

Package: birt (via r-universe)

October 31, 2026

Type Package

Title BIvariate Response Time Modeling with 'lavaan'-Style Syntax

Version 0.2.0

Description Fits item response theory models with one or two latent variables, for example ability and speed in joint models of response accuracy and response time (van der Linden, 2007, <[doi:10.1007/s11336-006-1478-z](https://doi.org/10.1007/s11336-006-1478-z)>), from 'lavaan'-style model syntax. The main engine fits the syntax in 'RTMB' by approximate Bayesian inference as an extended latent Gaussian model (Stringer, Brown and Stafford, 2023, <[doi:10.1080/10618600.2022.2099403](https://doi.org/10.1080/10618600.2022.2099403)>): the persons are integrated out by adaptive Gauss-Hermite quadrature and the remaining parameters by an outer adaptive quadrature, with marginal likelihoods for model comparison; the same engine gives maximum likelihood, with casewise scores for score-based invariance tests and DIF trees. Ordinal and continuous indicators, moderated nonlinear factor analysis statements, sinh-arcsinh residuals (Jones and Pewsey, 2009, <[doi:10.1093/biomet/asp053](https://doi.org/10.1093/biomet/asp053)>) whose quantile effects at every level come from one fit, conditional reliability and the information of each quantile regression of the response times. An optional 'JAGS' engine translates the model into a plain 'JAGS' model file, for spike-and-slab priors and asymmetric Laplace (quantile) distributions, with marginal-likelihood fit indices (DIC, WAIC, PSIS-LOO) and posterior predictive checks.

License GPL-3

URL <https://github.com/jiewenTsai/birt>,
<https://github.com/jiewenTsai/birtExamples>

BugReports <https://github.com/jiewenTsai/birt/issues>

Encoding UTF-8

Depends R (>= 4.1.0)

Imports lavaan, RTMB (≥ 1.7), coda, dplyr ($\geq 1.1.0$), tibble ($\geq 3.0.0$), loo, statmod, stats, utils, graphics, grDevices, parallel

Suggests R2jags, rjags, mirt, Matrix, callr, pkgload, testthat ($\geq 3.0.0$), knitr, rmarkdown, strucchange, sandwich, partykit

SystemRequirements JAGS ($\geq 4.3.0$) for engine = ``jags" (optional)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.2

Config/pak/sysreqs jags

Repository <https://jiewentsai.r-universe.dev>

Date/Publication 2026-10-29 06:14:17 UTC

RemoteUrl <https://github.com/jiwenTsai/birt>

RemoteRef HEAD

RemoteSha 1b10ffa1ae8f4f368defdbcd3c6dd6be9c66016

Contents

add_fit_indices	3
ald	4
algorithm	5
birt	6
birt-class	11
birt_print	12
birt_quantile	13
birt_rtmb	14
birt_summary	15
check_sbc	16
compare	18
cond_reliability	19
convergence	20
dif_tree	21
dpriors	23
equations	25
estfun.birt_rtmb	26
hgrm	27
ipcr	29
jags_code	31
ordinal_information	31
plot.birt	32
plot.birt_rtmb	33
ppc	34
quantile_information	35
quantile_score	36

<i>add_fit_indices</i>	3
quantile_test	39
quantiles	40
rt_information	41
rtirt_syntax	42
rtmb_code	43
rtmb_control	44
score_test	46
shash	48
sim_hgrm	49
sim_rtirt	50
subset_chains	51
Index	52

<code>add_fit_indices</code>	<i>Compute and store the marginal-likelihood fit indices</i>
------------------------------	--

Description

Integrates the latent variables out (see Details of [birt()]) to get the pointwise log-likelihood, then DIC, WAIC and PSIS-LOO. Called by [birt()] unless ‘fit_indices = FALSE’.

Usage

```
add_fit_indices(fit, n_draws = 1000, Q = 41, progress = TRUE)
```

Arguments

<code>fit</code>	A ‘birt’ object.
<code>n_draws</code>	Number of posterior draws used (reduced automatically when ALD adds quadrature layers).
<code>Q</code>	Gauss-Hermite nodes.
<code>progress</code>	Show progress.

Value

‘fit’ with ‘fit_indices’, ‘loglik’, ‘loo’ and ‘waic’ added.

ald

*Residual distributions for latent variables and continuous indicators***Description**

Used in the ‘family’ argument of [birt()] to replace the default normal distribution of a latent variable (its residual, in a latent regression) or of a continuous indicator by an asymmetric Laplace distribution (ALD), which turns the corresponding regression into a quantile regression (Yu & Moyeed, 2001).

Usage

```
ald(p = 0.5)

normal()
```

Arguments

p Quantile level(s) in (0, 1). A vector (for one entry of ‘family’ only) fits one model per quantile and returns a ‘birt_quantile’ object.

Details

‘family = list(speed = ald(0.25))’ makes the 0.25 quantile of ‘speed’ given its predictors equal to the linear predictor (a latent quantile regression; Wang, Feng & Song, 2016; Zhu, Gao & Zhang, 2021). ‘family = list(continuous = ald(0.5))’ gives every continuous indicator a median (least absolute deviation) measurement model, which is robust to outlying values such as rapid guesses. Names are resolved by their role in the model syntax: a latent variable name is the structural level, an indicator name the measurement level, and ‘.continuous’ all continuous indicators. Binary indicators always use the logit link.

ALD on a latent variable whose variance is fixed (e.g. ‘theta’, variance 1 for identification) fixes the ALD scale so that the implied variance equals the fixed value; with a free variance the scale is estimated. The factor correlation ‘f1 ~~ f2’ cannot be combined with ALD; write the relation as a regression ‘f2 ~ f1’.

Sampling uses the normal-exponential mixture of Kozumi and Kobayashi (2011). Posterior intervals under a working ALD likelihood can be too narrow (Yang, Wang & He, 2016).

‘normal()’ is the default; a heteroscedastic residual is written in the model syntax, ‘V(t1) ~ theta’ (log residual SD linear in moderators).

Value

An object of class ‘birt_family’.

