguments

<code>fit</code>	A fitted object from rasch or rasch_mfrm . EFRM fits are excluded because an ordinary split refit would discard their frame units.
<code>item</code>	Item name or index.
<code>by</code>	One or more person-factor names nominated in the fit (several names give interaction cells), or a grouping vector/data frame with one entry per person.
<code>p_adjust</code>	Familywise adjustment over the pairwise comparisons; default "holm".
<code>alpha</code>	Significance level for the adjusted probabilities.
<code>flag_logits</code>	Absolute difference flagged as practically significant.
<code>min_n</code>	Levels with fewer responders to the item are dropped (their resolved locations would be too unstable to compare), with a note.

Details

Let δ_i contain the resolved locations of item i , and let $\mathbf{c}_{ab}$ place 1 on level a , -1 on level b , and zero elsewhere. The reported difference and its standard error are

$$\Delta_{i,ab} = \mathbf{c}_{ab}^T \delta_i,$$

$$\text{SE}(\Delta_{i,ab}) = \sqrt{\mathbf{c}_{ab}^T \mathbf{V}_i \mathbf{c}_{ab}},$$

where $\mathbf{V}_i$ is the full covariance of the resolved locations. Wald probabilities are adjusted over the pairwise family.

With repeated person identifiers, the row-level calibration covariance does not represent within-person sampling dependence. Logit differences and practical flags are retained, but their standard errors and Wald tests are withheld. Use [dif_contrasts](#) for person-level inference in a repeated-measures design.

Value

A list of class "rasch_dif_size". `levels` contains the resolved location, standard error and sample size for each level. `pairs` contains logit differences, Wald statistics, confidence intervals, raw and adjusted probabilities, and practical flags. For dichotomous items it also contains `ets`; for polytomous items it contains the descriptive `signed_area`. Sampling-uncertainty fields are NA when person identifiers repeat.

Magnitude conventions

For dichotomous items, `ets` applies the ETS A, B and C rules to the itemwise comparison. On the logit scale the magnitude cut-points are $1/2.35 = 0.426$ and $1.5/2.35 = 0.638$. Category A also includes an itemwise test that is not significant. Category C requires a magnitude of at least 0.638 and rejection of the interval null $|\Delta| \leq 0.426$; B is the remainder. The category uses the raw itemwise probability, while `significant` uses `p_adjust` over the requested pairwise family.

For a partial credit item with m_i thresholds, the signed area between the two expected-score curves has the closed form

$$SA_{ab} = \int \{E_b(X | \theta) - E_a(X | \theta)\} d\theta = \sum_{k=1}^{m_i} (\delta_{iak} - \delta_{ibk}) = m_i(\beta_{ia} - \beta_{ib}).$$

This is returned as `signed_area`; a positive value means that level `a` has the harder resolved item. It is descriptive and is not given an A/B/C category: score-metric classifications for polytomous DIF are not interchangeable with a PCM logit difference. For pooled MFRM items, the areas use the same precision weight for a facet cell in every group. A comparison is withheld when the groups do not support the same observed response categories.

References

- Andrich, D. and Marais, I. (2019). A Course in Rasch Measurement Theory: Measuring in the Educational, Social and Health Sciences. Springer.
- Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2), 65–70.
- Zieky, M. (1993). Practical questions in the use of DIF statistics in item development. In P. W. Holland and H. Wainer (eds), *Differential Item Functioning* (pp. 337–364). Erlbaum.
- Linacre, J. M. and Wright, B. D. (1989). Mantel-Haenszel DIF and PROX are equivalent! *Rasch Measurement Transactions*, 3(2), 51–53.
- Cohen, A. S., Kim, S.-H. and Baker, F. B. (1993). Detection of differential item functioning in the graded response model. *Applied Psychological Measurement*, 17(4), 335–350.
- Raju, N. S. (1988). The area between two item characteristic curves. *Psychometrika*, 53(4), 495–502.

See Also

[dif_anova](#) and [dif_contrasts](#).

Examples

```
set.seed(1); n <- 600
d <- seq(-2, 2, length.out = 8); g <- rep(c("a", "b"), each = n / 2)
sh <- matrix(0, n, 8); sh[g == "b", 3] <- 0.8
X <- matrix(rbinom(n * 8, 1, plogis(outer(rnorm(n), d, "-") - sh))), n, 8)
colnames(X) <- paste0("I", 1:8)
fit <- rasch(data.frame(X, grp = g), factors = "grp")
dif_size(fit, "I3", by = "grp")
```

dimensionality_magnitude

Magnitude of multidimensionality from a subtest analysis

Description

Estimates how strongly two or more hypothesised subscales measure distinct traits, by Andrich's (2016) comparison of two reliability calculations: one treating all items as independent (which inflates reliability under multidimensionality) and one on the subtest analysis in which each subscale is combined into a single polytomous super-item (which absorbs the unique subscale variance).

Under the bifactor formalisation $\beta_{ns} = \beta_n + c \beta'_{ns}$ (Marais and Andrich 2008), with S subscales of K items,

$$c^2 = S(r_1/r_2 - 1) \frac{SK - 1}{S(K - 1)},$$

the latent correlation between subscales is $\rho = 1/(1 + c^2)$, and $A = S/(S + c^2)$ is the proportion of common (non-unique, non-error) variance. Both the person separation index and coefficient alpha versions are reported (Andrich and Marais 2019, ch. 24). Both the original and subtest calibrations must converge.

Usage

```
dimensionality_magnitude(fit, subtests)
```

Arguments

fit	A fitted object from rasch .
subtests	A list of character vectors assigning <i>every</i> item of the fit to one subscale (at least two subscales of two or more items). The published magnitude formula requires equal subscale sizes.

Value

A list of class "rasch_dim_magnitude": the comparison table (rows PSI and alpha; columns run1, subtest, c2, c, rho, A), the subtest refit, and the design constants S and K.

References

Andrich, D. (2016). Components of variance of scales with a bifactor subscale structure from two calculations of alpha. *Educational Measurement: Issues and Practice*, 35(4), 25-30.

Examples

```
set.seed(1); N <- 500
common <- rnorm(N); u1 <- rnorm(N); u2 <- rnorm(N)
d <- rep(seq(-1, 1, length.out = 5), 2)
X <- sapply(1:10, function(i) rbinom(N, 1,
  plogis(common + 0.8 * (if (i <= 5) u1 else u2) - d[i])))
colnames(X) <- paste0("I", 1:10)
fit <- rasch(X)
dimensionality_magnitude(fit,
  list(paste0("I", 1:5), paste0("I", 6:10)))$table
```

dimensionality_test *Residual-component test of unidimensionality*

Description

Estimates each person separately on two item subsets and compares the two estimates with a per-person t-test (Smith 2002). By default the subsets are defined by the sign of a residual-component loading (the first by default; any leading component may be chosen); they can also be nominated manually (for example, by content). Under unidimensionality and local independence the two subset estimates are independent given the person location, so $t = (\text{theta}_A - \text{theta}_B) / \sqrt{\text{se}_A^2 + \text{se}_B^2}$ is approximately standard normal and about alpha of the tests should reach significance. Persons with an extreme score on either subset are excluded (their weighted-likelihood estimates are most biased there). The proportion of significant tests is reported with an exact (Clopper-Pearson) binomial confidence interval; a lower bound above alpha signals multidimensionality. The test requires a converged calibration.

Usage

```
dimensionality_test(
  fit,
  alpha = 0.05,
  items_positive = NULL,
  items_negative = NULL,
  component = 1,
  min_score_points = 15L
)
```

Arguments

fit	A fitted object from rasch .
alpha	Nominal significance level for the per-person t-tests.
items_positive, items_negative	Optional character vectors naming the two item subsets; both must be given (disjoint, at least two items each), otherwise the sign of a residual component defines the split.
component	Which residual principal component's loading sign defines the default split (ignored when subsets are named). Default the first component.
min_score_points	Score-point threshold below which the verdict carries a caution. Andrich and Marais (2019) recommend subtests of roughly 15 score points for stable subtest estimates; shorter subsets (the norm for ordinary dichotomous tests) still receive a verdict, with a caution field noting the reduced stability. A quiet verdict under caution is inconclusive, not clean: with a four-item subtest the test lacks power where nonparametric alternatives still flag. The procedure is deliberately conservative – in cross-package comparison it held an exact null (no false flags)

while flagging a balanced planted second dimension in about two-thirds of replicates where the DETECT index flagged all; quasi-exact matrix-sampling tests showed elevated null rates on the same data.

Value

A list with the proportion of significant tests, its exact confidence interval, the sample sizes (n used, n_excluded_extreme), the item split and its source, a multidimensional verdict, a caution note when the subtests fall short of min_score_points, and paired_t, the paired t-test of the two subset means (the group-level comparison, which requires pairing because both estimates come from the same persons). When the comparison itself is unavailable (undefined split, degenerate subsets, too few persons) the list carries a note explaining why and multidimensional = NA.

References

Smith, E. V. Jr. (2002). Detecting and evaluating the impact of multidimensionality using item fit statistics and principal component analysis of residuals. *Journal of Applied Measurement*, 3(2), 205–231.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 8)
X <- matrix(rbinom(500 * 8, 1, plogis(outer(rnorm(500), d, "-"))), 500, 8)
colnames(X) <- paste0("I", 1:8)
dimensionality_test(rasch(X))$multidimensional
```

distractor_analysis *Distractor analysis for multiple-choice items*

Description

For every keyed item and response option: the count and proportion choosing it, the mean location of those persons, and the point-biserial correlation between choosing the option and the person measure. Locations and correlations use the rest measure (the person estimate from the other items), so the analysed item cannot credit its own takers. The keyed option should attract the ablest persons and carry the only positive point-biserial; a distractor whose takers are abler than the keyed option's (with at least min_n takers) is flagged as a possible miskey.

Usage

```
distractor_analysis(fit, items = NULL, min_n = 10)
```

Arguments

fit	A fitted object from rasch run with a key.
items	Optional subset of item names; defaults to every keyed item.
min_n	Minimum takers for an option to be eligible for the miskey flag.

Value

A data frame with one row per item-option: item, option, its assigned score, keyed (full credit), n, prop, mean_location, point_biserial, and flag.

Examples

```
set.seed(1); Np <- 400
th <- rnorm(Np)
raw <- sapply(seq(-1, 1, length.out = 6), function(d) {
  ok <- rbinom(Np, 1, plogis(th - d))
  ifelse(ok == 1, "A", sample(c("B", "C", "D"), Np, replace = TRUE))
})
colnames(raw) <- paste0("M", 1:6)
fit <- rasch(raw, key = setNames(rep("A", 6), colnames(raw)))
head(distractor_analysis(fit))
```

distractor_rescore	<i>Propose polytomous option scores from the distractor evidence</i>
--------------------	--

Description

Multiple-choice items can be rescored polytomously so that a distractor carrying information about the trait receives partial credit (Andrich and Styles 2011). This function proposes such a scoring from the rest-measure distractor analysis: within each keyed item, a distractor qualifies for credit when it attracts at least `min_n` takers, its takers' mean rest location exceeds that of the uncredited distractors by more than `z` standard errors of the difference, and it remains below the keyed option. Qualifying distractors are ranked by mean location and scored 1, 2, ... below the keyed option's top score. The result is a proposal for substantive review, not an automatic decision: inspect [plot_distractors](#) and the item content, edit as needed, then refit with `rasch(raw_data, key = proposal$option_scores)`.

Usage

```
distractor_rescore(fit, items = NULL, min_n = 20, z = 1.96)
```

Arguments

<code>fit</code>	A fitted object from rasch run with a key.
<code>items</code>	Optional subset of keyed item names.
<code>min_n</code>	Minimum takers for a distractor to be considered.
<code>z</code>	Required separation, in standard errors, between a credited distractor and the uncredited ones.

Value

A list of class "rasch_rescore": `option_scores`, a data frame (item, option, score) ready for `rasch(key = )` and covering every observed option of the examined items, and evidence, the distractor analysis with the proposed scores and the separation `z` per option.

References

Andrich, D. and Styles, I. (2011). Distractors with information in multiple choice items: A rationale based on the Rasch model. *Journal of Applied Measurement*, 12, 67-95.

Examples

```
set.seed(1); Np <- 600
th <- rnorm(Np)
raw <- sapply(seq(-0.5, 0.5, length.out = 4), function(d) {
  x <- vapply(th, function(b) sample(0:2, 1,
    prob = item_moments(b, c(d - 0.7, d + 0.7))$P), 0L)
  c("D", "B", "A")[x + 1] # B is an informative distractor
})
colnames(raw) <- paste0("M", 1:4)
fit <- rasch(raw, key = setNames(rep("A", 4), colnames(raw)))
pr <- distractor_rescore(fit)
pr$option_scores
```

drop_items

Drop items and refit

Description

Removes named items and refits the analysis with the same model specification.

Usage

```
drop_items(fit, items, boot_reps = NULL)
```

Arguments

fit	A fitted object from rasch or rasch_efrm . Many-facet fits are refused: remove the item's rows from the long-format data and refit rasch_mfrm instead.
items	Item names to remove.
boot_reps	Bootstrap replicates for an EFRM refit. The default retains the fitted specification; a number overrides it.

Details

The refit retains person identifiers and factors, class-interval settings, optimisation controls, anchors, multiple-choice scoring and PCM component constraints. An EFRM refit also retains the item-set and crossed-frame design, linking controls and uncertainty method. The operation is refused if it would remove every anchor, empty an item set or leave the model unidentified.

Item removal changes both the item calibration and the person estimates. For an EFRM it can also change the estimated frame units. Compare the original and refitted results as a sensitivity analysis.

Value

A refitted object of the same class as `fit`, carrying a note recording which items were dropped.

See Also

[frame_invariance](#) and [resolve_frames](#) for frame models; [split_items](#) and [resolve_dif](#) for DIF; and [combine_items](#) for response dependence.

Examples

```
d <- simulate_rasch(300, 8, seed = 1)
fit <- rasch(d, id = "id")
fit2 <- drop_items(fit, "I03")
nrow(fit2$items)
```

equate_tests	<i>Equate two test calibrations through their common items</i>
--------------	--

Description

Places two calibrations on a common origin using their shared items, then tests the shared items for drift. The reference may be a fitted model or an item bank.

Usage

```
equate_tests(fit, reference, shift = c("mean", "none"), independent = NULL)
```

Arguments

fit	A fitted object from rasch .
reference	A second rasch fit, or a data frame with columns <code>item</code> , <code>location</code> , and optionally <code>se</code> . Item names must be unique and locations finite. For bank-based drift inference, attach the bank's joint item-location covariance as a square matrix in <code>attr(reference, "cov_location")</code> , ordered like the bank rows (or named by item); marginal SEs alone do not carry the centring covariance. A bank treated as fixed may instead have zero SEs. A polytomous bank must also include <code>max</code> , the maximum item score.
shift	"mean" (default) allows a scale shift between the two analyses; "none" compares raw locations, appropriate when both analyses are already on a shared (anchored) scale.
independent	Whether the two calibrations use independent sampling units. For two fitted objects this must be stated explicitly: the default <code>NULL</code> withholds drift tests because cross-fit covariance is otherwise unknown. A bank table is treated as independent unless <code>FALSE</code> is supplied. When <code>FALSE</code> , descriptive equating is returned but inferential drift columns are withheld.

Details

Let d_j be the location difference for common item j and v_j its marginal variance. With `shift = "mean"` the scale shift is the precision-weighted mean

$$\hat{s} = \frac{\sum_j d_j / v_j}{\sum_j 1 / v_j},$$

and each item is tested using $d_j - \hat{s}$ with a variance that accounts for the estimated shift through the items' joint covariance. Drift inference requires independent calibrations and at least three common items with usable joint covariance information. Otherwise the function returns a descriptive link. Fitted calibrations must have converged.

Value

A list with the comparison table (locations, standard errors, difference, t, raw and Holm-adjusted p, drift flag), the estimated shift, the location correlation, the root mean square difference after shifting (rmsd), the number of common items `n_common`, the number with usable standard errors `n`, and whether drift inference was available (`inferential`). The `notes` component records exclusions and the reason inference was withheld, where applicable.

Examples

```
set.seed(1); d <- seq(-1.5, 1.5, length.out = 8)
mk <- function() {
  X <- matrix(rbinom(400 * 8, 1, plogis(outer(rnorm(400), d, "-"))), 400, 8)
  colnames(X) <- paste0("I", 1:8); rasch(X)
}
eq <- equate_tests(mk(), mk(), independent = TRUE)
eq$table
```

explanatory_diagnostics

Diagnose fixed departures from an explanatory model

Description

Fits each available item-location, polytomous threshold-structure or comparative-judgement object departure separately from the active model. Probabilities use Kent calibration and Holm adjustment over the complete candidate family.

Usage

```
explanatory_diagnostics(fit, p_adjust = "holm")
```

Arguments

<code>fit</code>	A fitted explanatory Rasch or comparative judgement model.
<code>p_adjust</code>	Multiplicity adjustment over the candidate departures.

Value

A data frame ordered by adjusted probability.

explanatory_test	<i>Compare an explanatory model with its free calibration</i>
------------------	---

Description

Tests explanatory item, threshold or object restrictions against the corresponding free calibration of the same responses. The inferential result uses the first-order Kent calibration for the fitted likelihood and sandwich covariance. The calibration coefficient of determination is

$$R_{cal}^2 = 1 - \frac{\sum_j (\hat{\eta}_j^{free} - \hat{\eta}_j^{expl} - \bar{d})^2}{\sum_j (\hat{\eta}_j^{free} - \bar{\eta}^{free})^2},$$

where $\bar{d}$ removes the arbitrary scale origin. It describes the proportion of variation in the well-determined free threshold calibration (Rasch models) or free object calibration (comparative judgement) reproduced by the explanatory model. It is at most one and may be negative. It is not adjusted for the number of predictors, so with few calibrated parameters it reads above zero even for an uninformative design. $r_squared_adj$ divides the unexplained proportion by its share of the degrees of freedom, $1 - (1 - R_{cal}^2)(n - 1)/df$, where n counts the calibrated parameters compared and df is n minus the rank of the retained explanatory design with its origin, so exclusions that remove a level's only support reduce it. The correction is exact for independent homoskedastic estimates fitted by least squares, which these calibrations are not, so read it as a descriptive optimism adjustment. Read either beside the test rather than in place of it.

Usage

```
explanatory_test(fit)
```

Arguments

`fit` A fitted explanatory Rasch or comparative judgement model.

Value

A one-row data frame containing the raw and Kent-calibrated statistics, degrees of freedom and parameter counts. The primary `p` and the retained `p_kent` are the Kent-calibrated probability. `p_naive` is the unscaled composite-likelihood probability and is provided for methodological inspection, not inference. `r_squared` is the calibration coefficient of determination, `r_squared_adj` its degrees-of-freedom-adjusted counterpart, and `r2_basis` names the calibrated parameters used.

fit_summary_table	<i>Test-of-fit summary as a table</i>
-------------------	---------------------------------------

Description

Returns the model, estimation method, trait chi-square, calibration and person fit-residual moments, fit-location correlations, chi-square flag count, and disordered-threshold count as a two-column table. MFRM and EFRM summaries label their fitted item-by-facet or item-by-frame columns as response cells.

Usage

```
fit_summary_table(fit)
```

Arguments

fit	A fitted object from rasch , or a paired-comparison fit from btl (which reports its own headline set: convergence, pairwise chi-square, object separation, thresholds structure, and dependence effects when estimated).
-----	--

Value

A data frame with columns statistic and value.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(300 * 6, 1, plogis(outer(rnorm(300), d, "-"))), 300, 6)
colnames(X) <- paste0("I", 1:6)
fit_summary_table(rasch(X))
```

frame_invariance	<i>Test item invariance across frames</i>
------------------	---

Description

Calibrates each frame separately and compares the locations and discriminations of items administered in more than one frame.