Examples

```
ald(0.25)
ald(c(0.1, 0.5, 0.9))
```

algorithm

*The estimation algorithm of a fit***Description**

Writes out how a fit was computed, as numbered steps with their formulas: for JAGS fits the full conditionals and the sampler JAGS assigned to every node (from `'rjags::list.samplers()'` on the compiled model) and the MCMC settings; for RTMB fits the marginal likelihood by the Laplace approximation or adaptive Gauss-Hermite quadrature (centres, nodes, optimization rounds, standard errors); for `'method = "elgm"'` the inner (per-person) and outer (parameter) quadrature, the node weights, the log marginal likelihood and how the posterior summaries are formed. The numbers (nodes, rounds, hyperparameters) are those of the fit. The model itself is given by `[equations()]`.

Usage

```
algorithm(object, ...)

## Default S3 method:
algorithm(object, ...)

## S3 method for class 'birt'
algorithm(object, format = c("unicode", "latex"), samplers = TRUE, ...)

## S3 method for class 'birt_quantile'
algorithm(object, format = c("unicode", "latex"), samplers = TRUE, ...)

## S3 method for class 'birt_rtmb'
algorithm(object, format = c("unicode", "latex"), ...)
```

Arguments

<code>object</code>	A fit of <code>[birt()]</code> (either engine) or <code>[hgrm()]</code> .
<code>...</code>	Unused.
<code>format</code>	<code>"unicode"</code> or <code>"latex"</code> .
<code>samplers</code>	JAGS fits: compile the model (without sampling) to list the samplers JAGS assigns (needs <code>rjags</code> ; a second or so).

Details

The same generic is exported by `birtRcpp` (Gibbs full conditionals and EM steps of its models); either package's `'algorithm()'` dispatches to the other's fits.

Value

A `'birt_algorithm'` object (printed like `[equations()]`; `'as.character()'` gives the lines).

Examples

```
if (requireNamespace("RTMB", quietly = TRUE)) {
  d <- sim_rtirt(N = 200, K = 3, seed = 1)
  m <- rtirt_syntax(paste0("y", 1:3), paste0("t", 1:3))
  fit <- birt(m, d, control = rtmb_control(method = "aghq"), progress = FALSE)
  algorithm(fit)
  algorithm(fit, "latex")
}
```

birt

Fit a Bayesian IRT / response time model from lavaan-style syntax

Description

By default the model is fitted in 'RTMB' by ELGM (extended latent Gaussian model; Stringer, Brown & Stafford, 2023): approximate Bayesian inference on the marginal likelihood, the persons integrated out by adaptive Gauss-Hermite quadrature and the remaining parameters by an outer adaptive quadrature, with the priors of [dpriors()]. 'rtmb_control(method = "aghq")' gives maximum likelihood on the same marginal likelihood (needed by the score-based tools: [score_test()], [quantile_score()], [dif_tree()]); 'engine = "jags"' samples the conditional likelihood (persons sampled with the parameters) with [R2jags::jags.parallel()], for spike-and-slab priors and [ald()] quantile models.

Usage

```
birt(
  model,
  data,
  binary = NULL,
  family = NULL,
  n_chains = 4,
  n_iter = 10000,
  n_burn = 5000,
  thin = NULL,
  seed = 2026,
  file = NULL,
  label = NULL,
  fit_indices = TRUE,
  progress = TRUE,
  engine = c("rtmb", "jags"),
  control = rtmb_control(),
  log_rt = FALSE,
  id = NULL,
  dp = dpriors(),
  ssp_p = NULL,
  ordered = NULL,
```

```

    itemtype = "grm",
    censor = NULL,
    ...
)

```

Arguments

model	Model syntax (see Details).
data	A data frame with the indicators and covariates.
binary	Names of the binary (0/1) indicators. By default every indicator whose observed values are all 0 or 1; all other indicators are continuous.
family	Named list of residual distributions, e.g. <code>'list(speed = ald(0.25))'</code> ; see <code>[ald()]</code> . Default: normal for every latent variable and continuous indicator.
n_chains, n_iter, n_burn, thin	MCMC settings of <code>'engine = "jags"'</code> : chains (run in parallel, on at most <code>'getOption("birt.cores")'</code> processes; default the number of physical cores), iterations per chain, burn-in, thinning (default: 1000 kept draws per chain).
seed	Random seed of the JAGS sampler (<code>'engine = "jags"'</code>).
file	Path for the generated JAGS model file (default: a temporary file).
label	Optional label shown in the output.
fit_indices	Compute the marginal-likelihood criteria after sampling.
progress	Show the progress of the fit: for the RTMB engine (the default) the steps as messages (Laplace start, AGHQ rounds with $-2\log L$, the ELGM outer nodes); for <code>'engine = "jags"'</code> a progress bar (the sampler then runs in a background R session with <code>'callr'</code>). <code>'FALSE'</code> : no progress output.
engine	<code>"rtmb"</code> (default): the marginal likelihood (the persons integrated out) in 'RTMB', by ELGM (default, see <code>[rtmb_control()]</code>): per-person AGHQ inside, adaptive quadrature over the hyperparameters outside; the same priors as JAGS, <code>'dp'</code> and <code>'prior(...)'</code> , no spike-and-slab) or by maximum likelihood (<code>'control = rtmb_control(method = "aghq")'</code> , which ignores the priors). <code>"jags"</code> : the conditional likelihood, Bayesian (the persons sampled together with the parameters). Instead of <code>[ald()]</code> the RTMB engine offers <code>[shash()]</code> residuals and moderated scales (<code>'V(y) ~ z'</code> in the syntax), and <code>[quantiles()]</code> gives the quantile effects at every level from one fit. Returns a <code>'birt_rtmb'</code> object.
control	RTMB settings, <code>[rtmb_control()]</code> : <code>'rtmb_control(method = "ml")'</code> (or <code>"aghq"</code>) for maximum likelihood. A plain list of its arguments, e.g. <code>'list(method = "ml")'</code> , is passed to <code>[rtmb_control()]</code> .
log_rt	<code>'TRUE'</code> : the continuous indicators are response times in seconds and are log-transformed before fitting (under the same names). A vector of column names log-transforms only those (e.g. when some continuous indicators are not times). Times ≤ 0 are an error, and a warning flags times that look already log-transformed (not right-skewed, median below 10).
id	Person identifiers for <code>[scores()]</code> : a column name of <code>'data'</code> or a vector (default <code>'1..N'</code>).

dp	Default priors, [dpriors()] (engine "jags", and engine "rtmb" with 'method = "elgm"); the prior of every parameter is shown in 'summary()'.
ssp_p	Prior inclusion probability of the spike-and-slab parameters: 'NULL' (default) learns it ('p_incl ~ beta(1, 1)', shared within loadings, within regressions and within the moderation effects of one kind and moderator); 'BF10' then uses the posterior mean of the learned inclusion probability as the prior odds ('prior_incl' in 'summary(fit)\$selection'). A number fixes it, e.g. '0.5' for Bayes factors with prior odds 1, or '0.2' when few nonzero effects are expected.
ordered	Ordinal (Likert) indicators: 'TRUE' (every indicator with integer values) or their names. They can be combined with binary and continuous indicators (e.g. response times) and two latent variables; see "Ordinal indicators" and "Moderation".
itemtype	Model of the ordinal indicators: "grm" (graded response, cumulative log-its), "gpcm" (generalized partial credit, adjacent-category logits) or "tppcm" (two-parameter partial credit: one discrimination per step); one value, or a named vector per item. "gpcm" and "tppcm" are experimental: with moderation statements they are not yet checked by simulation-based calibration.
censor	Censored continuous indicators (e.g. time limits): a named list with the limits of each, 'list(t1 = c(upper = 60), t2 = c(lower = 1, upper = 60))'. A value at or above 'upper' is right censored (only "at least 'upper'" is known; it contributes the log survival function, $\log P(y \geq \text{upper} f)$, to the likelihood) and a value at or below 'lower' is left censored ($\log P(y \leq \text{lower} f)$); other values contribute their density, as in the usual likelihood of censored data (Meeker & Escobar, 1998; for time limits in IRT, Lee & Ying, 2015). The limits are on the scale of the columns of 'data' as given: seconds for the columns log-transformed by 'log_rt', whose limits are log-transformed with them; on the data's own scale otherwise (e.g. 'log(60)' when 'data' already holds log times). Normal and [shash()] residuals; 'engine = "rtmb"' (ELGM and maximum likelihood). Values recorded beyond a limit are treated as censored at the limit. See "Censoring" in Details.
...	Not used: an argument that 'birt()' does not have is an error, and a setting of [rtmb_control()] (e.g. 'method = "ml"') is answered with the 'control = rtmb_control(...)' call to write.

Details

Syntax. At most two latent variables. Operators:

'f =~ y1 + y2'	loadings: the first '=' lines of a factor are its measurement loadings (positive), also when
'f =~ -1*t1'	fixed loading
'f =~ r*t1 + r*t2'	equality by label
'f =~ prior("ssp")*t1'	spike-and-slab prior
'f =~ prior("dnorm(0, 100)")*t1'	any JAGS prior (precision parameterization)
'f1 ~~ f2'	factor correlation (default 0)
'f ~~ 1*f'	fixed (residual) variance
'f ~ x1 + prior("ssp")*x2'	latent regression on covariates
'f2 ~ f1'	path between the latent variables

****Default priors.**** Measurement loadings hierarchical lognormal per latent variable ($\log \lambda \sim N(\mu_f, 1)$, $\mu_f \sim N(0, 10)$; see `[dpriors()]`); cross-loadings and regression coefficients $\text{dnorm}(0, 1)$; intercepts $\text{dnorm}(0, 1/9)$ (binary) and $\text{dnorm}(0, 1/100)$ (continuous); residual SDs, factor SDs and ALD scales half-t(3); correlation uniform(-1, 1); change them with `'dp = dpriors(...)'`. Spike-and-slab: $\text{incl} \sim \text{dbern}(p_{\text{incl}})$, $p_{\text{incl}} \sim \text{beta}(1, 1)$ (or fixed, `'ssp_p'`), slab $N(0, \sigma_{\text{slab}}^2)$, $\sigma_{\text{slab}} \sim N+(0, 1)$, shared within loadings, within regression coefficients and within the moderation effects of one kind (shifts $E(y) \sim z$ or log loadings $E(y) \sim z:f$) and moderator.

****Censoring**** (`'censor'`). With a time limit c the recorded time of a person who ran out of time is not their response time; treating it as observed biases the intercepts, the residual SDs and the speed variance downwards. `'censor'` replaces the density of such a value by the probability of being beyond the limit, $S(c | f) = 1 - \Phi(T^{-1}((c - \mu_{ij})/\sigma_{ij}))$ (T the identity for normal residuals, the standardized sinh-arcsinh transform for `[shash()]`), inside the integral over the person latents. The number of censored values is shown in the header; `[cond_reliability()]` of a censored fit gives the information under its limits, and `[rt_information()]` the information kept under other limits.

Value

With the default `'engine = "rtmb"'`, an object of class `'birt_rtmb'` (see `[birt_rtmb]` for its accessors and `[plot.birt_rtmb()]`). With `'engine = "jags"'`, an object of class `'birt'` (a list), or `'birt_quantile'` when a vector of quantiles is given in `'family'`; see `[birt-class]`.

References

- Lee, Y.-H., & Ying, Z. (2015). A mixture cure-rate model for responses and response times in time-limit tests. *Psychometrika*, 80(3), 748-775. doi:10.1007/s1133601494198
- Meeker, W. Q., & Escobar, L. A. (1998). *Statistical methods for reliability data*. Wiley.