Usage

```

frame_invariance(
  fit,
  alpha = 0.05,
  adjust = c("holm", "none"),
  se_method = c("conditional", "bootstrap"),
  boot_reps = 200,
  seed = NULL
)

```

Arguments

fit	A fitted object from rasch_efrm .
alpha	Significance level used for flags.
adjust	Either "holm" or "none". Both raw and adjusted probabilities are returned.
se_method	"conditional" treats the estimated frame units as fixed; "bootstrap" refits the complete analysis to person resamples within frame.
boot_reps	Number of bootstrap replicates. At least 30 are required.
seed	Optional bootstrap seed.

Details

Let $\hat{\delta}_{if}$ be the location of item i from a separate calibration of frame f , and let $\hat{\rho}_f$ be that frame's unit from the fitted EFRM. The common-scale location is $\hat{\delta}_{if}^* = \hat{\delta}_{if}/\hat{\rho}_f$. Because each separate calibration has its own origin, pairwise differences are centred over the common thresholds before testing.

The conditional method treats the fitted frame units as fixed. Let $w_i = m_i/\sum_j m_j$, where m_i is the number of thresholds for item i . If $C = I - \mathbf{1}\mathbf{w}^T$ centres the common items on the threshold-weighted origin, the covariance of the location differences is

$$C\{V_1/\hat{\rho}_1^2 + V_2/\hat{\rho}_2^2\}C^T.$$

This is fast and conditions on the estimated units. The discrimination table gives the difference between the two standardised infit statistics, divided by $\sqrt{2}$, together with fitted slopes and their ratio. These quantities are descriptive under the conditional method; it does not report discrimination probabilities.

With `se_method = "bootstrap"`, persons are resampled within frame and the EFRM and separate frame calibrations are refitted. Location tests then use the empirical covariance of the centred differences. The discrimination test uses the bootstrap standard error of the log slope ratio. This includes uncertainty in the fitted frame units but is more computationally demanding.

Raw and Holm-adjusted probabilities are reported. With conditional uncertainty, Holm adjustment covers the location comparisons. With bootstrap uncertainty, it covers the combined family of location and discrimination comparisons. The summary gives the root mean squared location difference and root mean squared standard error for each set and frame pair. Items from different sets cannot be compared because the sets partition the items. Location differences are relative to the mean difference of the common items. Concentrated DIF can therefore produce non-zero centred contrasts

for items that were not themselves shifted. The table identifies the pattern of relative departures; item content or external anchors are needed to determine which items provide the defensible reference. A compared set-by-frame cell must contain at least 50 persons with two or more responses. Items with weakly determined standard errors in either separate calibration are listed in excluded rather than tested.

A flagged item may be resolved with `resolve_frames` when it remains useful within frames, or removed with `drop_items` when it fits poorly more generally. Either change requires a refit. The invariance tests require a converged frame calibration.

Value

An object of class "rasch_frame_invariance". The locations and discrimination tables contain the pairwise item comparisons; summary contains set-level RMSD and RMSE summaries. Under the conditional method, discrimination `p`, `p_adj`, and `flagged` are NA. `excluded` lists items whose observed category structures differed between calibrations or whose separate-frame estimate was weakly determined. The remaining components record the multiplicity and uncertainty settings.

References

Humphry, S. M. (2005). *Maintaining a Common Arbitrary Unit in Social Measurement*. PhD thesis, Murdoch University.

See Also

`resolve_frames` to give a flagged item a location per frame, `drop_items` to remove it altogether, and `rasch_efrm` for the model whose assumption is tested.

Examples

```
d <- simulate_efrm(n_per_group = 300, items_per_set = 8, n_sets = 1,
                  n_groups = 2, group_unit_ratio = 1.4, seed = 2)
tr <- attr(d, "truth")
fit <- rasch_efrm(d, item_sets = tr$item_sets, groups = "group",
                 id = "id", boot_reps = 0)
frame_invariance(fit)
```

guttman_table

Guttman-ordered response matrix and reproducibility

Description

Orders persons by descending location and dichotomous items by ascending location (the Guttman scalogram arrangement), computing the reproducibility coefficient against the deterministic pattern implied by each person's total score. PCM category steps can interleave across items, so polytomous fits are rejected rather than forced into an invalid whole-item deterministic order.

Usage

```
guttman_table(fit)
```

Arguments

fit A fitted object from [rasch](#) whose columns form one administered item set. Expanded EFRM and MFRM response-cell matrices are not accepted; one-cell-per-item reductions are.

Value

A list with the ordered score matrix `matrix` (persons by items, row and column names carrying the ID and item labels), the person and item orderings, and the coefficient of reproducibility CR.

References

Guttman, L. (1944). A basis for scaling qualitative data. *American Sociological Review*, 9(2), 139–150.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(200 * 6, 1, plogis(outer(rnorm(200), d, "-"))), 200, 6)
colnames(X) <- paste0("I", 1:6)
guttman_table(rasch(X))$CR
```

item_moments

Category-score moments for a polytomous item

Description

Category-score moments for a polytomous item

Usage

```
item_moments(theta, tau_i, disc = 1)
```

Arguments

theta Person location, in logits.

tau_i Numeric vector of the item's threshold parameters.

disc Discrimination (frame unit) multiplier on the exponent; 1 for the ordinary Rasch model.

Value

A list with category probabilities P , expected score E , variance V , and third and fourth central moments μ_3 , μ_4 .

Examples

```
item_moments(0.5, c(-1, 0, 1))
```

judge_pair_surprise	<i>Unexpected judgements of one judge, pair by pair</i>
---------------------	---

Description

The comparison-level companion of [judge_surprise](#). Each pair the judge met is oriented to its stronger object (higher consensus location) and given a standardised residual: $z < 0$ means the stronger object won less than its lead predicts – the judge backed the underdog. A matchup is an unexpected judgement when z falls at or below $-\text{flag_z}$ and the pair was seen at least min_n times, i.e. the judge favoured the weaker object further than sampling noise explains. The fitted model must have converged.

Usage

```
judge_pair_surprise(fit, judge, min_n = 1L, flag_z = 1.96)
```

Arguments

fit	A paired-comparison fit from btl with judges.
judge	The judge to profile.
min_n	Pairs met fewer times are shown but never flagged.
flag_z	Absolute residual at or beyond which an upset is flagged.

Value

A list of class "rasch_btl_judge_pairs": pairs (per matchup: the stronger and weaker object and their locations, the location gap, times met n , residual z , the `net_winner`, and the surprise flag); `all_locations`; the judge and settings.

Examples

```
set.seed(1); objs <- LETTERS[1:6]; beta <- setNames(seq(-1.5, 1.5, len = 6), objs)
pr <- t(utils::combn(objs, 2))
d <- data.frame(a = rep(pr[, 1], each = 12), b = rep(pr[, 2], each = 12))
d$judge <- sample(paste0("J", 1:5), nrow(d), TRUE)
d$win <- ifelse(runif(nrow(d)) < plogis(beta[d$a] - beta[d$b]), d$a, d$b)
judge_pair_surprise(btl(d, "a", "b", "win", judge = "judge"), "J1")
```

judge_surprise	<i>Unexpected judgements of one judge</i>
----------------	---

Description

The paired-comparison counterpart of the kidmap. A judge has no ability to condition on, so the reference is the consensus object scale (the pooled locations). For the nominated judge, each object it met is given a standardised residual oriented to the object – how much more ($z > 0$, over-rated) or less ($z < 0$, under-rated) that judge favoured it than its consensus location predicts. A surprise is an object the judge treated against its standing: a strong object under-rated, or a weak object over-rated (residual opposite in sign to the location), beyond `flag_z` and seen at least `min_n` times. The fitted model must have converged.

Usage

```
judge_surprise(fit, judge, min_n = 2L, flag_z = 1.96)
```

Arguments

<code>fit</code>	A paired-comparison fit from btl with judges.
<code>judge</code>	The judge to profile (a value of the fit's judge column).
<code>min_n</code>	Objects met fewer times are shown but never flagged.
<code>flag_z</code>	Absolute residual at or beyond which a contrary judgement is flagged unexpected.

Value

A list of class "rasch_btl_judge": objects (per object met: location, times met `n`, residual `z`, surprise flag and its type); `all_locations` (every object, for orientation); the judge and settings.

Examples

```
set.seed(1); objs <- LETTERS[1:6]; beta <- setNames(seq(-1.5, 1.5, len = 6), objs)
pr <- t(utils::combn(objs, 2))
d <- data.frame(a = rep(pr[, 1], each = 12), b = rep(pr[, 2], each = 12))
d$judge <- sample(paste0("J", 1:5), nrow(d), TRUE)
d$win <- ifelse(runif(nrow(d)) < plogis(beta[d$a] - beta[d$b]), d$a, d$b)
judge_surprise(btl(d, "a", "b", "win", judge = "judge"), "J1")
```

lr_test

*Compare the partial credit and rating scale models***Description**

Compares a fitted partial credit model with the rating scale reparameterisation of the same data. Both raw and composite-likelihood adjusted statistics are returned.

Usage

```
lr_test(fit, maxit = 60, tol = 1e-08)
```

Arguments

fit	An unrestricted, unanchored "PCM" fit from rasch with equal maximum scores across items (the rating parameterisation requires them).
maxit, tol	Passed to the rating-scale refit.

Details

The pairwise conditional likelihood is a composite likelihood: each response contributes to every item pair in which it appears. Consequently, the raw statistic $W = 2(cl_{PCM} - cl_{RSM})$ does not have an ordinary chi-square reference distribution. Its limiting distribution is $\sum_j \lambda_j \chi_1^2$ (Kent 1982; Varin, Reid and Firth 2011), where the λ_j are obtained from the sensitivity matrix H , variability matrix J , and the constraints defining the RSM. The mean-matched statistic is

$$W_{adj} = rW / \sum_j \lambda_j,$$

with r degrees of freedom.

Use `p_adj` for inference. The unadjusted `p` is retained for descriptive comparison with conventional displays. The adjustment is a first-order approximation and can be mildly anti-conservative in small samples with long polytomous tests. Interpret values near the nominal level cautiously in such designs.

Value

A list of class "rasch_lr": raw `chisq`, `df`, `p` (the conventional display); adjusted `chisq_adj`, `p_adj`, and the eigenvalues `lambda`; the two log-likelihoods; and the rating-scale refit (`fit_rsm`).

References

Kent, J. T. (1982). Robust properties of likelihood ratio tests. *Biometrika*, 69, 19-27. Varin, C., Reid, N. and Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*, 21, 5-42.

Examples

```
set.seed(1)
tau <- c(-0.7, 0.7)
X <- sapply(seq(-1, 1, length.out = 6), function(d) vapply(rnorm(300),
  function(b) sample(0:2, 1, prob = item_moments(b, tau + d)$P), 0L))
colnames(X) <- paste0("Q", 1:6)
lr_test(rasch(X, model = "PCM"))
```

pcml	<i>Estimate Rasch thresholds by pairwise conditional maximum likelihood</i>
------	---

Description

Estimates PCM or RSM thresholds by Newton–Raphson maximisation of the pairwise conditional likelihood (Zwinderman 1995).

Usage

```
pcml(X, model = c("PCM", "RSM"), anchors = NULL, maxit = 60, tol = 1e-08)
```

Arguments

X	Persons-by-items integer score matrix (categories from 0). Missing values are handled by pairwise deletion, so linked booklet designs and random missingness estimate without imputation; the item-pair graph must be connected (some person answering items in both of any two blocks), otherwise relative locations between blocks are unidentified and the fit stops with an error naming the blocks – unless anchors fix an item in every block, the disjoint-form equating case.
model	"PCM" or "RSM".
anchors	Optional anchor table for equating: a data frame with columns item (name or column index), k, and tau (the fixed value). A numeric k fixes that single threshold (individual anchoring); k = NA fixes the item's mean location at tau while its thresholds remain free (average anchoring). The remaining parameters are estimated on the anchored scale and no recentring is applied. PCM only.
maxit, tol	Newton-Raphson iteration cap and convergence tolerance.

Details

For the PCM, the adjacent-category log odds are

$$\log\{P(X_{ni} = k)/P(X_{ni} = k - 1)\} = \theta_n - \delta_{ik}.$$

Conditioning on the score for an item pair removes θ_n . The PCM estimates each δ_{ik} ; the RSM imposes $\delta_{ik} = \beta_i + \tau_k$ through a design matrix.

Value

A list containing the threshold table `thr`, covariance matrix `cov_tau`, pairwise conditional log-likelihood, iteration count, convergence flag, notes, and maximum scores `m`. In `thr`, weak marks all thresholds of an item with fewer than eight responses in any category, or a threshold adjacent to a category with fewer than three responses. Standard errors for weak thresholds are reported as NA.

References

Zwinderman, A. H. (1995). Pairwise parameter estimation in Rasch models. *Applied Psychological Measurement*, 19(4), 369–375.

Examples

```
set.seed(1)
d <- seq(-1.5, 1.5, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
pcml(X)$thr
# anchor two items at fixed values (equating)
anchors <- data.frame(item = c("I1", "I6"), k = 1,
                      tau = c(-1.5, 1.5))
pcml(X, anchors = anchors)$thr
```

pcml_pc	<i>Estimate Rasch thresholds using a principal-component parameterisation</i>
---------	---

Description

Re-expresses each item’s thresholds as orthogonal polynomial components: location, spread, skewness, and kurtosis (Andrich 1978, 1985; Pedler 1987). Estimation uses the same pairwise conditional likelihood as `pcml`. With at most three thresholds per item the full parameterisation is exact. Items with four or more thresholds are fitted by a reduced-rank polynomial trend, which can stabilise sparse categories at the cost of restricting the threshold pattern.

Usage

```
pcml_pc(X, n_components = 4, maxit = 60, tol = 1e-08)
```

Arguments

X	Persons-by-items integer score matrix (categories from 0). Missing values are handled by pairwise deletion.
n_components	Maximum number of components per item: 1 (location only) up to 4 (location, spread, skewness, kurtosis; the highest derived by Pedler 1987). Capped per item at its own number of thresholds, and further wherever a component would be collinear with lower-order ones for that item’s threshold count (kurtosis is unidentified, and dropped, at exactly 4 thresholds).
maxit, tol	Newton-Raphson iteration cap and convergence tolerance.

Value

A list with the threshold table `thr` (columns `id`, `item`, `k`, `tau`, `se`), the component table components (one row per item, with location, spread, skewness, kurtosis and their standard errors, NA where an item's rank does not support that component), the threshold covariance matrix `cov_tau`, the pairwise conditional log-likelihood, the iteration count, a convergence flag, and the max-score vector `m`.

References

- Andrich, D. and Luo, G. (2003). Conditional pairwise estimation in the Rasch model for ordered response categories using principal components. *Journal of Applied Measurement*, 4(3), 205–221.
- Zwinderman, A. H. (1995). Pairwise parameter estimation in Rasch models. *Applied Psychological Measurement*, 19(4), 369–375.
- Andrich, D. (1978). A rating formulation for ordered response categories. *Psychometrika*, 43(4), 561–573.
- Andrich, D. (1985). An elaboration of Guttman scaling with Rasch models for measurement. In N. B. Tuma (Ed.), *Sociological Methodology 1985* (pp. 33–80). Jossey-Bass.
- Pedler, P. J. (1987). Accounting for psychometric dependence with a class of latent trait models. PhD thesis, University of Western Australia.

Examples

```
set.seed(1)
d <- seq(-1.5, 1.5, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
pcml_pc(X)$components
```

person_extrapolated	<i>Person measures with extrapolated extreme scores</i>
---------------------	---

Description

Extreme persons (zero or maximum raw score on their observed items) are excluded from calibration, but they cannot be left out of group comparisons; Andrich and Marais (2019, ch. 10) therefore describe an extrapolated measure for them, continuing the growth of the score-to-score differences so the last difference is the geometric mean of its neighbours (see [score_table](#)). This helper applies the same rule to the person table: for each missing-data pattern with extreme persons, the score-to-measure conversion over that pattern's items is extrapolated at its ends, and the extreme persons receive the extrapolated location with the standard error $1/\sqrt{I(\theta)}$ evaluated there. Non-extreme persons keep their estimates unchanged. The extrapolation continues the Warm (weighted likelihood) conversion, matching the package's person estimates.

Usage

```
person_extrapolated(fit)
```

Arguments

`fit` A fitted object from [rasch](#) (equal discriminations; not EFRM).

Value

The fit's person table with two added columns, `theta_extrapolated` and `se_extrapolated`: equal to `theta` and `se` for non-extreme persons, extrapolated for extreme persons. Patterns with fewer than three interior scores cannot be extrapolated and keep their Warm values.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 8)
X <- matrix(rbinom(300 * 8, 1, plogis(outer(rnorm(300, 0, 2), d, "-"))), 300, 8)
colnames(X) <- paste0("I", 1:8)
fit <- rasch(X)
pe <- person_extrapolated(fit)
head(pe[pe$extreme, c("theta", "theta_extrapolated", "se", "se_extrapolated")])
```

person_wle

Warm's weighted likelihood estimates by raw score

Description

Computes the weighted likelihood estimate (WLE) of person location for every possible raw score on a set of items, with standard errors. WLE estimates are finite at the extreme (zero and maximum) scores, unlike the maximum likelihood estimate.