Examples

```
d <- sim_rtirt(N = 300, K = 5, seed = 1)
model <- rtirt_syntax(paste0("y", 1:5), paste0("t", 1:5))
cat(model)

# the default: approximate Bayes (ELGM) in RTMB
fit <- birt(model, d, progress = FALSE)
summary(fit)
scores(fit)
# maximum likelihood: standard errors, p-values, AIC / BIC, score tests
f_ml <- birt(model, d, control = rtmb_control(method = "ml"), progress = FALSE)
summary(f_ml)
# JAGS engine (needs the JAGS library): spike-and-slab cross-loadings
if (requireNamespace("rjags", quietly = TRUE)) {
  m_ssp <- rtirt_syntax(paste0("y", 1:5), paste0("t", 1:5), cross = "ssp")
  f_jags <- birt(m_ssp,
    data = d, engine = "jags", n_chains = 2, n_iter = 2000, n_burn = 1000,
    progress = FALSE
  )
  summary(f_jags)
}
```

Description

A JAGS fit of `[birt()]` is a list of class `'c("birt", "birt_fit")'` with these elements and the accessors `'scores()'`, `'reliability()'`, `'fit_indices()'`, `'pointwise_loglik()'`, `'posterior_draws()'`, `'convergence()'`, `'jags_code()'`, `'compare()'` (RTMB fits, including `[hgrm()]`, are of class `'birt_rtmb'`). The tables returned by `'estimates()'`, `'scores()'`, `'fit_indices()'`, `'convergence()'` and `'compare()'` are tibbles, and `'reliability()'` and `'coef()'` named numbers; they print with 3 decimals but keep the full values (see `[birt_print()]`):

'estimates' lavaan-like parameter table: `'lhs'`, `'op'`, `'rhs'`, `'family'`, `'prior'`, posterior `'est'` (mean), `'sd'`, `'q025'`, `'q975'`, `'rhat'`, `'ess'`, ALD `'scale'`, spike-and-slab `'p_incl'` and `'BF10'` (posterior over prior odds of inclusion; the prior odds from `'ssp_p'`, or from the posterior mean of the learned inclusion probability of the parameter's group). Rows with `'op'` `'~~'` report variances (for ALD: the implied variance, with the scale in `'scale'`). `'estimates()'` returns it with the columns `'se'`, `'lower'`, `'upper'` in place of `'sd'`, `'q025'`, `'q975'`: the same names for every fit (JAGS, ELGM, maximum likelihood), as in `'as.data.frame(summary(fit))'`. The old names still work with `'$'`, `'[['` and `'['` (with a warning once per session).

'scores' one row per person: EAP (posterior mean) and posterior SD of each latent variable.

'reliability' empirical reliability of the EAP scores, $\text{var}(\text{EAP}) / (\text{var}(\text{EAP}) + \text{mean PSD}^2)$.

'items' item tables: binary items (a , $b = -d/a$, d) and continuous indicators (intercept, loadings, residual scale and variance, inclusion probability).

'loglik' draws x persons matrix of the marginal log-likelihood (latent variables integrated out); `'loo'`, `'waic'` the corresponding `'loo'` objects; `'fit_indices'` Deviance, pD, DIC, WAIC, LOOIC.

'draws' the posterior draws (`[coda::mcmc.list]`), including person parameters.

'code', **'file'**, **'partab'**, **'spec'**, **'data'**, **'mcmc'**, **'secs'** JAGS code and file, internal parameter table, parsed model, JAGS data, MCMC settings, run time.

Usage

```
estimates(object, ...)
```

```
## S3 method for class 'birt'
estimates(object, ...)
```

```
scores(object, ...)
```

```
## S3 method for class 'birt_fit'
scores(object, ...)
```

```
reliability(object, ...)
```

```

## S3 method for class 'birt_fit'
reliability(object, ...)

fit_indices(object, ...)

## S3 method for class 'birt_fit'
fit_indices(object, ...)

pointwise_loglik(object, ...)

posterior_draws(object, persons = FALSE, ...)

## S3 method for class 'birt'
coef(object, ...)

## S3 method for class 'birt'
print(x, ...)

## S3 method for class 'birt'
summary(object, digits = 3, irt = TRUE, ...)

```

Arguments

object, x	A ‘birt’ (or ‘birt_quantile’) object.
...	Unused.
persons	Include the person parameters.
digits	Number of decimals.
irt	Also show binary items in the IRT parameterization (a, b).

birt_print

Printing of birt tables and numbers

Description

The tables that birt returns (‘estimates()’, ‘scores()’, ‘fit_indices()’, ‘quantiles()’, ‘compare()’, ...) are tibbles or data frames of class ‘birt_tbl’, and the named numbers (‘reliability()’, ‘coef()’) are numeric vectors or matrices of class ‘birt_values’. They print numbers with 3 decimals in fixed notation (6e-17 prints 0.000; only numbers of 1e10 or more, such as a diverging estimate, print as e.g. 5.950e+99), counts (N, npar, df, nodes, ESS) without decimals, and p-values below 0.001 as ‘<0.001’. Only the display is rounded: the values themselves are not (‘x\$est’, ‘x[["ability"]]', arithmetic and ‘as.data.frame()’ use the full values; ‘unclass()’ removes the class).

Usage

```
## S3 method for class 'birt_tbl'
format(x, n = NULL, digits = 3, width = getOption("width"), ...)

## S3 method for class 'birt_tbl'
print(x, n = NULL, digits = 3, width = getOption("width"), ...)

## S3 method for class 'birt_values'
format(x, digits = 3, ...)

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

## S3 method for class 'birt_values'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	A 'birt_tbl' or 'birt_values' object.
n	Number of rows to print (default: all rows of tables with up to 20 rows, otherwise the first 10); 'Inf' prints all rows.
digits	Decimals.
width	Width of the printed lines (columns that do not fit continue below).
...	Unused.
row.names, optional	Passed to [as.data.frame()].

Value

'print()' returns 'x' invisibly; 'format()' the printed lines ('birt_tbl') or the formatted numbers ('birt_values').

birt_quantile

Quantile fits: several quantiles of one latent variable or indicator

Description

'birt(..., family = list(speed = ald(c(.1, .5, .9))))' fits one model per quantile and returns a 'birt_quantile' object: 'fits' (list of 'birt' objects), 'p', 'target'.

Usage

```
## S3 method for class 'birt_quantile'
estimates(object, ...)

## S3 method for class 'birt_quantile'
coef(object, ...)

## S3 method for class 'birt_quantile'
reliability(object, ...)

## S3 method for class 'birt_quantile'
fit_indices(object, ...)

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

## S3 method for class 'birt_quantile'
summary(object, ...)

## S3 method for class 'birt_quantile'
plot(x, terms = NULL, ...)
```

Arguments

object, x	A ‘birt_quantile’ object.
...	Unused.
digits	Number of decimals.
terms	Regression terms to plot (default: all regressions of the target).

birt_rtmb

The ‘birt_rtmb’ object (RTMB engine)

Description

A model fitted with ‘birt(..., engine = "rtmb")’: maximum likelihood with the latent variables integrated out by adaptive Gauss-Hermite quadrature (or Laplace). Accessors as for JAGS fits: ‘print’, ‘summary’, ‘coef’, [estimates()] (‘est’, ‘se’, ‘lower’, ‘upper’, and for maximum likelihood ‘z’ and ‘pvalue’; for ELGM the posterior mean, SD and 95% interval), [scores()] (EAP and posterior SD from the quadrature), [reliability()], [fit_indices()] (log-likelihood, AIC, BIC), [convergence()], ‘logLik’, ‘vcov’, ‘AIC’, ‘BIC’, [quantiles()] (quantile effects) and ‘plot’ (see [plot.birt_rtmb()]: scores, test information, quantile curves).

Usage

```
## S3 method for class 'birt_rtmb'
print(x, ...)

## S3 method for class 'birt_rtmb'
summary(object, digits = 3, ...)

## S3 method for class 'birt_rtmb'
estimates(object, ...)

## S3 method for class 'birt_rtmb'
coef(object, ...)

## S3 method for class 'birt_rtmb'
vcov(object, ...)

## S3 method for class 'birt_rtmb'
logLik(object, ...)

## S3 method for class 'birt_rtmb'
scores(object, ...)

## S3 method for class 'birt_rtmb'
reliability(object, ...)

## S3 method for class 'birt_rtmb'
fit_indices(object, ...)

## S3 method for class 'birt_rtmb'
convergence(object, ...)
```

Arguments

...	Unused, or passed to [graphics::plot()].
object, x	A 'birt_rtmb' fit.
digits	Number of decimals.

birt_summary	Summary tables
--------------	----------------

Description

'summary()' of a birt fit returns the tables of the fit as one object: 'header', 'fit' (fit indices), 'items' and 'se' (one row per item: estimates and their posterior SD or standard error, as 'coef(simplify = TRUE)' in mirt), 'parameters' (one row per parameter with its interval and, for Bayesian fits, its

prior, as in blavaan) and model-specific tables (e.g. ‘selection’, ‘moderation’, ‘precision’, ‘reliability’). Each element is a tibble; ‘as.data.frame()’ returns ‘parameters’ as a plain data frame. ‘print()’ shows the tables under their names, so ‘\$items’ gets the one printed as ‘\$items’.

Usage

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

## S3 method for class 'birt_summary'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

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

Arguments

x	A ‘birt_summary’, or one of its tables.
tables	Names of the tables to print (default all).
digits	Decimals.
...	Unused.
row.names, optional	Unused.
title	Print the title of a table.

Value

‘print()’ returns ‘x’ invisibly; ‘as.data.frame()’ the parameter table.

check_sbc

Simulation-based calibration of a JAGS model

Description

Checks that a model is fitted correctly by the JAGS engine. Each replication draws the parameters from their priors and data from the model (persons, covariates, moderators and the pattern of missing values as in ‘data’), fits the model with [birt()] and records the rank of every true parameter value among the posterior draws. If the model, the priors and the sampler agree, each rank is uniform (Talts et al., 2018). The simulator is written in R from the definition of the model, independently of the JAGS code that [birt()] generates, so the check also tests the translation of the syntax into JAGS.

Usage

```

check_sbc(
  model,
  data,
  n_rep = 100,
  n_draws = 100,
  binary = NULL,
  family = NULL,
  ordered = NULL,
  itemtype = "grm",
  dp = dpriors(),
  ssp_p = NULL,
  n_chains = 2,
  n_iter = 2000,
  n_burn = 1000,
  bins = 10,
  seed = 1,
  progress = TRUE
)

## S3 method for class 'birt_sbc'
print(x, ...)

## S3 method for class 'birt_sbc'
plot(x, ...)

```

Arguments

model, data, binary, family, ordered, itemtype, dp, ssp_p	As in [birt()]; ‘data’ gives the persons, covariates, moderators and missing values.
n_rep	Number of replications (100 or more for a real check; each one is a JAGS fit).
n_draws	Posterior draws used for each rank (equally spaced, which also thins).
n_chains, n_iter, n_burn	MCMC settings of each fit.
bins	Number of bins of the uniformity test.
seed	Random seed.
progress	Show progress.
x	A ‘birt_sbc’ object.
...	Unused.

Details

Ties (spike-and-slab parameters at 0) get randomized ranks. Replications whose data would change the model (an ordinal category or a binary value that is never observed) are drawn again. Asymmetric Laplace families are not simulated.

Value

A ‘birt_sbc’ object: ‘ranks’ (replications x parameters), ‘test’ (per parameter: the chi-square uniformity statistic and p-value, and the mean rank relative to its expectation), ‘redraws’ and ‘rhat’ (the largest R-hat of each replication; ranks are only meaningful when the chains converged, Talts et al., 2018, Section 5). ‘plot()’ shows the rank histograms.

References

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. (2018). Validating Bayesian inference algorithms with simulation-based calibration. arXiv:1804.06788.

Examples

```
if (requireNamespace("rjags", quietly = TRUE)) {
  d <- sim_rtirt(N = 100, K = 3, seed = 1)
  m <- rtirt_syntax(paste0("y", 1:3), paste0("t", 1:3))
  s <- check_sbc(m, d, n_rep = 20, n_iter = 1000, n_burn = 500, progress = FALSE)
  s
}
```

compare

Compare fitted models

Description

Compare fitted models

Usage

```
compare(..., digits = 3)
```

Arguments

...	‘birt’ objects (named arguments give the model names), or one named list. For RTMB fits (‘birt_rtmb’): $-2 \log L$, number of parameters, AIC, BIC and their differences to the best model.
digits	Number of decimals.

Value

Invisibly, a tibble: for JAGS fits the reliability, DIC, WAIC, LOOIC and the ELPD difference to the best model (from `[loo::loo_compare()]`); for ELGM fits the log marginal likelihood ‘logML’, its difference to the best model ‘dlogML’ and the Bayes factor of the best model against each model ‘BF_best’ (numeric); for maximum likelihood fits the deviance, AIC, BIC and their differences to the best model.

cond_reliability	<i>Conditional reliability of the first latent variable</i>
------------------	---

Description

For a model fitted with ‘engine = "rtmb"’, the reliability of the first latent variable (e.g. ability) as a function of its value, $\rho(\theta) = I(\theta)/(I(\theta) + 1/\sigma^2)$ (Nicewander, 2018), where $I(\theta)$ is the expected Fisher information of all indicators about θ , with the second latent variable (speed) treated as a nuisance (Schur complement, at its conditional median) and σ^2 the variance of θ given the covariates. The information is split into the accuracy part (categorical indicators: binary and ordinal), the response-time part (continuous indicators only) and both; for distributional models (‘V(y) ~ f’ in the syntax, ‘shash(skew =)’ moderated by the latent variables) the scale and the shape of the response-time distribution contribute too.

Usage