Usage

```
person_wle(tau_list, disc = 1)
```

Arguments

`tau_list` List of per-item threshold vectors.

`disc` Common discrimination (frame unit) of the items; with a constant discrimination the raw score remains sufficient.

Details

For raw score R , let $E(\theta)$, $V(\theta)$, and $\mu_3(\theta)$ be the sums of the item expected scores, variances, and third central moments. The estimate solves Warm's weighted score equation

$$R - E(\theta) + \frac{\mu_3(\theta)}{2V(\theta)} = 0.$$

With common discrimination d , its explicit multiplier cancels from this equation, although the moments are evaluated under d . The reported standard error is

$$SE(\hat{\theta}) = \{d^2 V(\hat{\theta})\}^{-1/2}.$$

Value

A list with theta and se, each named by raw score.

References

Warm, T. A. (1989). Weighted likelihood estimation of ability in item response theory. *Psychometrika*, 54(3), 427–450.

See Also

[score_table](#) and [person_extrapolated](#).

Examples

```
person_wle(list(c(-1, 0), c(-0.5, 0.5), c(0, 1)))
```

plot_btl

Plot Bradley-Terry-Luce object locations

Description

Caterpillar plot of object locations with 95 per cent error bars. Objects beyond the specified fit-residual band are marked.

Usage

```
plot_btl(fit, band = 2.5)
```

Arguments

fit	An object from btl .
band	Absolute fit-residual value beyond which an object is highlighted.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
beta <- c(A = -1, B = -0.3, C = 0.4, D = 0.9)
pairs <- t(combn(names(beta), 2))
d <- data.frame(a = rep(pairs[, 1], each = 30),
                b = rep(pairs[, 2], each = 30))
p <- plogis(beta[d$a] - beta[d$b])
d$win <- ifelse(runif(nrow(d)) < p, d$a, d$b)
plot_btl(btl(d, "a", "b", "win"))
```

plot_btl_categories *Plot polytomous-comparison category curves*

Description

For a polytomous paired-comparison fit, the probability of each response category as a function of the location difference $\beta_a - \beta_b$, with the symmetric threshold structure marked. The display is the paired-comparison counterpart of the category probability curves of a polytomous item.

Usage

```
plot_btl_categories(fit, grid = seq(-4, 4, 0.05))
```

Arguments

fit	A polytomous fit from <code>btl</code> (with response).
grid	Difference grid, in logits.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
beta <- c(A = -1, B = -0.3, C = 0.4, D = 0.9)
pr <- t(combn(names(beta), 2))
d <- data.frame(a = rep(pr[, 1], each = 40), b = rep(pr[, 2], each = 40))
P <- vapply(seq_len(nrow(d)), function(r)
  item_moments(beta[d$a[r]] - beta[d$b[r]], c(-1, 0, 1))$P, numeric(4))
d$grade <- apply(P, 2, function(p) sample(0:3, 1, prob = p))
plot_btl_categories(btl(d, "a", "b", response = "grade"))
```

plot_btl_dependence *Plot a within-judge dependence effect*

Description

The graphical display of a paired-comparison dependence effect, the counterpart of the DIF characteristic curve. For every comparison the departure of the observed response from what the object locations alone predict is taken, and the contribution of the *other* dependence effect is removed (a partial-residual display); these departures are then averaged in bins of the effect's own history covariate and plotted against it, with the model's fitted contribution overlaid. Observed points that rise with the covariate along the fitted line are the effect the coefficient summarises; a flat, scattered cloud means the estimate rests on little. Only the informative comparisons (a non-zero covariate: the two objects' histories differ) carry the effect, and the count in each bin is printed so a thin exposure tail is visible.

Usage

```
plot_btl_dependence(fit, effect = c("exposure", "carry_over"), bins = 6)
```

Arguments

fit	An object from <code>btl</code> fitted with an order column, so <code>fit\$dependence_data</code> is present.
effect	Which effect to display: "exposure" (the seen-before advantage, the default) or "carry_over" (response dependence).
bins	Number of covariate bins for the continuous carry-over display; exposure takes its three natural levels (-1, 0, +1).

Value

Called for its plotting side effect; invisibly a data frame of the binned covariate value, observed and fitted departure, and bin count.

References

Davidson, R. R., & Beaver, R. J. (1977). On extending the Bradley-Terry model to incorporate within-pair order effects. *Biometrics*, 33(4), 693-702.

Examples

```
set.seed(1)
beta <- c(A = -0.8, B = -0.2, C = 0.4, D = 0.9)
pr <- t(combn(names(beta), 2))
d <- data.frame(a = rep(pr[, 1], each = 40), b = rep(pr[, 2], each = 40))
d$judge <- sample(sprintf("J%02d", 1:8), nrow(d), TRUE)
d <- d[order(d$judge), ]; d$t <- ave(seq_len(nrow(d)), d$judge, FUN = seq_along)
d$win <- ifelse(runif(nrow(d)) < plogis(beta[d$a] - beta[d$b]), d$a, d$b)
f <- btl(d, "a", "b", winner = "win", judge = "judge", order = "t")
plot_btl_dependence(f, "carry_over")
```

plot_btl_dim_map

Residual map of the leading paired-comparison dimension

Description

Objects placed in the leading dimension plane. Reading round the swirl, an object sits “upstream” of those it over-beats relative to the fitted locations; a clear rotational arrangement is the second attribute, a formless blob near the origin is noise. Point size grows with the object’s location on the primary scale.

Usage

```
plot_btl_dim_map(x, ...)
```

Arguments

`x` A "rasch_btl_dim" object.
`...` Unused.

Value

Called for its plotting side effect.

Examples

```
d <- simulate_btl(7, 12, reps_per_pair = 20, seed = 1)
fit <- btl(d, "object_a", "object_b", winner = "winner", judge = "judge")
dimensions <- btl_dimensionality(fit, reps = 20)
plot_btl_dim_map(dimensions)
```

plot_btl_equate	<i>Plot a paired-comparison equating comparison</i>
-----------------	---

Description

Scatter of the two calibrations' common-object locations with the shifted identity line, per-object 95 per cent error bars, and a dotted guide band at the average pooled precision; objects that drift (after the multiplicity adjustment) are highlighted and labelled. The counterpart of [plot_equate](#) for Bradley-Terry-Luce scales.

Usage

```
plot_btl_equate(fit1, fit2, ...)
```

Arguments

`fit1` A fitted object from [btl](#).
`fit2` A second [btl](#) fit, or a bank data frame with columns object, location, and optionally se.
`...` Passed to [btl_equate](#) (e.g. alpha, p_adjust).

Value

Called for its plotting side effect; invisibly the [btl_equate](#) result.

Examples

```
set.seed(1)
beta <- setNames(seq(-2, 2, length.out = 8), paste0("0", 1:8))
sim <- function(objs) {
  pr <- t(utils::combn(objs, 2))
  d <- data.frame(a = rep(pr[, 1], each = 40), b = rep(pr[, 2], each = 40))
  d$win <- ifelse(runif(nrow(d)) < plogis(beta[d$a] - beta[d$b]), d$a, d$b)
  btl(d, "a", "b", "win")
}
plot_btl_equate(sim(paste0("0", 1:7)), sim(paste0("0", 2:8)),
  independent = TRUE)
```

plot_btl_icc

Plot an object characteristic curve

Description

Plots the expected response for one object against opponent location, with the observed mean response against each sufficiently observed opponent. For dichotomous fits the curve is the win probability; for ordered fits it is the expected response.

Usage

```
plot_btl_icc(fit, object, group = NULL, grid = NULL, min_n = 10)
```

Arguments

fit	An object from btl .
object	Object name.
group	Optional judge grouping for a DIF overlay: either one value per comparison row of fit\$comparisons or a vector named by judge. Observed means are then drawn separately per group, as plot_icc draws person groups.
grid	Opponent-location grid, in logits.
min_n	An opponent's observed point is drawn only when the object (or, in the grouped display, that judge group) met it at least this many times; sparser pairs from incomplete or unbalanced designs are omitted.

Value

Called for its plotting side effect; invisibly the names of the opponents drawn (the ungrouped display), or NULL for the grouped display.

Examples

```
set.seed(1)
beta <- c(A = -1, B = -0.3, C = 0.4, D = 0.9)
pr <- t(combn(names(beta), 2))
d <- data.frame(a = rep(pr[, 1], each = 30), b = rep(pr[, 2], each = 30))
p <- plogis(beta[d$a] - beta[d$b])
d$win <- ifelse(runif(nrow(d)) < p, d$a, d$b)
plot_btl_icc(btl(d, "a", "b", winner = "win"), "C")
```

plot_btl_judge_map	<i>Unexpected-judgement map for one judge (pair level)</i>
--------------------	--

Description

The judge counterpart of the kidmap, drawn matchup by matchup. Each pair the judge met is a segment on the consensus location axis, spanning its two objects, positioned horizontally by how surprising the verdict was: at zero (the dashed line, inside the shaded band) the stronger object won as its lead predicts; to the left the judge backed the underdog. A filled dot marks the object the judge's verdict favoured, hollow the other – so an upset is a red segment on the left with its filled dot at the lower end. The rug marks every object's location.

Usage

```
plot_btl_judge_map(fit, judge, min_n = 1L, flag_z = 1.96, ...)
```

Arguments

fit	A paired-comparison fit from btl with judges.
judge	The judge to map.
min_n, flag_z	Passed to judge_pair_surprise .
...	Unused.

Value

Called for its plotting side effect; invisibly the `rasch_btl_judge_pairs` object.

Examples

```
d <- simulate_btl(6, 10, reps_per_pair = 20, seed = 1)
fit <- btl(d, "object_a", "object_b", winner = "winner", judge = "judge")
plot_btl_judge_map(fit, judge = "J1")
```

plot_btl_scee	<i>Scree of paired-comparison residual bimensions</i>
---------------	---

Description

Bimension strengths against the model-simulated noise reference (its mean and 95th percentile band). A leading bar clearing the band is structured residual dependence – a likely second attribute.

Usage

```
plot_btl_scee(x, ...)
```

Arguments

x	A "rasch_btl_dim" object.
...	Unused.

Value

Called for its plotting side effect.

Examples

```
d <- simulate_btl(7, 12, reps_per_pair = 20, seed = 1)
fit <- btl(d, "object_a", "object_b", winner = "winner", judge = "judge")
dimensions <- btl_dimensionality(fit, reps = 20)
plot_btl_scee(dimensions)
```

plot_btl_targeting	<i>Targeting plot for a paired-comparison design</i>
--------------------	--

Description

The paired-comparison counterpart of a test-information display. Every object is a dot at its location (x) and its design information (y, the pooled Fisher information of the comparisons it took part in), the dot sized by how many comparisons that is. For an equal-unit fit, a reference curve on the right axis traces the information a single *new* comparison would carry against an opponent at each location. It peaks at gap zero and explains why adaptive designs favour near neighbours. A frame fit has no single reference curve because the information also depends on the fitted panel and set units.

Usage

```
plot_btl_targeting(fit, grid = NULL)
```

Arguments

fit	A paired-comparison fit from btl .
grid	Optional location grid for the equal-unit reference curve.

Value

Called for its plotting side effect; invisibly NULL.

See Also

[btl_information](#), [btl_next_pairs](#)

Examples

```
set.seed(1)
beta <- c(A = -1, B = -0.3, C = 0.4, D = 0.9)
pr <- t(combn(names(beta), 2))
d <- data.frame(a = rep(pr[, 1], each = 30), b = rep(pr[, 2], each = 30))
d$win <- ifelse(runif(nrow(d)) < plogis(beta[d$a] - beta[d$b]), d$a, d$b)
plot_btl_targeting(btl(d, "a", "b", "win"))
```

plot_btl_transitivity *Consistency plot for paired-comparison transitivity*

Description

With by = "judge" (the default when judges exist), plots each judge's consistency – one minus the circular-triad rate over the chance rate – as a dot against the chance line at zero: the individual-judge lens, a judge-fit analogue. With by = "object", plots each object's circular-triad involvement instead: the structural lens, showing which objects sit in the most contradictions.

Usage

```
plot_btl_transitivity(x, by = c("auto", "judge", "object"), ...)
```

Arguments

x	A "rasch_btl_transitivity" object.
by	"auto" (judges if present, else objects), "judge", or "object".
...	Unused.

Value

Called for its plotting side effect.

Examples

```
d <- simulate_btl(6, 10, reps_per_pair = 20, seed = 1)
fit <- btl(d, "object_a", "object_b", winner = "winner", judge = "judge")
tr <- btl_transitivity(fit)
plot_btl_transitivity(tr)
```

plot_btl_units	<i>Plot the frame units of a paired-comparison EFRM fit</i>
----------------	---

Description

Caterpillar plot of panel units `phi_g` and set units `alpha_s` on the log scale, with 95 per cent intervals and unit one marked.

Usage

```
plot_btl_units(fit)
```

Arguments

`fit` A fitted object from [btl_efrm](#).

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
# see ?btl_efrm for a complete simulated example
```

plot_catfreq	<i>Plot category frequencies</i>
--------------	----------------------------------

Description

Observed response distribution over the categories of one item.

Usage

```
plot_catfreq(fit, item)
```

Arguments

`fit` A fitted object from [rasch](#).
`item` Item name or column index.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
simP <- function(th, t) { x <- 0:length(t); p <- exp(x * th - c(0, cumsum(t))); p / sum(p) }
th <- rnorm(400)
X <- sapply(1:4, function(i)
  sapply(th, function(t)
    sample(0:3, 1, prob = simP(t, c(-1, 0, 1)))))
colnames(X) <- sprintf("P%02d", 1:4)
plot_catfreq(rasch(X), "P01")
```

plot_ccc

Plot category probability curves

Description

Plot category probability curves

Usage

```
plot_ccc(
  fit,
  item,
  grid = seq(-6, 6, 0.05),
  observed = FALSE,
  n_groups = fit$n_groups
)
```

Arguments

fit	A fitted object from rasch .
item	Item name or column index.
grid	Logit grid over which to draw the curves.
observed	Whether to add category proportions by class interval (Andrich and Marais 2019, ch. 20).
n_groups	Class intervals for the observed points.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```

set.seed(1)
simP <- function(th, t) { x <- 0:length(t); p <- exp(x * th - c(0, cumsum(t))); p / sum(p) }
th <- rnorm(400)
X <- sapply(1:4, function(i)
  sapply(th, function(t)
    sample(0:3, 1, prob = simP(t, c(-1, 0, 1)))))
colnames(X) <- sprintf("P%02d", 1:4)
plot_ccc(rasch(X), "P01", observed = TRUE)

```

plot_distractors	<i>Plot multiple-choice option curves</i>
------------------	---

Description

The proportion choosing each response option across class intervals of the rest measure (the person estimate from the other items), with the keyed option drawn solid and bold. The keyed option should rise with the trait and every distractor should fall; a rising distractor is the graphical signature of a miskey or an ambiguous option.

Usage

```
plot_distractors(fit, item, n_groups = fit$n_groups)
```

Arguments

fit	A fitted object from rasch run with a key.
item	Keyed item name.
n_groups	Number of class intervals.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```

set.seed(1); Np <- 400
th <- rnorm(Np)
raw <- sapply(seq(-1, 1, length.out = 6), function(d) {
  ok <- rbinom(Np, 1, plogis(th - d))
  ifelse(ok == 1, "A", sample(c("B", "C", "D"), Np, replace = TRUE))
})
colnames(raw) <- paste0("M", 1:6)
fit <- rasch(raw, key = setNames(rep("A", 6), colnames(raw)))
plot_distractors(fit, "M3")

```

plot_equate	<i>Plot a test-equating comparison</i>
-------------	--

Description

Scatter of the two calibrations' common-item locations with the shifted identity line and per-item 95 per cent bands; drifting items (Holm-adjusted) are highlighted and labelled.

Usage

```
plot_equate(fit, reference, shift = c("mean", "none"), independent = NULL)
```

Arguments

fit	A fitted object from rasch .
reference	A second rasch fit, or a data frame with columns item, location, and optionally se; a polytomous bank also needs max.
shift	Passed to equate_tests .
independent	Passed to equate_tests .

Value

Called for its plotting side effect; invisibly the [equate_tests](#) result.

Examples

```
set.seed(1); d <- seq(-1.5, 1.5, length.out = 8)
mk <- function() {
  X <- matrix(rbinom(400 * 8, 1, plogis(outer(rnorm(400), d, "-"))), 400, 8)
  colnames(X) <- paste0("I", 1:8); rasch(X)
}
plot_equate(mk(), mk(), independent = TRUE)
```

plot_facets	<i>Plot facet severities</i>
-------------	------------------------------

Description

Caterpillar plot of the severity of each level of a facet from a many-facet analysis, with 95 per cent error bars; levels with pooled fit residuals beyond the band are highlighted.