```
cond_reliability(
  object,
  theta = seq(-3, 3, by = 0.1),
  at = NULL,
  se = TRUE,
  bins = 10,
  k = 41
)

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

Arguments

object	A fit from ‘birt(..., engine = "rtmb")’.
theta	Values of the first latent variable.
at	Values of the covariates and moderators (a named vector); default their means.
se	Delta-method standard errors of the curves (from ‘vcov()’).
bins	Number of EAP bins of the empirical check (0: none).
k	Gauss-Hermite nodes over SHASH residuals.
x	A ‘birt_cond_rel’ object.
...	Passed to [graphics::plot()].

Details

* Ordinal indicators contribute their item information (graded response: Samejima, 1969; partial credit: the variance of the cumulative step discriminations), the same as [ordinal_information()]. * Moderated loadings and shifts (‘E(y) ~ z:f’, ‘E(y) ~ z’) are evaluated at the moderator values ‘at’ (default: their means), so the curve is the reliability for persons with these moderator values. * For

a fit with censored indicators (`birt(..., censor = )`) the information is that of the censored data at the fit's limits: the density part between the limits plus the probability of a censored value times the squared score of $\log S(c)$ or $\log F(c)$ (Escobar & Meeker, 1998). `[rt_information()]` shows what other limits would keep.

The curve is checked against the data: `'empirical'` bins the persons by their EAP and compares $1 - \text{PSD}^2/\sigma^2$ from the adaptive quadrature with the curve at the bin mean.

Value

A data frame (class `'birt_cond_rel'`) with `'theta'`, the information `'info'`, `'info_acc'`, `'info_rt'` and the reliabilities `'rel'`, `'rel_acc'`, `'rel_rt'` (and `'se_*`' with `'se = TRUE'`); attribute `'empirical'` with the binned check. `'plot()'` draws the curves.

References

- Nicewander, W. A. (2018). Conditional reliability coefficients for test scores. *Psychological Methods*, 23*(2), 351-362.
- Escobar, L. A., & Meeker, W. Q. (1998). Fisher information matrices with censoring, truncation, and explanatory variables. *Statistica Sinica*, 8*, 221-237.
- Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement*, 34*(4, Pt. 2).

Examples

```
d <- sim_rtirt(N = 500, K = 6, seed = 1)
m <- paste("theta =~ y1 + y2 + y3 + y4 + y5 + y6",
  "speed =~ -1*t1 + -1*t2 + -1*t3 + -1*t4 + -1*t5 + -1*t6",
  "theta =~ t1 + t2 + t3 + t4 + t5 + t6",
  sep = "\n"
)
fit <- birt(paste(m, "V(t1 + t2 + t3 + t4 + t5 + t6) ~ theta", sep = "\n"), d,
  control = rtmb_control(method = "aghq"), progress = FALSE
)
cr <- cond_reliability(fit)
plot(cr)
```

convergence

Convergence summary

Description

JAGS fits: R-hat and effective sample sizes, and the parameters with R-hat above `'threshold'`.
 RTMB fits: optimizer code, maximum gradient and the AGHQ rounds.

Usage

```
convergence(object, ...)
```

```
## S3 method for class 'birt_fit'
```

```
convergence(object, threshold = 1.05, ...)
```

Arguments

object	A ‘birt’ or ‘birt_rtmb’ fit.
...	Unused.
threshold	Rhat threshold for listing parameters.

Value

A tibble of the parameters with Rhat above ‘threshold’ (invisibly).

dif_tree	<i>DIF tree (deprecated)</i>
----------	------------------------------

Description

****Deprecated.**** ‘dif_tree()’ still works, with a warning once per session, but gets no new features and will be removed in a later version. To find which parameters vary with person covariates, use [ipcr()] (individual parameter contribution regression) on a maximum likelihood fit.

Usage

```
dif_tree(
  model,
  data,
  partition,
  impact = TRUE,
  nbins = 10,
  alpha = 0.05,
  minsize = NULL,
  maxdepth = Inf,
  ...
)
```

Arguments

model	Model syntax, as in [birt()].
data	A data frame with the indicators.
partition	A data frame of partitioning variables (one row per person; numeric, ordered or factor).

impact	Model latent mean differences along the partitioning variables.
nbins	Numeric partitioning variables with more than ‘nbins’ distinct values are cut into ‘nbins’ groups at their quantiles (an ordered factor; the tree prints the groups as intervals). A split is then searched among the ‘nbins - 1’ group boundaries (two maximum likelihood fits each) instead of every observed value (about two fits per person in the node), which makes the tree minutes faster; the cost is that a cut point is located only up to its quantile group. ‘NULL’ keeps them numeric (the supLM test and the exhaustive search of [partykit::mob()]).
alpha, minsize, maxdepth	Significance level (Bonferroni over variables), minimum node size and maximum depth (see [partykit::mob_control()]).
...	Passed to [birt()] (e.g. ‘ordered’, ‘family’; ‘control’ defaults to the maximum likelihood fit, ‘rtmb_control(method = "aghq")’, whose scores the tests use).

Details

Recursively splits the persons by the partitioning variables where the measurement parameters of the binary or ordinal items (loadings, intercepts, thresholds) are not invariant: model-based recursive partitioning (Zeileis, Hothorn & Hornik, 2008; for IRT, Strobl, Kopf & Zeileis, 2015). At each node the model is fitted by maximum likelihood (‘engine = "rtmb"’) and the casewise scores of the measurement parameters ([estfun.birt_rtmb()]) are tested along every partitioning variable with fluctuation tests, as in [score_test()]; the most significant variable (Bonferroni-adjusted) is split if it is below ‘alpha’.

****Impact.**** With ‘impact = TRUE’ (default) the partitioning variables are covariates of every latent variable in each node (‘f ~ z’, centred and scaled within the node; factors as dummies), so differences in the latent means between groups are modelled and the tree looks for DIF beyond impact. With ‘impact = FALSE’ any variable related to the latent variables splits the tree.

****Tests.**** Numeric partitioning variables are tested by the supLM (maxLM) statistic, ordered factors (including the quantile groups of ‘nbins’) by its version for ordinal variables (maxLM_o, ‘mob_control(ordinal = "L2")’; Merkle, Fan & Zeileis, 2014) and unordered factors by the chi-square statistic.

****ELGM.**** The tree is a maximum likelihood method: each node is fitted with ‘rtmb_control(method = "aghq")’ (the default ‘control’) and an ELGM ‘control’ is refused, since the fluctuation tests need casewise scores at the ML estimate (see "ELGM fits" in [score_test()]). There is no ELGM counterpart of the recursive search; a single candidate split can be checked by comparing ELGM fits with and without ‘E(y) ~ z’ ([compare()]).

Only the scores of the measurement block are tested (decorrelated within the block), so partitioning variables built from the response times (e.g. the total log time) are allowed. Parameters are on the scale of each node, so compare nodes through the tests rather than raw parameter differences.

Value

A ‘modelparty’ object (partykit): ‘print()’, ‘plot()’, ‘coef()’ (measurement parameters per terminal node, internal scale), ‘strucchange::sctest(tree, node =)’ for the tests at a node.

References

- Debelak, R., Pawel, S., Strobl, C., & Merkle, E. C. (2022). Score-based measurement invariance checks for Bayesian maximum-a-posteriori estimates in item response theory. *British Journal of Mathematical and Statistical Psychology*, 75*(3), 728-752. doi:10.1111/bmsp.12275
- Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17*, 492-514. doi:10.1198/106186008X319331
- Merkle, E. C., Fan, J., & Zeileis, A. (2014). Testing for measurement invariance with respect to an ordinal variable. *Psychometrika*, 79*, 569-584. doi:10.1007/s1133601393767
- Strobl, C., Kopf, J., & Zeileis, A. (2015). Rasch trees: A new method for detecting differential item functioning in the Rasch model. *Psychometrika*, 80*, 289-316. doi:10.1007/s1133601393883

Examples

```
if (requireNamespace("partykit", quietly = TRUE) && requireNamespace("RTMB", quietly = TRUE)) {
  d <- sim_rtirt(N = 600, K = 6, seed = 1)
  z <- data.frame(group = factor(sample(c("A", "B"), 600, TRUE)), x = rnorm(600))
  dif_tree(rtirt_syntax(paste0("y", 1:6), paste0("t", 1:6)), d, z) # no DIF: a single node
}
```

dpriors

Default priors of a JAGS fit

Description

As ‘dpriors()’ in blavaan: the default prior of each parameter class, as JAGS distributions (normal priors in the ****precision**** parameterization, ‘dnorm(mean, 1 / sd^2)’). A ‘prior(...)’ modifier in the model syntax overrides the default for that parameter (loadings and regression coefficients); ‘prior("ssp")’ gives it a spike-and-slab prior. The prior of every parameter is listed in the ‘prior’ column of ‘summary()’ and [estimates()].

Usage

```
dpriors(
  loading = "hier",
  cross = "dnorm(0, 1)",
  beta = "dnorm(0, 1)",
  intercept_binary = "dnorm(0, 1/9)",
  intercept = "dnorm(0, 1/100)",
  resid_sd = "dt(0, 1, 3) T(0,)",
  factor_sd = "dt(0, 1, 3) T(0,)",
  cor = "dunif(-1, 1)",
  slab_sd = "dnorm(0, 1) T(0,)",
  moderation = "dnorm(0, 1)",
  threshold = "dnorm(0, 1/4)",
  threshold_step = "dlnorm(-0.5, 1)",
```

```

    step_ratio = "dlnorm(0, 1)",
    skew = "dnorm(0, 1)",
    tail = "dlnorm(0, 4)"
  )

```

Arguments

loading	Measurement loadings (the first ‘=~’ lines of a factor, before any line of the other factor; positive). The default ‘"hier"’ is a hierarchical lognormal prior per latent variable, $\log \lambda_i \sim N(\mu_f, 1)$, $\mu_f \sim N(0, 10)$: the spread of ‘dlnorm(0, 1)’, with a centre that adapts to the units of the indicators. A fixed prior such as ‘"dlnorm(0, 1)"’ (the earlier default) pulls all loadings of a latent variable towards 1; with many indicators on another scale (log response times: loadings about 0.1-0.3) this shrinks the latent variance and biases its correlation (with 170 items and 723 persons: .66 against .52).
cross	Cross-loadings (later ‘=~’ lines).
beta	Regression coefficients (‘~’).
intercept_binary, intercept	Intercepts of binary (logit) and continuous indicators.
resid_sd	Residual SD of continuous indicators (for [ald()]: of the implied SD).
factor_sd	SD of a latent variable with a free variance.
cor	Correlation of the latent variables (‘f1 ~~ f2’).
slab_sd	SD of the spike-and-slab slab (shared within loadings / within regressions).
moderation	Log-scale moderation effects: ‘E(y) ~ z:f’ (log loading), ‘V() ~ z’ (log SD) and the ‘skew =’ moderators of [shash()].
threshold, threshold_step	Ordinal indicators: the first threshold (graded response; every step difficulty of a partial credit item), and the positive steps between the next thresholds of a graded response item (lognormal; Bayesian engines).
step_ratio	Ratio of a later step discrimination to the first one (‘itemtype = "tppcm"’).
skew, tail	Engine ‘"rtmb"’ with ‘method = "elgm"’ only: the SHASH skewness and the SHASH tail weight (‘dlnorm(0, 4)’: log tail weight with SD 0.5, centred at the normal).

Details

The same priors are used by the approximate Bayesian (ELGM) fit of the RTMB engine, ‘birt(..., engine = "rtmb", control = rtmb_control(method = "elgm"))’, which supports ‘dnorm’, ‘dlnorm’, ‘dt’, ‘dunif’, ‘dgamma’, ‘dexp’, ‘dbeta’, ‘ddexp’ and ‘dlogis’, with truncation ‘T(a, b)’.