Usage

```
plot_facets(fit, facet = NULL, band = 2.5)
```

Arguments

fit	A fitted object from rasch_mfrm .
facet	Facet name; defaults to the first facet.
band	Fit residual band beyond which a level is highlighted.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
simP <- function(th, tau) {
  x <- 0:length(tau)
  p <- exp(x * th - c(0, cumsum(tau)))
  p / sum(p)
}
persons <- sprintf("P%03d", 1:120); raters <- paste0("R", 1:4)
th <- setNames(rnorm(120, 0, 1.3), persons)
rho <- setNames(c(-0.6, -0.2, 0.2, 0.6), raters)
tau <- list(A = c(-1, 1), B = c(-0.5, 1.2), C = c(-1.2, 0.4))
d <- expand.grid(person = persons, item = names(tau), rater = raters,
  stringsAsFactors = FALSE)
d$score <- mapply(function(p, i, r)
  sample(0:2, 1, prob = simP(th[p], tau[[i]] + rho[r])),
  d$person, d$item, d$rater)
plot_facets(rasch_mfrm(d, "person", "item", "score", facets = "rater"))
```

plot_frames

Plot frame units

Description

Caterpillar plot of the frame units $\rho_{sg} = \alpha_s \phi_g$ on the log scale, grouped by item set and coloured by person group, with 95 per cent error bars; frames with pooled fit residuals beyond the band are highlighted.

Usage

```
plot_frames(fit, band = 2.5)
```

Arguments

fit	A fitted object from rasch_efrm .
band	Pooled fit residual band beyond which a frame is highlighted.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
# see ?rasch_efrm for a complete simulated example
```

plot_guttman

Plot the Guttman scalogram

Description

Displays the dichotomous Guttman-ordered response matrix as a heatmap (dark for a score of one), with persons sorted by location down the rows and items by location across the columns. The coefficient of reproducibility is shown in the subtitle.

Usage

```
plot_guttman(fit, max_persons = 80)
```

Arguments

fit	A fitted object from rasch .
max_persons	Persons are thinned to at most this many evenly spaced rows for legibility on large samples.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(200 * 6, 1, plogis(outer(rnorm(200), d, "-"))), 200, 6)
colnames(X) <- paste0("I", 1:6)
plot_guttman(rasch(X))
```

plot_icc	<i>Plot an item characteristic curve</i>
----------	--

Description

Draws the model expected-score curve with observed class-interval means overlaid. Several items may be drawn together; their expected scores are then expressed as proportions of their maximum scores. With group supplied, observed means are drawn separately per group, the conventional graphical DIF display. For an MFRM fit, a single item may be named; its observed item-by-facet response cells are aligned by their fitted facet and interaction shifts before the class-interval means are formed.

Usage

```
plot_icc(
  fit,
  item,
  group = NULL,
  n_groups = NULL,
  grid = seq(-5, 5, 0.05),
  observed = TRUE
)
```

Arguments

fit	A fitted object from rasch .
item	One or more item names or column indices. Up to eight items may be overlaid. A group overlay requires a single item.
group	Optional person grouping vector, or one or more names of factors nominated in the fit, for a DIF overlay; several names give the factor-combination cells (the factorial display).
n_groups	Number of class intervals for the observed means; by default the fit's own count, or – with a group overlay – a count adapted to keep the smallest group's interval cells adequately filled.
grid	Logit grid over which to draw the model curve.
observed	Whether to add observed class-interval means.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
```

```
colnames(X) <- sprintf("I%02d", 1:6)
plot_icc(rasch(X), "I03")
```

plot_icc_frames

Plot an item's characteristic curves across frames

Description

Plots the model expected-score curve for one item in each frame, with observed class-interval means overlaid. Differences between the model curves reflect the fitted frame units. A nominated non-frame person factor separates the observed means within each frame for a DIF display.

Usage

```
plot_icc_frames(
  fit,
  item,
  n_groups = fit$n_groups,
  grid = seq(-5, 5, 0.05),
  group = NULL
)
```

Arguments

fit	A fitted object from rasch_efrm .
item	Underlying item name.
n_groups	Number of class intervals for the observed means.
grid	Logit grid.
group	Optional person grouping vector, or one or more names of non-frame factors nominated in the fit. Several names define their factor-combination cells.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
# see ?rasch_efrm for a complete simulated example
```

plot_item_map	<i>Plot the item map (location against fit residual)</i>
---------------	--

Description

Fitted columns plotted by location and fit residual, with the conventional acceptance band at ± 2.5 . MFRM and EFRM points are response cells; ordinary Rasch points are items.

Usage

```
plot_item_map(fit, band = 2.5)
```

Arguments

fit	A fitted object from rasch .
band	Fit residual acceptance band.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
plot_item_map(rasch(X))
```

plot_kidmap	<i>Plot a kidmap</i>
-------------	----------------------

Description

The person diagnostic map (Wright, Mead and Ludlow 1980): item thresholds the person achieved to the right of a vertical logit axis and thresholds not achieved to the left, with the person's location drawn as a dashed line inside its confidence band. Achieved thresholds above the band (unexpected successes) and unachieved thresholds below it (unexpected failures) are highlighted; a clean response pattern shows achieved thresholds below the band and unachieved ones above it.

Usage

```
plot_kidmap(
  fit,
  person,
  level = 0.95,
  bins = 35,
  xlim = NULL,
  cex_labels = 0.8
)
```

Arguments

fit	A fitted object from rasch .
person	Row number of the person, or an ID matching fit\$person\$id.
level	Confidence level of the band around the person location used to mark unexpected responses.
bins	Number of vertical bins used to stack the threshold labels.
xlim	Optional logit range; thresholds outside it are omitted.
cex_labels	Character expansion for the threshold labels.

Value

Called for its plotting side effect; invisibly NULL.

References

Wright, B. D., Mead, R. J., & Ludlow, L. H. (1980). *KIDMAP: person-by-item interaction mapping* (Research Memorandum No. 29). Chicago: University of Chicago, MESA Psychometric Laboratory.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 12)
X <- matrix(rbinom(300 * 12, 1, plogis(outer(rnorm(300), d, "-"))), 300, 12)
colnames(X) <- paste0("I", 1:12)
plot_kidmap(rasch(X), person = 1)
```

plot_pca

Plot residual principal-component loadings

Description

Residual-component loadings against item location. Opposing clusters suggest a further dimension. Any returned component may be plotted.

Usage

```
plot_pca(fit, component = 1)
```

Arguments

fit A fitted object from [rasch](#).
component Which residual principal component to plot (default the first component).

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
plot_pca(rasch(X))
```

plot_pca_biplot	<i>Biplot of the first two residual components</i>
-----------------	--

Description

Item loadings on the first two residual principal components – the pair that usually carries any interpretable second dimension – plotted against one another on equal (isometric) axes. Items far from the origin with opposing signs on PC1 mark a possible contrast, and PC2 separates them further. Point colour follows the sign of the PC1 loading, the split the unidimensionality t-test uses by default.

Usage

```
plot_pca_biplot(fit)
```

Arguments

fit A fitted object from [rasch](#).

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 8)
X <- matrix(rbinom(500 * 8, 1, plogis(outer(rnorm(500), d, "-"))), 500, 8)
colnames(X) <- paste0("I", 1:8)
plot_pca_biplot(rasch(X))
```

plot_pcc	<i>Plot a person characteristic curve</i>
----------	---

Description

The person characteristic curve: the probability of success as a function of item location at the person's estimated measure, with the person's observed responses overlaid, grouped into item-difficulty intervals (proportion of maximum score per interval). Erratic responding (for example lucky guessing on hard items by a low-proficiency person) shows as observed points far from the curve, complementing the person fit residual.

Usage

```
plot_pcc(fit, person, n_groups = 5, grid = seq(-5, 5, 0.05))
```

Arguments

fit	A fitted object from rasch .
person	Row number of the person, or an ID matching fit\$person\$id.
n_groups	Number of item-difficulty intervals for the observed points (capped by the number of observed items).
grid	Item-location grid over which to draw the curve.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 12)
X <- matrix(rbinom(300 * 12, 1, plogis(outer(rnorm(300), d, "-"))), 300, 12)
colnames(X) <- paste0("I", 1:12)
plot_pcc(rasch(X), person = 1)
```

plot_person_fit	<i>Plot person fit</i>
-----------------	------------------------

Description

Person locations against person fit residuals with the +/- 2.5 band; persons beyond the band respond erratically (positive) or too deterministically (negative).

Usage

```
plot_person_fit(fit, band = 2.5)
```

Arguments

fit	A fitted object from rasch .
band	Fit residual acceptance band.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
plot_person_fit(rasch(X))
```

plot_pimap	<i>Plot the person-item threshold distribution</i>
------------	--

Description

The targeting display: the person location distribution above the axis and the calibration threshold distribution mirrored below it, on a shared logit scale. MFRM and EFRM thresholds belong to response cells.

Usage

```
plot_pimap(fit, bins = 35, xlim = NULL, information = FALSE)
```

Arguments

fit	A fitted object from rasch .
bins	Number of histogram bins.
xlim	Optional logit range for the shared scale; persons and thresholds outside it are omitted. By default the range is extended to labelled tick marks beyond the most extreme plotted estimate.
information	Whether to overlay the test information function on a separate right-hand axis. Fits with more than one administrable design receive one curve per design.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
plot_pimap(rasch(X))
```

plot_recovery

Plot fitted against generating parameters

Description

One true-versus-estimated panel per parameter type, with the identity line and the correlation and RMSE.

Usage

```
plot_recovery(x, ...)
```

Arguments

```
x          A "rasch_recovery" object.
...        Unused.
```

Value

Called for its plotting side effect.

Examples

```
d <- simulate_rasch(300, 8, seed = 1)
fit <- rasch(d, id = "id")
plot_recovery(sim_recovery(fit, d))
```

plot_resid_cor

Plot the residual-correlation heatmap

Description

Only the lower triangle is drawn – the matrix is symmetric, so each pair is shown once. With `stat = "q3star"` (the default) cells are coloured by $Q3^*$ – each pair's residual correlation minus the average off-diagonal correlation – so white marks the value expected under local independence and warm colour marks dependence; with `stat = "q3"` the raw residual correlation is coloured, white at zero. The scale saturates at cap rather than the ± 1 of an ordinary correlation: a residual correlation seldom reaches even 0.5 under a fitting model, so the colour is spent where the values actually discriminate. A $Q3^*$ value of 0.2 is sometimes used as a heuristic screen, but it is not a universal critical value (Christensen, Makransky and Horton 2017).

Usage

```
plot_resid_cor(fit, stat = c("q3star", "q3"), cap = 0.5)
```

Arguments

fit	A fitted object from rasch .
stat	Which statistic to colour: "q3star" (adjusted Q3, the default) or "q3" (the raw residual correlation).
cap	Value at which the colour saturates (default 0.5).

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
plot_resid_cor(rasch(X))
```

plot_resid_dist	<i>Plot the fit residual distribution</i>
-----------------	---

Description

A histogram of the item or person fit residuals – the log-transformed statistic or its untransformed natural form – against the standard normal density they should approximate under fit (Andrich and Marais 2019, ch. 15). The natural residual is visibly skewed (that is why the log transform is reported); both are available.

Usage

```
plot_resid_dist(
  fit,
  what = c("items", "persons"),
  statistic = c("fit_resid", "natural"),
  bins = 25
)
```

Arguments

fit	A fitted object from rasch .
what	"items" or "persons".
statistic	"fit_resid" (log-transformed, default) or "natural".
bins	Number of histogram bins.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 10)
X <- matrix(rbinom(400 * 10, 1, plogis(outer(rnorm(400), d, "-"))), 400, 10)
colnames(X) <- paste0("I", 1:10)
plot_resid_dist(rasch(X), what = "persons")
```

plot_scee

Scree plot of the residual components with parallel analysis

Description

Eigenvalues of the residual correlation matrix for the leading components, with a model-simulated parallel-analysis reference: responses are simulated from the calibrated model (observed missingness kept), the item calibration and every person are re-estimated, and the residual eigenvalues recomputed. Because estimating the person locations couples the residuals within a person, this reference sits above the classical random-normal one and is calibrated under the fitted model (Raiche 2005; Chou & Wang 2010). Observed eigenvalues above the reference suggest structure beyond what the model itself produces.

Usage

```
plot_scee(fit, n_components = 10, parallel = TRUE, reps = 50)
```

Arguments

<code>fit</code>	A fitted object from rasch .
<code>n_components</code>	Number of leading components to display.
<code>parallel</code>	Draw the parallel-analysis reference line.
<code>reps</code>	Model-simulated replicates for the reference.

Value

Called for its plotting side effect; invisibly the eigen table.

References

Raiche, G. (2005). Critical eigenvalue sizes (variances) in standardized residual principal components analysis. *Rasch Measurement Transactions*, 19(1), 1012.

Chou, Y.-T., & Wang, W.-C. (2010). Checking dimensionality in item response models with principal component analysis on standardized residuals. *Educational and Psychological Measurement*, 70(5), 717-731.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 8)
X <- matrix(rbinom(500 * 8, 1, plogis(outer(rnorm(500), d, "-"))), 500, 8)
colnames(X) <- paste0("I", 1:8)
plot_scee(rasch(X), reps = 20)
```

plot_tcc

*Plot the test characteristic curve***Description**

Expected total score against person location. Structural fits draw one curve for each administrable frame or facet design.

Usage

```
plot_tcc(fit, grid = seq(-6, 6, 0.05))
```

Arguments

fit	A fitted object from rasch .
grid	Logit grid.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
plot_tcc(rasch(X))
```

plot_threshold_map	<i>Plot the threshold map</i>
--------------------	-------------------------------

Description

Each fitted column's threshold locations on a common logit scale, ordered by location, with disordered thresholds highlighted. The columns are response cells for MFRM and EFRM fits.

Usage

```
plot_threshold_map(fit, order_by_location = TRUE)
```

Arguments

fit	A fitted object from rasch .
order_by_location	Order items by their location (the default) rather than their original sequence.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
plot_threshold_map(rasch(X))
```

plot_threshold_prob	<i>Plot threshold probability curves</i>
---------------------	--

Description

Conditional probability of success at each threshold, $P(X = k \mid X = k - 1 \text{ or } k)$, a logistic ogive crossing 0.5 at the threshold location. Disordered thresholds are immediately visible as out-of-sequence ogives. With observed = TRUE the observed conditional proportions per class interval are overlaid, the direct check on whether each threshold discriminates (and hence whether collapsing categories could ever be justified; Andrich and Marais 2019, ch. 22).

Usage

```
plot_threshold_prob(
  fit,
  item,
  grid = seq(-6, 6, 0.05),
  observed = FALSE,
  n_groups = fit$n_groups
)
```

Arguments

fit	A fitted object from rasch .
item	Item name or column index.
grid	Logit grid over which to draw the curves.
observed	Overlay the observed conditional threshold proportions per class interval.
n_groups	Class intervals for the observed points.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
simP <- function(th, t) { x <- 0:length(t); p <- exp(x * th - c(0, cumsum(t))); p / sum(p) }
th <- rnorm(400)
X <- sapply(1:4, function(i)
  sapply(th, function(t)
    sample(0:3, 1, prob = simP(t, c(-1, 0, 1)))))
colnames(X) <- sprintf("P%02d", 1:4)
plot_threshold_prob(rasch(X), "P01")
```

plot_tif

Plot the test information function

Description

Test information across the logit scale with the standard error of measurement overlaid on a second axis.

Usage

```
plot_tif(fit, grid = seq(-6, 6, 0.05))
```

Arguments

fit	A fitted object from rasch .
grid	Logit grid.

Value

Called for its plotting side effect; invisibly NULL.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
plot_tif(rasch(X))
```

plot_wright

Plot a Wright map

Description

The conventional vertical person-item map (Wright and Stone 1979): the person distribution to the left of a shared logit axis and the calibration thresholds stacked to its right. MFRM and EFRM labels identify item-by-facet or item-by-frame response cells rather than additional items.

Usage

```
plot_wright(fit, bins = 35, xlim = NULL, cex_labels = 0.8)
```

Arguments

fit	A fitted object from rasch .
bins	Number of bins for the person distribution and the threshold label rows.
xlim	Optional logit range for the shared scale; persons and thresholds outside it are omitted.
cex_labels	Character expansion for the threshold labels.

Value

Called for its plotting side effect; invisibly NULL.

References

Wright, B. D., & Stone, M. H. (1979). *Best Test Design*. Chicago: MESA Press.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
plot_wright(rasch(X))
```

rack_data

*Reshape repeated measurements for racked or stacked analysis***Description**

Repeated measurements (the same persons and items at two or more time points) enter a Rasch analysis in one of two designs (Andrich & Marais 2019, ch. 26). *Racking* keeps one row per person and duplicates the items per time point (columns `item@time`), so change over time shows in the item estimates. *Stacking* keeps one column per item and duplicates the persons per time point (rows), so change shows in the person estimates and DIF of items over time can be examined with `time` as a within-person factor. The returned `id` is the original person identifier and therefore repeats across occasions; `row_id` uniquely identifies each person-occasion row.

Usage

```
rack_data(data, person, time, items)
```

```
stack_data(data, person, time, items)
```

Arguments

<code>data</code>	A long data frame with one measurement per row.
<code>person, time</code>	Names of the person and time-point columns.
<code>items</code>	Character vector naming the item columns.

Value

`rack_data`: a wide data frame with one row per person and `length(items) * n_times` item columns. `stack_data`: a data frame with one row per person-time, the repeated original `id`, a unique `row_id`, the original item columns, and `time` as a factor column for repeated-measures DIF analysis.

Examples

```
d <- data.frame(pid = rep(1:100, 2), t = rep(1:2, each = 100),
               Q1 = rbinom(200, 1, 0.6), Q2 = rbinom(200, 1, 0.5))
racked <- rack_data(d, person = "pid", time = "t", items = c("Q1", "Q2"))
names(racked)
stacked <- stack_data(d, person = "pid", time = "t", items = c("Q1", "Q2"))
head(stacked)
# the follow-up analysis assigns every reshaped column a role: the
# repeated person id, time as a within-person factor, and the items
fit <- rasch(stacked, id = "id", factors = "time",
            items = c("Q1", "Q2"))
```

 rasch

Fit a Rasch model

Description

Fits the partial credit model (PCM) or rating scale model (RSM) by pairwise conditional maximum likelihood. Person locations are Warm weighted likelihood estimates. The fitted object contains item and person fit, targeting, reliability, threshold diagnostics, residuals, and a score-to-measure table.

Usage

```

rasch(
  data,
  model = c("PCM", "RSM"),
  id = NULL,
  factors = NULL,
  items = NULL,
  n_groups = NULL,
  adjust_N = NA,
  anchors = NULL,
  na_codes = -1,
  key = NULL,
  pc_components = NULL,
  maxit = 60,
  tol = 1e-08
)

```

Arguments

data	Persons-by-items integer score matrix (categories from 0), or a data frame also containing ID and person-factor columns. Missing values are allowed subject to the identification and ignorability conditions described above.
model	Either "PCM" (partial credit) or "RSM" (rating scale).
id	Optional name of an ID column in data, or a vector of IDs; carried through to the person estimates.
factors	Optional character vector of person-factor column names in data (for DIF analysis), a data frame of factors, or one grouping vector with one entry per data row.
items	Optional character vector naming the item columns; by default every column not named in id or factors.
n_groups	Number of class intervals for the item-trait chi-square and ANOVA item fit. The default NULL applies the rule of Andrich and Marais (2019, ch. 15): as many intervals of at least 50 non-extreme persons as the sample allows, at most 10, at least 2. The resolved value is stored in <code>fit\$n_groups</code> .

adjust_N	Optional reference sample size used to rescale the item-trait chi-squares. See Details.
anchors	Optional anchor table for equating: a data frame with columns <code>item</code> , <code>k</code> , and <code>tau</code> ; see pcml . Anchors determine the scale origin.
na_codes	Values to read as missing. Defaults to <code>-1</code> , the conventional missing-response code; any negative score is also treated as missing, since valid category scores start at zero.
key	Optional multiple-choice key: a named item-to-option vector, an item/key table, or an item/option/score table. See Details.
pc_components	NULL (the default) estimates all PCM thresholds freely. Values from 1 to 4 use the principal-components form in pcml_pc : location, then spread, skewness, and kurtosis. This can stabilise sparse categories. Component estimates are stored in the estimation details. Available for PCM fits without anchors.
maxit, tol	Newton-Raphson iteration cap and convergence tolerance of the pairwise conditional estimation.

Details

For scores $x = 0, \dots, m_i$, the PCM is

$$P(X_{ni} = x) = \frac{\exp\{x\theta_n - \sum_{k=1}^x \delta_{ik}\}}{\sum_{y=0}^{m_i} \exp\{y\theta_n - \sum_{k=1}^y \delta_{ik}\}}.$$

The RSM constrains $\delta_{ik} = \beta_i + \tau_k$, where β_i is the item location and τ_k is common across items. Dichotomous items are the one-threshold case of the PCM.

Pairwise conditioning removes θ_n from the item likelihood. Missing responses are omitted from pairwise contributions, and person measures are estimated within each observed item pattern. The observed item-pair graph must identify a common scale. This covers planned linked designs and ignorable missingness; informative missingness can still bias the estimates.

The fit residual is the log-of-mean-square statistic described by Andrich and Marais (2019, ch. 23). It is approximately standard normal under fit; positive values indicate under-discrimination and negative values indicate over-discrimination. The item-trait chi-square and class-interval F tests are large-sample diagnostic approximations and should be considered with the residual statistics, effect sizes, and item content.

Multiple-choice responses may be scored from a named item-to-key vector, an item/key table, or an item/option/score table. A slash separates alternative correct options. The third form assigns integer category scores to nominated options and fits the resulting item as polytomous; unlisted options score zero. Raw responses are retained in `fit$mc` for distractor analysis.

If `adjust_N` is supplied, each item-trait chi-square is multiplied by the reference sample size divided by the number of classified persons. The scaling is global: an item answered by a subset retains its proportionally smaller share of the reference sample.

Value

An object of class `"rasch"`. Its principal components are the item summary, threshold table, person table, score table, residuals, reliability, targeting, item-trait statistics, threshold diagnostics, and

estimation details. The component `summary_stats` contains the distribution summaries, fit-location correlations, and the cell degrees-of-freedom factor. The item summary carries a `disc` column described below.

Estimated item discrimination

The item summary includes a post-estimation slope `disc`. For item i , it maximises that item's response likelihood over a_i while holding the fitted person locations and thresholds fixed:

$$\hat{a}_i = \arg \max_{a_i} \sum_n \log P(X_{ni} = x_{ni} \mid \hat{\theta}_n, \hat{\delta}_i, a_i).$$

The same slope multiplies every threshold of a polytomous item. It is a descriptive index, not a freely estimated parameter of the Rasch model, and no sampling standard error or hypothesis test is attached to it.

Item-fit probabilities

The item-trait chi-square supplies the principal inferential test of invariance over class intervals. The class-interval ANOVA is a conventional residual diagnostic whose F reference is approximate. Its probability can be anti-conservative in short tests because each response contributes appreciably to the person grouping used to test that item. The same issue can affect the item-trait probability when fewer than about ten responses locate each person. In short administrations, read the statistics with the characteristic curve and residual fit rather than as stand-alone decisions.

References

- Rasch, G. (1960). Probabilistic Models for Some Intelligence and Attainment Tests. Copenhagen: Danish Institute for Educational Research. (Expanded edition, 1980, Chicago: University of Chicago Press.)
- Rasch, G. (1961). On general laws and the meaning of measurement in psychology. In Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability (Vol. 4, pp. 321–333). Berkeley: University of California Press.
- Andrich, D. and Luo, G. (2003). Conditional pairwise estimation in the Rasch model for ordered response categories using principal components. *Journal of Applied Measurement*, 4(3), 205–221.
- Andrich, D. and Marais, I. (2019). *A Course in Rasch Measurement Theory: Measuring in the Educational, Social and Health Sciences*. Springer.
- Warm, T. A. (1989). Weighted likelihood estimation of ability in item response theory. *Psychometrika*, 54(3), 427–450.

See Also

[rasch_mfrm](#), [rasch_efrm](#), [btl](#), [dif_anova](#), [test_information](#), and [run_app](#).

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 8)
X <- matrix(rbinom(500 * 8, 1, plogis(outer(rnorm(500), d, "-"))), 500, 8)
```

```
colnames(X) <- paste0("I", 1:8)
fit <- rasch(X, model = "PCM")
fit$items
fit$psi$PSI
```

rasch_efrm

Fit the extended frame of reference model

Description

Fits Humphry's extended frame of reference model, in which the unit can differ across item-set by person-group frames. For item i in set s and person n in group g ,

$$P(X_{ni} = x) = \frac{\exp\{\rho_{sg}[x\theta_n - \sum_{k=1}^x \delta_{ik}]\}}{\sum_{y=0}^{m_i} \exp\{\rho_{sg}[y\theta_n - \sum_{k=1}^y \delta_{ik}]\}}, \quad \rho_{sg} = \alpha_s \phi_g.$$

Usage

```
rasch_efrm(
  data,
  item_sets,
  groups,
  id = NULL,
  factors = NULL,
  items = NULL,
  n_groups = NULL,
  adjust_N = NA,
  na_codes = -1,
  maxit = 50,
  tol = 1e-07,
  min_link_persons = 30,
  se_method = c("hybrid", "bootstrap"),
  boot_reps = NULL,
  progress = NULL,
  cancel = NULL,
  workers = 4L,
  seed = NULL
)
```

Arguments

data	Persons-by-items data (matrix or data frame, like rasch), plus a person-group column.
item_sets	A named list mapping set names to item-column names, or a named character vector mapping item names to set names. Items not mentioned form their own set "(rest)" when a list is given.

groups	Name of the person-group column in data, or a vector with one entry per person. Several columns define crossed group cells. Their units are returned in <code>phi_table</code> ; <code>phi_factorial</code> and <code>phi_factorial_tests</code> contain the GLS factorial decomposition and omnibus Wald tests. Structurally unidentified units are refused. Very imprecise but identified units are retained with a warning.
id	Person identifier, either a column name or one value per row. EFRM data require one response row per person, so identifiers must be unique.
factors, items, n_groups, adjust_N, na_codes	As in rasch .
maxit, tol	Outer iteration cap and convergence tolerance of the bilinear pairwise stage.
min_link_persons	Minimum number of common persons required for a set pair to contribute to the unit linking.
se_method	"hybrid" (sandwich + linking bootstrap + delta propagation; default) or "bootstrap" (full person bootstrap of all stages).
boot_reps	Bootstrap replicates; defaults to 300 for the linking bootstrap and 200 for the full bootstrap. Use zero to omit unit uncertainty; otherwise at least 30 are required.
progress	Optional function called as <code>progress(stage, current, total)</code> during long uncertainty calculations. It is intended for interfaces and batch logging and does not alter estimation.
cancel	Optional zero-argument function checked between bootstrap batches. Returning TRUE stops with a <code>rasch_cancelled</code> condition. A serial fit uses one replicate per batch.
workers	Number of parallel bootstrap workers. The default is four, reduced when fewer physical cores are available or the R process has a lower system limit. Random samples are generated before distribution, so a fixed seed gives the same result for any worker count. Every worker holds its own copy of the bootstrap state.
seed	Optional bootstrap seed. The caller's random-number state is restored when estimation finishes.

Details

The partial credit model holds within each frame in its natural unit. Person-group units ϕ_g are identified from common item thresholds across groups. Item sets partition the items, so set units α_s are identified instead from persons observed in more than one set. The set-linking graph and the group-by-set frame graph must each connect to a common scale.

Set units use a semiparametric likelihood for persons observed in each linked pair of sets. For sets a and b , it maximises

$$\prod_n \int P(X_{na} | u) P(X_{nb} | ru + c) dF_{g(n)}(u),$$

where the masses of each observed group's F_g , the scale ratio r and the offset c are estimated jointly on a fixed grid. This avoids prescribing a normal or common person distribution across groups. The conditional thresholds and group units are held fixed in this step; only r , c , and the nuisance masses are estimated. The linked parameters are then

$$\delta_{ik} = \tilde{\delta}_{ik} / \alpha_s + \mu_s, \quad \rho_{sg} = \alpha_s \phi_g.$$

Score moments supply starting values and screen weak links. Response patterns must span a score range of at least four within a set. Overlapping item sets are not permitted. The public convergence flag covers both estimation stages; `stage1_converged` records the conditional stage separately.

The hybrid covariance combines the pairwise Godambe covariance with a person bootstrap for set linking. Each replicate jointly redraws the within-frame thresholds and group units, then rebuilds the link. The joint draws retain covariance among common-scale thresholds, set units and group units. With `se_method = "bootstrap"`, the complete model is refitted to each person resample.

The `efrm_vs_rasch` component records the within-frame composite log-likelihood comparison between group-dependent and equal group units. This difference is descriptive and contains no information about set units, which are identified at the linking stage. The accompanying Wald omnibus tests provide inference for the group- and set-unit families. Unit estimates are retained for sparse designs, but probabilities require at least 50 persons or effective persons in every group and at least 50 common persons on every set-link edge.

The model assumes that an item retains its location and discrimination across the frames in which it appears, apart from the frame unit. `frame_invariance` examines this assumption by separate frame calibrations. Misfit concentrated within one item set can also distort its estimated unit; inspect item fit and targeting before interpreting unit differences. `drop_items` and `resolve_frames` provide refitted sensitivity analyses.

The dichotomous model follows Humphry (2005) and Humphry and Andrich (2008). The polytomous, multigroup and crossed-frame forms are extensions implemented in this package. The discrete nonparametric margin follows the Rasch estimation approach of Follmann (1988); its use for linked item-set units is an extension implemented here.

Value

An object of classes `"rasch_efrm"` and `"rasch"`. Model-specific components include frames, `phi_table`, `alpha_table`, `set_table`, common-unit item and threshold tables, group-specific `score_curves`, `efrm_vs_rasch`, and linking, and the person support used for unit inference in `unit_support`. See the extended frame of reference vignette for their interpretation.

References

- Andrich, D. (1982). An extension of the Rasch model for ratings providing both location and dispersion parameters. *Psychometrika*, 47(1), 105–113.
- Andrich, D. and Luo, G. (2003). Conditional pairwise estimation in the Rasch model for ordered response categories using principal components. *Journal of Applied Measurement*, 4(3), 205–221.
- Andrich, D. and Marais, I. (2019). *A Course in Rasch Measurement Theory: Measuring in the Educational, Social and Health Sciences*. Springer.
- Follmann, D. (1988). Consistent estimation in the Rasch model based on nonparametric margins. *Psychometrika*, 53, 553–562. doi:10.1007/BF02294407
- Humphry, S. M. (2005). *Maintaining a Common Arbitrary Unit in Social Measurement*. PhD thesis, Murdoch University.
- Humphry, S. M. (2010). Modeling the effects of person group factors on discrimination. *Educational and Psychological Measurement*, 70(2), 215–231.
- Humphry, S. M. (2012). Item set discrimination and the unit in the Rasch model. *Journal of Applied Measurement*, 13(2), 165–180.

Montuoro, P. and Humphry, S. M. (2024). Modeling the effect of reading item clarity on item discrimination. *Journal of Applied Measurement*, 24(3/4), 121–132.

Humphry, S. M. and Andrich, D. (2008). Understanding the unit in the Rasch model. *Journal of Applied Measurement*, 9(3), 249–264.

See Also

[frame_invariance](#), which tests the item invariance this model assumes rather than imposing it, and [drop_items](#), which removes an item the test flags and refits. Also [rasch](#), [rasch_mfrm](#), [test_information](#), and [simulate_efrm](#).

Examples

```
set.seed(1); Np <- 400
simP <- function(th, tau, r) { x <- 0:length(tau)
  p <- exp(r * (x * th - c(0, cumsum(tau))))); p / sum(p) }
grp <- rep(c("A", "B"), each = Np / 2)
phi <- c(A = 0.8, B = 1.25)
d <- seq(-1.5, 1.5, length.out = 10)
theta <- rnorm(Np)
X <- sapply(seq_along(d), function(i) sapply(seq_len(Np), function(n)
  sample(0:1, 1, prob = simP(theta[n], d[i], phi[grp[n]]))))
colnames(X) <- sprintf("I%02d", seq_along(d))
fit <- rasch_efrm(data.frame(X, grp = grp), item_sets = list(core = colnames(X)),
  groups = "grp")
fit$phi_table
```

rasch_explanatory

Fit an explanatory Rasch model

Description

Fits the linear logistic test model (LLTM) for dichotomous responses or the linear partial credit model (LPCM) for polytomous responses. Item or threshold locations are linear functions of observed predictors. The response model remains Rasch and is estimated by pairwise conditional maximum likelihood.

Usage

```
rasch_explanatory(
  data,
  predictors,
  formula,
  items = NULL,
  level = c("item", "threshold"),
  id = NULL,
  factors = NULL,
```

```

n_groups = NULL,
adjust_N = NA,
na_codes = -1,
key = NULL,
maxit = 60,
tol = 1e-08
)

```

Arguments

`data`, `items`, `id`, `factors`, `n_groups`, `adjust_N`, `na_codes`, `key`, `maxit`, `tol`
As in [rasch](#).

`predictors` Data frame containing an item column and the predictors named in formula. With `level = "threshold"`, it must also contain `threshold`, with one row for every fitted item threshold.

`formula` One-sided explanatory formula. For example, `~ format + operation + format:operation`. The reserved threshold factor permits threshold-specific effects.

`level` Whether `predictors` contains one row per "item" or per "threshold". Item rows are expanded over their thresholds.

Details

For threshold k of item i ,

$$\delta_{ik} = z_{ik}^T \gamma.$$

The adjacent-category log odds are

$$\log\{P(X_{ni} = k)/P(X_{ni} = k - 1)\} = \theta_n - \delta_{ik}.$$

The threshold origin is fixed to the same mean-item-location zero used by [rasch](#). An intercept therefore sets the arbitrary origin and is not separately estimated. Numeric predictors are continuous, unordered factors are categorical, and ordered factors use successive contrasts between adjacent levels. Character predictors are converted to unordered factors. The reserved factor `threshold` identifies the within-item threshold number; `threshold_number` supplies its integer value.

A free PCM reference is fitted to the same prepared responses and retained on the object. [explanatory_test](#) applies the first-order Kent calibration required for the pairwise composite likelihood.

Value

An object of class "rasch_explanatory" inheriting from "rasch". Standard item, person, fit and diagnostic components use the explanatory thresholds. The explanatory component contains the formula, metadata and design matrices; `reference_fit` is the free PCM calibration.

References

- Fischer, G. H. (1973). The linear logistic test model as an instrument in educational research. *Acta Psychologica*, 37, 359–374.
- Fischer, G. H. and Ponocny, I. (1994). An extension of the partial credit model with an application to the measurement of change. *Psychometrika*, 59, 177–192.

See Also

[explanatory_test](#), [explanatory_diagnostics](#), and [relax_explanatory](#).

Examples

```
set.seed(1)
q <- data.frame(item = paste0("I", 1:8),
                operation = rep(0:1, each = 4),
                format = rep(c("A", "B"), 4))
difficulty <- -1 + 0.7 * q$operation + 0.4 * (q$format == "B")
X <- matrix(rbinom(500 * 8, 1,
  plogis(outer(rnorm(500), difficulty, "-"))), 500, 8)
colnames(X) <- q$item
fit <- rasch_explanatory(X, predictors = q,
                        formula = ~ operation + format)

fit$est$coefficients
explanatory_test(fit)
```

 rasch_mfrm

Fit a many-facet Rasch model

Description

Fits an additive many-facet Rasch model (Linacre 1989) to scored responses indexed by person, item, and one or more facets such as rater, task, or occasion. Facet severities, item thresholds, person locations, and fit statistics are reported on a common logit scale.

Usage

```
rasch_mfrm(
  data,
  person,
  item = NULL,
  score = NULL,
  facets,
  items = NULL,
  n_groups = NULL,
  adjust_N = NA,
  na_codes = -1,
  interaction = NULL,
  factors = NULL,
  maxit = 60,
  tol = 1e-08
)
```

Arguments

<code>data</code>	Long-format data frame.
<code>person</code>	Name of the person identifier column.
<code>item</code>	Name of the item column.
<code>score</code>	Name of the integer score column (categories from 0; gaps are collapsed per item with a note).
<code>facets</code>	Character vector naming one or more facet columns (for example a rater column).
<code>items</code>	Optional character vector of item score columns for data in wide format: one row per person-by-facet combination (for example one row per script per rater) with one column per item or criterion. The long form (<code>item + score</code>) remains available for data where the facet varies within items.
<code>n_groups</code>	Number of class intervals for the item-trait chi-square; NULL (the default) applies the class-interval rule of Andrich and Marais (2019, ch. 15) (at least 50 non-extreme persons per interval, at most 10 intervals, at least 2).
<code>adjust_N</code>	Optional reference sample size for the chi-square.
<code>na_codes</code>	Score values to read as missing (default -1); any negative score is also treated as missing.
<code>interaction</code>	Optional name of one facet to interact with the items (interactive facet mode). See Details.
<code>factors</code>	Optional person factors for DIF analysis: a character vector naming columns constant within person, or a data frame with one row per data row or unique person. Facets belong in <code>facets</code> , not here.
<code>maxit, tol</code>	Newton-Raphson iteration cap and convergence tolerance.

Details

For person n , item i , and facet levels $f_1, \dots, f_Q$, the additive model is

$$P(X_{nif} = x) = \frac{\exp\{x\theta_n - \sum_{k=1}^x [\delta_{ik} + \sum_{q=1}^Q \rho_{qf_q}]\}}{\sum_{y=0}^{m_i} \exp\{y\theta_n - \sum_{k=1}^y [\delta_{ik} + \sum_{q=1}^Q \rho_{qf_q}]\}}.$$

Positive facet values therefore denote greater severity. The item thresholds have a common sum-zero origin and the levels of each facet sum to zero. If `interaction` is requested, an item-by-level term is added with both its item and facet margins constrained to sum to zero.

Estimation represents each observed item-by-facet combination as a virtual item and imposes the additive structure in the pairwise conditional likelihood. The person parameter cancels before calibration. The covariance of the structural parameters is the transformed Godambe sandwich covariance.

Facet levels must be connected through common persons and items. A facet nested within an item or a person-disjoint block can be confounded with the item location. The function checks the structural rank and response graph before fitting the model.

An item-by-facet interaction retains equal discrimination but allows facet differences to vary by item. The omnibus Wald test in `interaction_test` is the primary test; cell tests are Holm-adjusted

follow-ups. Interaction probabilities require at least $\max\{30, q + 2\}$ persons and effective persons at every level of the interactive facet, where q is the omnibus degrees of freedom. Estimates remain descriptive when this condition is not met.

Value

An object of classes "rasch_mfrm" and "rasch". Model-specific components describe the facets, items, thresholds, and facet specification. Interactive fits also contain an omnibus test and the corresponding item-by-facet effects. The component `fit_resid` averages virtual-item residuals within a margin. Its response-weighted counterpart is `fit_resid_pooled`; its degrees of freedom are in `df_fit`.

References

Andrich, D. and Marais, I. (2019). A Course in Rasch Measurement Theory: Measuring in the Educational, Social and Health Sciences. Springer.

Linacre, J. M. (1989). Many-Facet Rasch Measurement. Chicago: MESA Press.

See Also

[rasch](#), [rasch_efrm](#), [dif_anova](#), and [simulate_mfrm](#).

Examples

```
set.seed(1)
simP <- function(th, tau) {
  x <- 0:length(tau)
  p <- exp(x * th - c(0, cumsum(tau)))
  p / sum(p)
}
persons <- sprintf("P%03d", 1:120); raters <- paste0("R", 1:4)
th <- setNames(rnorm(120, 0, 1.3), persons)
rho <- setNames(c(-0.6, -0.2, 0.2, 0.6), raters)
tau <- list(A = c(-1, 1), B = c(-0.5, 1.2), C = c(-1.2, 0.4))
d <- expand.grid(person = persons, item = names(tau), rater = raters,
  stringsAsFactors = FALSE)
d$score <- mapply(function(p, i, r)
  sample(0:2, 1, prob = simP(th[p], tau[[i]] + rho[r])),
  d$person, d$item, d$rater)
fit <- rasch_mfrm(d, person = "person", item = "item", score = "score",
  facets = "rater")
fit$facet_effects$rater
```

`relax_btl_explanatory` *Add a fixed object departure to an explanatory comparative judgement model*

Description

Add a fixed object departure to an explanatory comparative judgement model

Usage

```
relax_btl_explanatory(fit, object)
```

Arguments

fit	A fitted object from btl_explanatory .
object	Object name.

Value

A refitted explanatory comparative judgement model.

relax_explanatory	<i>Relax a nominated explanatory restriction</i>
-------------------	--

Description

Adds either one fixed item-location departure or a fixed block describing an item's threshold structure, then repeats the complete conditional calibration and downstream Rasch analysis. The departure is fixed rather than random; raw-score sufficiency and the common discrimination remain.

Usage

```
relax_explanatory(fit, item, component = c("location", "thresholds"))
```

Arguments

fit	A fitted explanatory Rasch model.
item	Item name.
component	Either "location" or "thresholds".

Value

A partially relaxed "rasch_explanatory" fit.

report_document

Write an editable or print-ready analysis report

Description

Renders the active Rasch or paired-comparison fit as a self-contained HTML document, an editable Word document, or a PDF. The report contains the principal estimates, model-specific tables, diagnostic figures, and software provenance. Complete machine-readable results remain available from [save_outputs](#).

Usage

```
report_document(
  fit,
  file,
  format = c("auto", "html", "docx", "pdf"),
  title = "Rasch measurement analysis"
)
```

Arguments

fit	A fitted object from rasch , rasch_mfrm , rasch_efrm , btl , or btl_efrm .
file	Output path ending in .html, .docx, or .pdf.
format	Output format. By default it is inferred from file.
title	Report title.

Details

Word and HTML output require Pandoc, supplied with RStudio and available through **rmarkdown**. PDF output also requires a LaTeX installation such as TinyTeX.

Value

Invisibly, the output path.

Examples

```
## Not run:
fit <- rasch(matrix(rbinom(3000, 1, .5), 300, 10))
report_document(fit, file.path(tempdir(), "analysis.docx"))

## End(Not run)
```

report_html

*Write a self-contained HTML report of a Rasch analysis***Description**

Writes one HTML file containing the summary statistics, diagnostic tables, and test-level plots. Images and styles are embedded in the file.

Usage

```
report_html(fit, file, title = "Rasch measurement analysis", dpi = 150)
```

Arguments

fit	A fitted object from rasch .
file	Path of the HTML file to write.
title	Report title.
dpi	Resolution of the embedded plots.

Value

Invisibly, file.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(300 * 6, 1, plogis(outer(rnorm(300), d, "-"))), 300, 6)
colnames(X) <- paste0("I", 1:6)
out <- file.path(tempdir(), "report.html")
report_html(rasch(X), out)
```

residual_correlations *Residual correlations for local dependence (Yen's Q3)***Description**

The pairwise correlations of the standardised response residuals are Yen's (1984) Q3 statistics. Under unidimensionality and local independence the off-diagonal values sit near $-1/(L-1)$; large positive values flag local dependence between item pairs. Following Christensen, Makransky and Horton (2017), each Q3 is also reported relative to the average off-diagonal value (q3_star). There is no universal adjusted-Q3 critical value: it depends on sample size, test length, category structure, and missingness. The default therefore reports the statistics without a binary flag. A user-supplied flag is an explicitly heuristic screening threshold, not a calibrated significance test.

Usage

```
residual_correlations(fit, flag = NULL)
```

Arguments

- fit A fitted object from [rasch](#).
- flag Optional heuristic excess above the average off-diagonal Q3 at which a pair is flagged. The default NULL withholds binary flags.

Value

A list with the Q3 matrix, the adjusted-Q3 star_matrix (each Q3 less the average off-diagonal value, diagonal empty), the average off-diagonal value, pairs (every item pair with q3, q3_star and a flagged indicator, sorted by q3), and the subset of flagged pairs.

References

Yen, W. M. (1984). Effects of local item dependence on the fit and equating performance of the three-parameter logistic model. *Applied Psychological Measurement*, 8(2), 125-145.

Christensen, K. B., Makransky, G., & Horton, M. (2017). Critical values for Yen’s Q3: identification of local dependence in the Rasch model using residual correlations. *Applied Psychological Measurement*, 41(3), 178-194.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 8)
X <- matrix(rbinom(500 * 8, 1, plogis(outer(rnorm(500), d, "-"))), 500, 8)
colnames(X) <- paste0("I", 1:8)
residual_correlations(rasch(X))$average
```

residual_pca	<i>Principal components of the residual correlations</i>
--------------	--

Description

The first residual component (PC1) carries any second dimension; items with opposing loadings define the split used by the unidimensionality t-test. Loadings for the leading components and the eigenvalue table support inspection beyond the first component.

Usage

```
residual_pca(fit, n_components = 10)
```

Arguments

- fit A fitted object from [rasch](#).
- n_components Number of leading components to return, capped at the number of items.

Value

A list with the residual eigenvalues, their proportions, the first-component loadings (sorted), the `loadings_matrix` for the leading components, the `eigen_table` (component, eigenvalue, proportion, cumulative), and the `first_eigenvalue`.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 8)
X <- matrix(rbinom(500 * 8, 1, plogis(outer(rnorm(500), d, "-"))), 500, 8)
colnames(X) <- paste0("I", 1:8)
residual_pca(rasch(X))$first_eigen
```

resolve_dif

*Resolve differential item functioning by iterative item splitting***Description**

Splits items with uniform DIF one at a time, beginning with the largest estimated effect, and refits after each split. This order addresses the artificial DIF that a large departure can induce in otherwise invariant items (Andrich and Hagquist 2012, 2015). Each split gives the item a separate location and threshold structure in every factor cell. A location split does not model a group-specific discrimination, so items with non-uniform DIF are left for review rather than being made untestable by a split. The procedure stops when no resolvable uniform DIF remains or the remaining unsplit reference set reaches `min_anchors`. Items fixed by external anchors are not split.

Usage

```
resolve_dif(
  fit,
  factors = NULL,
  alpha = 0.05,
  p_adjust = "holm",
  min_anchors = NULL,
  max_splits = NULL
)
```

Arguments

<code>fit</code>	A fitted object from rasch carrying person factors.
<code>factors</code>	Person factors to test, as in dif_anova ; defaults to every nominated factor.
<code>alpha</code>	Significance level for the adjusted probabilities.
<code>p_adjust</code>	Multiplicity adjustment across items each round.
<code>min_anchors</code>	Minimum number of original items to leave unsplit as the internal reference set. The procedure stops before this set becomes smaller; pervasive DIF is not artificial DIF. Default <code>max(3, items / 4)</code> .
<code>max_splits</code>	Hard cap on the number of splits. Default: the number of items.

Value

A list of class "rasch_resolve_dif": the final resolved fit, the splits performed (order, item, factor, partial eta-squared, source item, DIF magnitude in logits), the stopped reason, and the residual dif table for the final fit.

References

Andrich, D., & Hagquist, C. (2012). Real and artificial differential item functioning. *Journal of Educational and Behavioral Statistics*, 37(3), 387-416.

See Also

[split_items](#) for a single split, [drop_items](#) to remove an item instead, and [dif_anova](#) for the test it resolves.

Examples

```
set.seed(1); n <- 600
d <- seq(-2, 2, length.out = 8); g <- rep(c("a", "b"), each = n / 2)
sh <- matrix(0, n, 8); sh[g == "b", 3] <- 1.2      # one strong DIF item
X <- matrix(rbinom(n * 8, 1, plogis(outer(rnorm(n), d, "-") - sh)), n, 8)
colnames(X) <- paste0("I", 1:8)
fit <- rasch(data.frame(X, grp = g), factors = "grp")
resolve_dif(fit)$splits
```

resolve_frames	Resolve items that do not hold across frames
----------------	--

Description

Gives each named item a separate location in every frame in which it was administered, then refits the EFRM.

Usage

```
resolve_frames(fit, items, boot_reps = NULL)
```

Arguments

fit	A fitted object from rasch_efrm .
items	Item names to resolve.
boot_reps	Bootstrap replicates for the refit. The default retains the fitted specification; a number overrides it.

Details

A resolved item continues to contribute to person measurement within each frame but no longer constrains the link between those frames. Its versions are named "item (frame)". The remaining common items and the linked set design must still identify the frame units; otherwise the refit is refused by the model's connectivity and rank checks.

Resolve an item when its within-frame measurement remains defensible but its cross-frame location does not. This refit does not estimate a separate discrimination and therefore does not resolve a discrimination-only flag from [frame_invariance](#). Review or remove such an item instead. Use [drop_items](#) when the item should no longer contribute to measurement.

Value

A refitted object of class "rasch_efrm", carrying a note for each item resolved. The resolved versions appear in the item table as "item (frame)".

See Also

[frame_invariance](#), which identifies the items to resolve; [drop_items](#), which removes an item instead; and [split_items](#), the equivalent for an ordinary fit.

Examples

```
d <- simulate_efrm(n_per_group = 200, items_per_set = 6, n_sets = 2,
                  n_groups = 2, set_unit_ratio = 1.3, seed = 5)
tr <- attr(d, "truth")
fit <- rasch_efrm(d, item_sets = tr$item_sets, groups = "group",
                 id = "id", boot_reps = 0)
fit2 <- resolve_frames(fit, "S1I02", boot_reps = 0)
grep("S1I02", fit2$items$item, value = TRUE)
```

run_app

Launch the rasch point-and-click graphical interface

Description

Opens the Shiny application for fitting models and examining their tables, plots and diagnostics. The R code for each result is available in the app. Analyses can be saved and reopened, or exported as HTML, Word or PDF reports.

Usage

```
run_app(...)
```

Arguments

```
...          Passed to shiny::runApp.
```

Details

The app's interface packages ('shiny', 'bslib', 'DT', 'bsicons', and 'callr' for cancellable EFRM estimation) are suggested rather than required by the package. If any are missing, run_app lists them all and, in an interactive session, offers to install them before launching.

Value

Called for its side effect of launching the app.

Examples

```
if (interactive()) run_app()
```

save_item_plots	<i>Save a plot for every item</i>
-----------------	-----------------------------------

Description

Writes one plot per item – the item characteristic curve, category probability curves, threshold probability curves, or category frequencies – to a single multi-page PDF or a ZIP archive of PNGs, chosen by the extension of file.

Usage

```
save_item_plots(
  fit,
  what = c("icc", "ccc", "tpc", "cfreq"),
  file,
  items = NULL,
  n_groups = fit$n_groups,
  grid = seq(-5, 5, 0.05),
  observed = TRUE,
  width = 8,
  height = 5.5,
  dpi = 300
)
```

Arguments

fit	A fitted object from rasch .
what	Which plot: "icc", "ccc", "tpc", or "cfreq".
file	Output path ending in .pdf (one page per item) or .zip (one PNG per item).
items	Item names or indices; all items by default.
n_groups	Class intervals for observed overlays.
grid	Logit grid for the curves.
observed	Overlay observed proportions on the category and threshold probability curves.
width, height, dpi	Device size in inches and PNG resolution.

Value

Invisibly, the output path.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(400 * 6, 1, plogis(outer(rnorm(400), d, "-"))), 400, 6)
colnames(X) <- paste0("I", 1:6)
f <- rasch(X)
save_item_plots(f, "icc", file.path(tempdir(), "icc_all.pdf"))
```

save_outputs

Save the outputs of a Rasch analysis

Description

Writes the summary, estimates, diagnostic tables, person measures, and model-specific results as CSV. Plots are written as PNG and, optionally, PDF, together with a plain-text analysis summary. For MFRM and EFRM fits, item estimates and response-cell diagnostics are saved separately.

Usage

```
save_outputs(
  fit,
  dir,
  formats = c("png", "pdf"),
  width = 9,
  height = 6,
  dpi = 300,
  item_plots = TRUE
)
```

Arguments

fit	A fitted object from rasch .
dir	Output directory; created if absent.
formats	Plot formats, any of "png" and "pdf".
width, height	Plot size in inches.
dpi	PNG resolution.
item_plots	Also write the per-item plot set (one ICC, category curve, threshold curve, and frequency chart per item).

Value

Invisibly, the vector of files written.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(300 * 6, 1, plogis(outer(rnorm(300), d, "-"))), 300, 6)
colnames(X) <- paste0("I", 1:6)
out <- file.path(tempdir(), "rasch-out")
save_outputs(rasch(X), out, formats = "png", item_plots = FALSE)
```

save_person_plots	<i>Save a kidmap for every person</i>
-------------------	---------------------------------------

Description

Writes one kidmap ([plot_kidmap](#)) per person to a single multi-page PDF or a ZIP archive of PNGs, chosen by the extension of file. Persons without a location estimate are skipped.

Usage

```
save_person_plots(
  fit,
  file,
  persons = NULL,
  level = 0.95,
  width = 8,
  height = 6,
  dpi = 300
)
```

Arguments

fit	A fitted object from rasch .
file	Output path ending in .pdf (one page per person) or .zip (one PNG per person).
persons	Row numbers or IDs; all estimated persons by default.
level	Confidence level of the band marking unexpected responses.
width, height, dpi	Device size in inches and PNG resolution.

Value

Invisibly, the output path.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(60 * 6, 1, plogis(outer(rnorm(60), d, "-"))), 60, 6)
colnames(X) <- paste0("I", 1:6)
f <- rasch(X)
save_person_plots(f, file.path(tempdir(), "kidmaps.pdf"), persons = 1:5)
```

score_table

Raw score to measure conversion table

Description

The score-to-logit conversion for complete responders: every possible raw score with its location, standard error, and the frequency and cumulative percentage of complete responders at that score (the complete-data estimates table of Andrich and Marais 2019, ch. 10).

Usage

```
score_table(
  fit,
  method = c("wle", "mle"),
  extremes = c("model", "extrapolated")
)
```

Arguments

fit	A fitted object from rasch .
method	"wle" (Warm, default) or "mle".
extremes	Treatment of the extreme scores. "model" keeps the estimator's own values; these are NA for MLE. "extrapolated" applies the geometric extrapolation.

Details

Two estimators are available. "wle" (the default) is Warm's weighted likelihood estimate, finite at the extreme scores. "mle" is the plain maximum likelihood estimate, infinite at the extremes. extremes = "extrapolated" replaces the extreme-score entries by the geometric extrapolation described in Andrich and Marais (2019, ch. 10): successive score-to-score differences grow towards the extremes, so the last difference is continued geometrically – the extrapolated top difference d solves $b = \sqrt{ad}$ where a, b are the two preceding differences (equivalently $d = b^2/a$), and symmetrically at zero. The standard error at an extrapolated location is $1/\sqrt{I(\theta)}$ evaluated there. With method = "wle" the extrapolation replaces the finite Warm estimates at the extremes, giving the extrapolated form of the conversion table from a WLE analysis.

Value

A data frame with score, theta, se, freq, cum_pct (omitted when no complete responders exist), and extrapolated; NULL when the fitted items do not share one discrimination or an item is represented by several MFRM or EFRM response cells.

References

Andrich, D. and Marais, I. (2019). A Course in Rasch Measurement Theory: Measuring in the Educational, Social and Health Sciences. Springer.

Warm, T. A. (1989). Weighted likelihood estimation of ability in item response theory. *Psychometrika*, 54(3), 427–450.

Examples

```
set.seed(1)
d <- seq(-1.5, 1.5, length.out = 6)
X <- matrix(rbinom(300 * 6, 1, plogis(outer(rnorm(300), d, "-"))), 300, 6)
colnames(X) <- paste0("I", 1:6)
score_table(rasch(X), method = "mle", extremes = "extrapolated")
```

simulate_btl

Simulate paired-comparison data

Description

Generates dichotomous or ordered paired comparisons from the Bradley–Terry–Luce model. Optional arguments introduce a second object attribute, erratic judges, or within-judge dependence. Generating values are stored in `attr(x, "truth")`.

Usage

```
simulate_btl(
  n_objects = 8,
  n_judges = 12,
  reps_per_pair = 25,
  model = c("dichotomous", "polytomous", "graded"),
  n_categories = 4,
  object_sd = 1,
  second_attribute = NULL,
  erratic_judges = 0,
  dependence = NULL,
  seed = NULL
)
```

Arguments

n_objects, n_judges	Objects to scale and judges comparing them.
reps_per_pair	Comparisons made of each object pair.
model	"dichotomous" (a winner) or "polytomous" (a rated margin in n_categories categories; an earlier development-era value "graded" is accepted as an alias).
n_categories	Categories for the polytomous model.
object_sd	Spread of the object locations (evenly spaced, sum-zero).
second_attribute	NULL, or list(rho=): half the judges rank by a second object attribute correlated rho with the first. This introduces residual dimensionality and possible intransitivity.
erratic_judges	Proportion of judges who choose at random.
dependence	NULL, or list(exposure=, carry_over=): within-judge order effects (a seen-before advantage and a pull from the judge's own earlier verdicts). Adds an order column. Feeds the dependence effects fitted by btl .
seed	Optional RNG seed.

Value

A data frame of class "rasch_sim": object_a, object_b, winner (or response when polytomous), judge, and order when dependence is planted; with attr(x, "truth").

Examples

```
d <- simulate_btl(8, 12, erratic_judges = 0.15, seed = 1)
bt <- btl(d, "object_a", "object_b", winner = "winner", judge = "judge")
bt$judges          # the erratic judges carry large fit residuals
```

simulate_btl_efrm	<i>Simulate paired-comparison EFRM data with differing frame units</i>
-------------------	--

Description

Generates dichotomous paired comparisons whose latent unit differs across judge-panel by object-set frames – the paired-comparison extension of the extended frame of reference model (Humphry 2005) fitted by [btl_efrm](#). Objects in set s have a within-set calibration location β_s ; their common-scale value is $v = \alpha_s \beta_s + \kappa_s$. A comparison judged in panel g carries the panel unit ϕ_{ig} : within a set the comparison logit is $\phi_{ig} (\beta_{sa} - \beta_{sb})$, across sets it is $\phi_{ig} (v_a - v_b)$. The planted panel units, set units and origins are recovered by [btl_efrm](#).

Usage

```
simulate_btl_efrm(
  n_objects_per_set = 8,
  n_sets = 2,
  n_judges_per_panel = 6,
  n_panels = 2,
  reps_within = 20,
  reps_cross = 20,
  panel_units = NULL,
  set_units = NULL,
  set_origins = NULL,
  object_sd = 1,
  seed = NULL
)
```

Arguments

<code>n_objects_per_set</code> , <code>n_sets</code>	Objects in each set and number of sets.
<code>n_judges_per_panel</code> , <code>n_panels</code>	Judges in each panel and number of panels.
<code>reps_within</code>	Replications of each within-set object pair.
<code>reps_cross</code>	Replications of each cross-set object pair.
<code>panel_units</code>	Panel units ϕ (length <code>n_panels</code>); the default is all one, and any supplied vector is rescaled to geometric mean one.
<code>set_units</code>	Set units α (length <code>n_sets</code>); the default is all one, and <code>alpha_1</code> is forced to one (the reference set).
<code>set_origins</code>	Set origins κ (length <code>n_sets</code>); the default is all zero, and <code>kappa_1</code> is forced to zero.
<code>object_sd</code>	Spread of the within-set calibration locations.
<code>seed</code>	Optional RNG seed.

Value

A data frame of class "rasch_sim" with columns `object_a`, `object_b`, `winner`, `judge` and `panel`, and `attr(x, "truth")` holding the common-scale values v , the per-set β , the units ϕ , α , κ , and the `object_sets` map to pass to [btl_efrm](#).

Examples

```
d <- simulate_btl_efrm(6, 2, set_units = c(1, 1.4), seed = 1)
bt <- btl_efrm(d, "object_a", "object_b", winner = "winner",
  judge = "judge", panels = "panel",
  object_sets = attr(d, "truth")$object_sets,
  se_method = "conditional")
bt$alpha_table # recovers the ~1.4 set unit
```

simulate_efrm

*Simulate extended frame-of-reference data with differing units***Description**

Generates data whose latent unit differs across item-set by person-group frames (Humphry 2005): a person in group g responding to an item in set s does so at the frame unit $\rho = \alpha_{\text{set}} * \phi_{\text{group}}$ scaling the whole exponent. The planted set- and group-unit ratios are recovered by [rasch_efrm](#).

Usage

```
simulate_efrm(
  n_per_group = 300,
  items_per_set = 8,
  n_sets = 2,
  n_groups = 2,
  set_unit_ratio = 1.3,
  group_unit_ratio = 1,
  n_categories = 2,
  theta_sd = 1.3,
  seed = NULL
)
```

Arguments

<code>n_per_group</code>	Persons in each group.
<code>items_per_set</code>	Items in each set.
<code>n_sets, n_groups</code>	Numbers of item sets and person groups.
<code>set_unit_ratio, group_unit_ratio</code>	Geometric span of the set and group units across their levels (1 = equal units, i.e. an ordinary Rasch fit).
<code>n_categories</code>	Response categories per item: 2 (the default) gives dichotomous items; larger values give partial credit items whose evenly spaced thresholds are centred on the item locations, with the frame unit scaling the whole exponent as in the dichotomous case.
<code>theta_sd</code>	Spread of person ability.
<code>seed</code>	Optional RNG seed.

Value

A wide data frame of class "rasch_sim", containing an ID, item columns, and group. Its truth attribute contains the item-set map required by [rasch_efrm](#).

Examples

```
d <- simulate_efrm(200, 6, set_unit_ratio = 1.3, seed = 1)
tr <- attr(d, "truth")
ef <- rasch_efrm(d, item_sets = tr$item_sets, groups = "group",
                boot_reps = 0)    # point estimates only
ef$alpha_table  # planted ratio 1.3, recovered within small-sample noise
```

simulate_mfrm

Simulate many-facet Rasch data

Description

Generates fully crossed ratings from a many-facet Rasch model (Linacre 1989), with optional erratic raters, item-by-rater interaction, or halo.

Usage

```
simulate_mfrm(
  n_persons = 80,
  n_items = 5,
  n_raters = 6,
  n_categories = 4,
  theta_sd = 1.2,
  item_sd = 1,
  rater_severity_sd = 0.6,
  erratic_raters = 0,
  interaction = NULL,
  halo = 0,
  seed = NULL
)
```

Arguments

n_persons, n_items, n_raters	Facet sizes (fully crossed).
n_categories	Rating categories.
theta_sd, item_sd	Spread of person ability and item difficulty.
rater_severity_sd	Spread of rater severities (the core facet; recovered in facet_effects).
erratic_raters	Proportion of raters who rate at random (feeds the rater fit residual).
interaction	NULL, or list(rater=, item=, bias=): one rater is unusually harsh (positive) or lenient (negative) on one item. Feeds the item-by-rater interaction (fit with interaction =).

halo	Proportion of raters showing a halo effect: they rate by the person's overall level and barely differentiate items (feeds the rater fit residual and the item-by-rater interaction).
seed	Optional RNG seed.

Value

A long data frame of class "rasch_sim" (person, item, rater, score) ready for `rasch_mfrm`, with the truth attached.

Examples

```
d <- simulate_mfrm(60, 5, 6, rater_severity_sd = 0.8, seed = 1)
mf <- rasch_mfrm(d, person = "person", item = "item", score = "score",
  facets = "rater")
cor(mf$facet_effects$rater$severity, attr(d, "truth")$severity) # recovered
```

simulate_rasch	<i>Simulate person-by-item Rasch data</i>
----------------	---

Description

Generates dichotomous, partial credit, or rating scale data. Optional arguments introduce item misfit, guessing, multidimensionality, local dependence, DIF, response styles, or missingness. Generating values are stored in `attr(x, "truth")`.

Usage

```
simulate_rasch(
  n_persons = 500,
  n_items = 20,
  model = c("dichotomous", "PCM", "RSM"),
  n_categories = 3,
  theta_mean = 0,
  theta_sd = 1,
  theta_dist = "normal",
  difficulty = c(-2.5, 2.5),
  threshold_spread = 1.2,
  discrimination = 1,
  guessing = 0,
  second_dim = NULL,
  dependence = NULL,
  dif = NULL,
  careless = 0,
  response_style = NULL,
  speeded = 0,
  disordered = NULL,
```

```

    n_groups = 1,
    missing = 0,
    seed = NULL
)

```

Arguments

<code>n_persons, n_items</code>	Sample size and test length.
<code>model</code>	"dichotomous", "PCM", or "RSM". Under "RSM" every item shares one category-threshold pattern (items differ by location only); under "PCM" each item's threshold spacings and span are drawn afresh, as the partial credit model allows.
<code>n_categories</code>	Response categories for polytomous models (≥ 3).
<code>theta_mean, theta_sd</code>	Mean and standard deviation of the person distribution.
<code>theta_dist</code>	Shape of the person distribution: "normal", "uniform", "skew", or "bimodal".
<code>difficulty</code>	Either the two endpoints of an evenly spaced location range, or one location per item.
<code>threshold_spread</code>	Half-range of the category thresholds about each item location (polytomous).
<code>discrimination</code>	The item slope, supplied as one value or one per item. Values above 1 produce steeper responses and negative fit residuals. Values below 1 produce flatter responses and positive fit residuals.
<code>guessing</code>	Scalar or length- <code>n_items</code> lower asymptote (dichotomous): low-location persons answer correctly by chance.
<code>second_dim</code>	NULL, or <code>list(items=, rho=)</code> : the named items load on a second trait correlated ρ with the first.
<code>dependence</code>	NULL, or <code>list(pairs=, strength=)</code> : each pair's second item responds partly to the first. This departure feeds the residual-dependence diagnostics.
<code>dif</code>	NULL, or <code>list(items=, uniform=, nonuniform=)</code> : the named items function differently for the last person group: a location shift (uniform) and/or a slope change (nonuniform). Needs <code>n_groups</code> ≥ 2 .
<code>careless</code>	Proportion of persons who answer at random.
<code>response_style</code>	NULL, or <code>list(type=, prop=, strength=)</code> with type "extreme" or "middle": a proportion <code>prop</code> of persons favour the end (or middle) categories regardless of the trait, with distortion strength (default 1.6) on the log-probability scale (polytomous).
<code>speeded</code>	Proportion not reached at the last item: a growing tail of missing responses over the final items.
<code>disordered</code>	NULL or item names/indices given disordered thresholds (polytomous; feeds the threshold diagnostics).
<code>n_groups</code>	Number of equal person groups (a group factor column is added when > 1 , for DIF).
<code>missing</code>	Proportion of responses set missing (completely at random).
<code>seed</code>	Optional RNG seed.

Value

A data frame of class "rasch_sim" (item columns I01..., an id column, and a group column when grouped), with attr(x, "truth") holding the generating parameters and the planted departures.

Examples

```
# a clean scale with one over-discriminating item and one DIF item
d <- simulate_rasch(400, 12, discrimination = c(3, rep(1, 11)),
  dif = list(items = "I06", uniform = 1), n_groups = 2,
  seed = 1)
fit <- rasch(d, id = "id", factors = "group")
fit$items[c("item", "infit_ms", "outfit_ms")] # item 1 misfits
dif_anova(fit)$summary                       # item 6 flags
```

sim_apply

*Apply a statistic across a simulation batch***Description**

Applies FUN to each replicate of a [sim_replicate](#) batch, catching replicates on which FUN errors – for example a small or disconnected draw the estimator refuses as unidentified – so a single failure does not abort the whole Monte-Carlo run. Failed replicates contribute NA; the number of failures and the distinct error messages are attached as attributes.

Usage

```
sim_apply(batch, FUN, ...)
```

Arguments

batch	A "rasch_sim_batch" from sim_replicate (or any list of datasets).
FUN	A function of one dataset returning a scalar statistic.
...	Further arguments passed to FUN.

Value

A vector of per-replicate statistics, with NA where the function failed. Attribute n_failed gives the failure count; failure_messages contains the distinct messages.

Examples

```
batch <- sim_replicate(simulate_rasch, 10, n_persons = 300, n_items = 8,
  seed = 1)
psi <- sim_apply(batch, function(d) rasch(d)$psi$PSI)
mean(psi, na.rm = TRUE)
```

sim_recovery	<i>Compare fitted and generating parameters</i>
--------------	---

Description

Compares fitted parameters with the generating values from a `simulate_*` function (carried on the data as `attr(sim, "truth")`): item difficulties and person abilities for a Rasch fit, object locations for a paired-comparison fit, rater severities (with item and person measures) for a many-facet fit, and the set units for a frames fit. Locations are mean-centred before comparison, since the model identifies them only up to an origin.

Usage

```
sim_recovery(fit, sim)
```

Arguments

<code>fit</code>	A fit of the simulated data (rasch , btl , rasch_mfrm , or rasch_efrm).
<code>sim</code>	The simulated data (from a <code>simulate_*</code> function).

Value

A list of class "rasch_recovery": summary (per parameter type: n, correlation, RMSE, bias) and pieces (the true and estimated values behind each).

Examples

```
d <- simulate_rasch(500, 12, seed = 1)
sim_recovery(rasch(d, id = "id"), d)$summary
```

sim_replicate	<i>Replicate a simulation for Monte Carlo studies</i>
---------------	---

Description

Calls one of the `simulate_*` functions `n` times with successive seeds, returning the datasets as a list – for power, Type-I, or parameter-recovery studies.

Usage

```
sim_replicate(FUN, n, ..., seed = NULL)
```

Arguments

<code>FUN</code>	A simulator, e.g. simulate_rasch .
<code>n</code>	Number of datasets.
<code>...</code>	Arguments passed to <code>FUN</code> (the same each replicate).
<code>seed</code>	Seed of the first replicate (each subsequent one increments it).

Value

A list of class "rasch_sim_batch", one simulated dataset per element.

Examples

```
# 8 datasets with a planted DIF item; how often is it flagged?
batch <- sim_replicate(simulate_rasch, 8, n_persons = 400, n_items = 10,
                      dif = list(items = "I05", uniform = 0.8), n_groups = 2,
                      seed = 1)
# sim_apply() is resilient: a replicate the estimator refuses (e.g. a
# small or disconnected draw) contributes NA instead of aborting the run
flagged <- sim_apply(batch, function(d)
  dif_anova(rasch(d, id = "id", factors = "group"))$summary$uniform_DIF[5])
mean(flagged, na.rm = TRUE)
```

split_items

Split items by a person factor to resolve DIF

Description

Replaces each nominated item with one item per level of a person factor, each carrying that level's responses only (other levels missing). Every group then receives its own item location, which resolves the invariance violation flagged by [dif_anova](#); the distance between the split locations estimates the DIF size. The model is refitted with the same settings.

Usage

```
split_items(fit, items, by)
```

Arguments

<code>fit</code>	A fitted object from rasch .
<code>items</code>	Character vector naming the item(s) to split.
<code>by</code>	The name of a person factor nominated in the fit, or a grouping vector with one entry per person.

Value

A new [rasch](#) fit in which each split item appears as "item (level)", with the splits recorded in its notes. Person and item estimates are recalculated with the fitted grouping, keyed scoring, anchors on unchanged items, PCM constraints and optimisation controls. An anchored item cannot be split.

See Also

[resolve_dif](#), which applies this iteratively; [drop_items](#), which removes an item rather than resolving it; and [dif_anova](#), which identifies the items to split.

Examples

```
set.seed(1); n <- 600
d <- seq(-2, 2, length.out = 8); g <- rep(c("a", "b"), each = n / 2)
sh <- matrix(0, n, 8); sh[g == "b", 3] <- 1
X <- matrix(rbinom(n * 8, 1, plogis(outer(rnorm(n), d, "-") - sh)), n, 8)
colnames(X) <- paste0("I", 1:8)
fit <- rasch(data.frame(X, grp = g), factors = "grp")
fit2 <- split_items(fit, "I3", by = "grp")
fit2$items$item
```

spread_test

*Spread-parameter test for dependence within subtests***Description**

Andrich's (1985) least-upper-bound screen: the spread component λ of a polytomous item (half the distance between successive thresholds in the principal-components parameterisation, estimated here by [pcml_pc](#)) cannot fall below the value implied by the binomial distribution when the item is a subtest of equally difficult, independent dichotomous items; different difficulties only raise it. Sampling uncertainty matters when the fitted spread lies near the bound. The function therefore reports whether the point estimate is below the bound separately from a one-sided test of $H_0 : \lambda \geq \lambda_0$ against $H_1 : \lambda < \lambda_0$. Evidence of dependence requires the adjusted one-sided probability to be below alpha; a point estimate below the bound alone is not treated as a verdict. Applied to the superitems recorded by [combine_items](#). The binomial bound applies only when every component was dichotomous; a composite containing a polytomous item is shown but its bound and verdict are withheld. The input calibration and the principal-components refit must both converge.

Usage

```
spread_test(fit, maxit = 60, tol = 1e-08, alpha = 0.05, p_adjust = "holm")
```

Arguments

fit	A fitted object from rasch .
maxit, tol	Passed to the pcml_pc refit.
alpha	Significance level for the one-sided dependence screen.
p_adjust	Multiplicity adjustment across the eligible superitems; one of <code>stats::p.adjust.methods</code> .

Value

A data frame with one row per recorded superitem: `item`, `m`, whether the binomial bound is eligible, the spread estimate and its `se`, the bound `lub` (available for dichotomous-component subtests with maximum scores 2 to 8), $z = (\text{spread} - \text{lub})/\text{se}$, the one-sided `p` and adjusted `p_adj`, `below_bound` for the point-estimate comparison, and dependent for adjusted evidence at `alpha`. Items not formed by `combine_items()` are omitted. The result retains `alpha` and `p_adjust` as attributes.

References

Andrich, D. (1985). An elaboration of Guttman scaling with Rasch models for measurement. In N. B. Tuma (Ed.), *Sociological Methodology 1985* (pp. 33–80). Jossey-Bass.

Andrich, D. and Marais, I. (2019). *A Course in Rasch Measurement Theory: Measuring in the Educational, Social and Health Sciences*. Springer.

Examples

```
set.seed(1); N <- 600
d0 <- seq(-1.5, 1.5, length.out = 8)
X <- matrix(rbinom(N * 8, 1, plogis(outer(rnorm(N), d0, "-"))), N, 8)
X[, 5] <- ifelse(runif(N) < .85, X[, 4], 1 - X[, 4])
X[, 6] <- ifelse(runif(N) < .75, X[, 4], 1 - X[, 4]) # a dependent triple
colnames(X) <- paste0("I", 1:8)
fit2 <- combine_items(rasch(X), list(c("I4", "I5", "I6"), c("I1", "I2", "I3")))
spread_test(fit2)
```

tailored_analysis	<i>Tailored analysis for guessing</i>
-------------------	---------------------------------------

Description

Runs the four-step tailored procedure of Andrich, Marais and Humphry (2012) on a dichotomous analysis. Step 1 is the supplied fit. Step 2 (tailored) sets to missing every observed response whose modelled probability of success, at the step-1 person and item estimates, is below chance, and re-estimates items and persons. Step 3 (origin-equated) re-analyses the *original* data with the mean location of the anchor items fixed at their tailored values by average anchoring, so the two calibrations share an origin. Step 4 (all-anchored) fixes every item at its tailored difficulty and re-estimates persons on the original data. Guessing is indicated when difficult items are estimated harder in the tailored analysis than in the origin-equated one; the comparison table and [plot_equate](#) on the two calibrations show it directly.

Usage

```
tailored_analysis(
  fit,
  chance = 0.25,
  anchor_items = NULL,
```

```

    se_method = c("none", "bootstrap"),
    boot_reps = 999L
  )

```

Arguments

<code>fit</code>	An unanchored, unconstrained dichotomous fit from rasch . The procedure estimates its own common origin.
<code>chance</code>	The guessing floor: the probability of success by chance (1/number of options; default 0.25).
<code>anchor_items</code>	Items whose mean location fixes the common origin in step 3. The default takes the third of the test (at least two items) least affected by tailoring – fewest responses removed, ties broken towards the easier tailored location – which are the easy items the procedure trusts.
<code>se_method</code>	"none" (default) reports the item shifts descriptively. "bootstrap" resamples persons and repeats the complete four-step procedure, including automatic anchor selection, to obtain standard errors, percentile intervals, and Holm-adjusted tests. When a person identifier occurs on several rows, all of that person's rows are resampled together.
<code>boot_reps</code>	Person-bootstrap replicates when <code>se_method = "bootstrap"</code> ; at least 50, default 999. The sign-count bootstrap p-value has resolution floor $2/(boot_reps + 1)$, so after the Holm adjustment across m items the smallest achievable adjusted p is $2m/(boot_reps + 1)$; a warning fires when that floor exceeds 0.05 (detection would be impossible).

Value

A list of class "rasch_tailored": tailored, origin-equated, and anchored fits, the comparison table (initial, tailored, origin-equated locations, the tailored-minus-equated shift; bootstrap uncertainty columns when requested), the number of responses removed, the anchor items used, `se_method`, and the number of usable bootstrap replicates `boot_reps_used`.

References

Waller, M. I. (1989). Modeling guessing behavior: A comparison of two IRT models. *Applied Psychological Measurement*, 13, 233-243. Andrich, D., Marais, I. and Humphry, S. (2012). Using a theorem by Andersen and the dichotomous Rasch model to assess the presence of random guessing in multiple choice items. *Journal of Educational and Behavioral Statistics*, 37, 417-442.

Examples

```

set.seed(1); N <- 800
d <- seq(-2, 2.5, length.out = 10); th <- rnorm(N)
P <- plogis(outer(th, d, "-"))
P <- 0.25 + 0.75 * P           # uniform guessing floor
X <- matrix(rbinom(N * 10, 1, P), N, 10)
colnames(X) <- paste0("I", 1:10)
ta <- tailored_analysis(rasch(X), chance = 0.25)
ta$table

```

targeting_table	<i>Targeting and reliability summary as a table</i>
-----------------	---

Description

The person and calibration location moments, threshold range and coverage, and the applicable separation and reliability indices as a two-column table suitable for saving and reporting. MFRM and EFRM fits describe item-by-facet or item-by-frame response cells rather than additional items. Coefficient alpha is not applicable when an item has several response cells; it is retained for the one-cell-per-item reduction.

Usage

```
targeting_table(fit)
```

Arguments

`fit` A fitted object from [rasch](#).

Value

A data frame with columns `statistic` and `value`.

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(300 * 6, 1, plogis(outer(rnorm(300), d, "-"))), 300, 6)
colnames(X) <- paste0("I", 1:6)
targeting_table(rasch(X))
```

test_information	<i>Test information function</i>
------------------	----------------------------------

Description

Fisher information over a grid of person locations, with the corresponding standard error of measurement. Ordinary Rasch fits return one whole-test curve. EFRM fits return one curve per person group and per item-set administration pattern actually observed within that group (in a linking design, persons who took only the core set get a core-only curve, and the linking subsample gets the pooled one). MFRM fits return one curve per set of facet conditions observed together for a person, so ratings that jointly inform the same person measure are added and mutually exclusive designs remain separate.

Usage

```
test_information(fit, grid = seq(-6, 6, by = 0.1))
```

Arguments

- fit A fitted object from [rasch](#).
- grid Logit grid over which to evaluate the information.

Details

For an administrable block $\mathcal{A}$, the information and standard error of measurement are

$$I(\theta) = \sum_{i \in \mathcal{A}} d_i^2 \text{Var}(X_i \mid \theta), \quad \text{SEM}(\theta) = I(\theta)^{-1/2},$$

where d_i is the frame unit or discrimination multiplier. For an ordinary Rasch fit, $d_i = 1$.

Value

A data frame with `theta`, `info`, and `sem`. For EFRM and MFRM fits it also contains a design column identifying the administrable frame or facet design.

References

Andrich, D. and Marais, I. (2019). A Course in Rasch Measurement Theory: Measuring in the Educational, Social and Health Sciences. Springer.

See Also

[targeting_table](#) and [plot_tif](#).

Examples

```
set.seed(1)
d <- seq(-1.5, 1.5, length.out = 6)
X <- matrix(rbinom(300 * 6, 1, plogis(outer(rnorm(300), d, "-"))), 300, 6)
colnames(X) <- paste0("I", 1:6)
head(test_information(rasch(X)))
```

threshold_index	<i>Enumerate item-category thresholds</i>
-----------------	---

Description

Builds the index mapping each item-category threshold to a global id, given the maximum score of each item.

Usage

```
threshold_index(m)
```

Arguments

`m` Integer vector of maximum scores per item (1 for dichotomous items).

Value

A data frame with columns `id`, `item`, and `k` (the within-item threshold number).

Examples

```
threshold_index(c(1, 3, 2))
```

```
wright_map
```

Draw a Wright map with WrightMap

Description

Prepares person estimates and item locations from a fitted model and passes them to `WrightMap::wrightMap()`. Person panels may be formed from variables retained in the fit or supplied as a matrix. Item panels use the `item.groups` facility in `WrightMap 1.5`.

Usage

```
wright_map(
  fit,
  type = c("thresholds", "locations"),
  person_panels = NULL,
  item_panels = NULL,
  ...
)
```

Arguments

<code>fit</code>	A fitted object from rasch , including explanatory, MFRM and EFRM fits. Comparative-judgement models do not estimate person locations and are not supported.
<code>type</code>	Plot category "thresholds" or one "locations" estimate per item.
<code>person_panels</code>	Optional person-panel specification. Supply the name of one or more variables retained in <code>fit\$person</code> , a vector or factor of panel memberships with one value per person, or a numeric matrix or data frame whose columns contain the estimates for the panels. Missing estimates are permitted. For EFRM fits, "groups" uses the fitted person groups.
<code>item_panels</code>	Optional vector or factor assigning each item row to a panel. A named vector is matched to item names; an unnamed vector is used in item order. A named list may instead map panel names to item names. For EFRM fits, use "sets", "groups", or <code>c("sets", "groups")</code> to arrange the calibrated response columns by item set, person group, or frame. This option requires <code>WrightMap 1.5</code> or later.
<code>...</code>	Further arguments passed to <code>WrightMap::wrightMap()</code> , such as <code>person.side</code> , <code>item.side</code> , <code>main.title</code> , or graphical settings.

Details

For partial credit and rating scale models, `type = "thresholds"` displays each item's estimated category thresholds, labelled `t1`, `t2`, and so on. A wholly dichotomous scale omits the redundant `t1` labels; in a mixed scale, dichotomous items retain `t1` to align them with the polytomous items. `type = "locations"` displays one location per item. In MFRM and EFRM fits, the rows are the calibrated item-by-facet or item-by-frame response columns. For EFRM fits, `person_panels = "groups"` and `item_panels = "sets"` use the fitted frame design; both may be specified together. A single person panel has no heading by default. When several person panels are requested, their panel labels are printed above the distributions. The plot has no overall title unless one is supplied through `main.title`.

Value

Invisibly, the threshold matrix returned by `WrightMap::wrightMap()` when its `return.thresholds` argument is `true`; otherwise `NULL`.

References

Torres Irribarra, D., and Freund, R. (2025). WrightMap: IRT item-person map with ConQuest integration. R package version 1.5.

See Also

[plot_wright](#)

Examples

```
set.seed(1)
d <- seq(-2, 2, length.out = 6)
X <- matrix(rbinom(300 * 6, 1, plogis(outer(rnorm(300), d, "-"))), 300, 6)
colnames(X) <- paste0("I", 1:6)
fit <- rasch(X)
if (requireNamespace("WrightMap", quietly = TRUE)) {
  wright_map(fit)
}
```

Index

btl, [4](#), [8](#), [9](#), [12–14](#), [17](#), [19](#), [20](#), [22](#), [44](#), [48](#), [49](#),
[55–60](#), [62](#), [86](#), [96](#), [107](#), [114](#)
btl_dif, [7](#), [7](#)
btl_dimensionality, [9](#)
btl_efrm, [7](#), [10](#), [63](#), [96](#), [107](#), [108](#)
btl_equate, [13](#), [58](#)
btl_explanatory, [5](#), [15](#), [95](#)
btl_information, [7](#), [17](#), [19](#), [62](#)
btl_next_pairs, [18](#), [18](#), [62](#)
btl_transitivity, [7](#), [20](#)

chisq_detail, [21](#)
combine_items, [21](#), [41](#), [116](#)
compare_fits, [22](#)
ctt_table, [24](#)

dependence_magnitude, [25](#)
dif_anova, [26](#), [31](#), [33](#), [35](#), [86](#), [94](#), [99](#), [100](#),
[115](#), [116](#)
dif_contrasts, [28](#), [29](#), [32–35](#)
dif_posthoc, [28](#), [31](#)
dif_size, [27](#), [28](#), [31](#), [33](#), [33](#)
dimensionality_magnitude, [35](#)
dimensionality_test, [37](#)
distractor_analysis, [38](#)
distractor_rescore, [39](#)
drop_items, [22](#), [40](#), [46](#), [89](#), [90](#), [100](#), [101](#), [116](#)

equate_tests, [41](#), [66](#)
explanatory_diagnostics, [17](#), [42](#), [92](#)
explanatory_test, [17](#), [43](#), [91](#), [92](#)

fit_summary_table, [44](#)
frame_invariance, [28](#), [41](#), [44](#), [89](#), [90](#), [101](#)

guttman_table, [46](#)

item_moments, [47](#)

judge_pair_surprise, [48](#), [60](#)
judge_surprise, [48](#), [49](#)

lr_test, [23](#), [50](#)

pcml, [51](#), [52](#), [85](#)
pcml_pc, [52](#), [85](#), [116](#)
person_extrapolated, [53](#), [55](#)
person_wle, [54](#)
plot_btl, [55](#)
plot_btl_categories, [56](#)
plot_btl_dependence, [56](#)
plot_btl_dim_map, [57](#)
plot_btl_equate, [58](#)
plot_btl_icc, [59](#)
plot_btl_judge_map, [60](#)
plot_btl_scee, [61](#)
plot_btl_targeting, [18](#), [19](#), [61](#)
plot_btl_transitivity, [62](#)
plot_btl_units, [13](#), [63](#)
plot_catfreq, [63](#)
plot_ccc, [64](#)
plot_distractors, [39](#), [65](#)
plot_equate, [58](#), [66](#), [117](#)
plot_facets, [66](#)
plot_frames, [67](#)
plot_guttman, [68](#)
plot_icc, [59](#), [69](#)
plot_icc_frames, [70](#)
plot_item_map, [71](#)
plot_kidmap, [71](#), [104](#)
plot_pca, [72](#)
plot_pca_biplot, [73](#)
plot_pcc, [74](#)
plot_person_fit, [74](#)
plot_pimap, [75](#)
plot_recovery, [76](#)
plot_resid_cor, [76](#)
plot_resid_dist, [77](#)
plot_scee, [78](#)
plot_tcc, [79](#)
plot_threshold_map, [80](#)
plot_threshold_prob, [80](#)

plot_tif, [81](#), [120](#)
plot_wright, [82](#), [122](#)

rack_data, [83](#)
rasch, [21](#), [22](#), [24](#), [25](#), [27](#), [30](#), [32](#), [34](#), [36–41](#),
 [44](#), [47](#), [50](#), [54](#), [63–66](#), [68](#), [69](#), [71–75](#),
 [77–82](#), [84](#), [87](#), [88](#), [90](#), [91](#), [94](#), [96–99](#),
 [102–105](#), [114–116](#), [118–121](#)
rasch_efrm, [13](#), [22](#), [27](#), [40](#), [45](#), [46](#), [67](#), [70](#), [86](#),
 [87](#), [94](#), [96](#), [100](#), [109](#), [114](#)
rasch_explanatory, [90](#)
rasch_mfrm, [22](#), [27](#), [30](#), [32](#), [34](#), [40](#), [67](#), [86](#), [90](#),
 [92](#), [96](#), [111](#), [114](#)
relax_btl_explanatory, [17](#), [94](#)
relax_explanatory, [92](#), [95](#)
report_document, [96](#)
report_html, [97](#)
residual_correlations, [21](#), [22](#), [97](#)
residual_pca, [98](#)
resolve_dif, [28](#), [41](#), [99](#), [116](#)
resolve_frames, [41](#), [46](#), [89](#), [100](#)
run_app, [86](#), [101](#)

save_item_plots, [102](#)
save_outputs, [96](#), [103](#)
save_person_plots, [104](#)
score_table, [53](#), [55](#), [105](#)
sim_apply, [113](#)
sim_recovery, [114](#)
sim_replicate, [113](#), [114](#)
simulate_btl, [7](#), [106](#)
simulate_btl_efrm, [13](#), [107](#)
simulate_efrm, [90](#), [109](#)
simulate_mfrm, [94](#), [110](#)
simulate_rasch, [111](#), [115](#)
split_items, [41](#), [100](#), [101](#), [115](#)
spread_test, [116](#)
stack_data (rack_data), [83](#)

tailored_analysis, [117](#)
targeting_table, [119](#), [120](#)
test_information, [86](#), [90](#), [119](#)
threshold_index, [120](#)

wright_map, [121](#)