Value

A named list of prior strings (class ‘birt_dpriors’).

Examples

```
dpriors()
dpriors(cross = "dnorm(0, 100)") # BSEM-style small-variance cross-loadings (SD 0.1)
dpriors(loading = "dlnorm(0, 1)") # the earlier fixed loading prior
```

equations

Model equations

Description

Writes the fitted model as equations, with the parameters as they appear in [summary()], the sets of items they apply to and, for Bayesian fits, the priors. The default is Unicode text for the console; ‘format = "latex"’ gives an ‘aligned’ block for a paper or an appendix (e.g. ‘writeLines(as.character(equations(fit, "latex")), "model.tex")’).

Usage

```
equations(object, ...)

## S3 method for class 'birt_equations'
as.character(x, ...)

## S3 method for class 'birt_equations'
print(x, ...)

## S3 method for class 'birt'
equations(object, format = c("unicode", "latex"), priors = TRUE, ...)

## S3 method for class 'birt_rtmb'
equations(object, format = c("unicode", "latex"), priors = TRUE, ...)

## S3 method for class 'birt_quantile'
equations(object, format = c("unicode", "latex"), priors = TRUE, ...)
```

Arguments

object	A fitted model (‘birt()’ with either engine, or ‘hgrm()’).
...	Unused.
x	A ‘birt_equations’ object.
format	“unicode” or “latex”.
priors	Include the priors (Bayesian fits).

Value

A ‘birt_equations’ object (printed); ‘as.character()’ gives the lines.

Examples

```
cat(rtirt_syntax(paste0("y", 1:3), paste0("t", 1:3)))

if (requireNamespace("RTMB", quietly = TRUE)) {
  d <- sim_rtirt(N = 200, K = 3, seed = 1)
  fit <- birt(rtirt_syntax(paste0("y", 1:3), paste0("t", 1:3)), d,
    control = rtmb_control(method = "aghq"), progress = FALSE
  )
  equations(fit)
  equations(fit, "latex")
}
```

estfun.birt_rtmb

Casewise scores of an RTMB fit

Description

‘estfun()’ returns, for a maximum likelihood fit of [birt()] with ‘engine = "rtmb"’ (adaptive Gauss-Hermite quadrature, also for ‘ordered = TRUE’), the matrix of casewise scores: one row per person, one column per free parameter, the derivative of that person’s marginal log-likelihood (latent variables integrated out, with the quadrature nodes of the final objective) at the estimates. The columns sum to the negative gradient of the objective, about zero. Parameters are on the internal scale of the fit (log of loadings constrained positive, log SDs, atanh of correlations; for ordinal items log a, the first threshold and log increments); the attribute ‘parameters’ describes each column (‘label’, indicator ‘item’, ‘block’: “measurement” for binary and ordinal items, “rt” for continuous indicators, “structural”). These are the estimating functions used by [score_test()] and by ‘struchange::gefp()’. An ELGM fit is refused (see “ELGM fits” in [score_test()]).

Usage

```
## S3 method for class 'birt_rtmb'
estfun(x, ...)
```

```
## S3 method for class 'birt_rtmb'
bread(x, ...)
```

Arguments

x	A fit of ‘birt(..., engine = "rtmb")’ (maximum likelihood, ‘rtmb_control(method = "aghq")’).
...	Unused.

Value

A persons x parameters matrix.

‘bread()’: the inverse of the average observed information (internal scale), as in the ‘sandwich’ package.

hgrm

*Moderated (heteroscedastic) graded response model***Description**

A graded response model whose item discriminations and thresholds, and the mean and variance of the latent trait, depend on person-level moderators (moderated nonlinear factor analysis for ordinal items; Bauer, 2017). The typical use is one response time per person (e.g. the total completion time of a questionnaire) moderating a Likert scale: slower or faster respondents may answer with different precision (discrimination, Ferrando, 2009) or shifted thresholds, a DIF-like effect of time. ‘hgrm()’ is a shortcut: it writes the model in the syntax of [birt()] with ‘ordered = TRUE’ (stored in ‘fit\$model’), where every effect can also be set item by item, and fits it with ‘engine = "rtmb"’: maximum likelihood with per-person adaptive quadrature (marginal likelihood), or ELGM with ‘control = rtmb_control(method = "elgm)”’.

Usage

```
hgrm(
  model,
  data,
  moderators,
  a = "common",
  b = "none",
  impact = "mean",
  anchor = NULL,
  standardize = TRUE,
  control = rtmb_control(),
  dp = dpriors(),
  ...
)
```

Arguments

model	One-factor syntax, “f =~ y1 + y2 + ...”. Items are ordinal with integer categories; each item’s observed categories are recoded to 1..K.
data	A data frame.
moderators	Names of numeric person-level moderators (e.g. “logT”); several are allowed (e.g. a log time and its square, or piecewise terms). Standardized before fitting unless ‘standardize = FALSE’, so effects are per SD.
a, b	Moderation of discriminations (‘a’: alpha) and thresholds (‘b’: beta): “free” one effect per item (DIF), “common” one effect shared by all items (for ‘a’: a person-level precision effect as in Ferrando, 2009), or “none”. A vector of length ‘length(moderators)’ sets one option per moderator.
impact	Which latent moments depend on the moderators: any of “mean”, “var”, or “none”.

anchor	DIF-free anchor items: no threshold shifts and no item-specific ("free") discrimination effects. A common discrimination effect ('a = "common"') is a person-level precision effect and applies to every item.
standardize	Standardize the moderators.
control	RTMB settings, [rtmb_control()]; 'method = "elgm"' for the posterior.
dp	Priors of the ELGM fit, [dpriors()], as in [birt()].
...	Passed to [birt()] ('progress', 'label', 'itemtype', ...).

Details

$$P(y_{ij} > k) = \text{logit}^{-1}\{a_{ij}(\theta_j - b_{ik} - \delta_{ij})\}$$

$$\log a_{ij} = \log a_i + \sum_m \alpha_{im} z_{jm}, \quad \delta_{ij} = \sum_m \beta_{im} z_{jm}$$

$$\theta_j \sim N\{\sum_m \gamma_m z_{jm}, \exp(2 \sum_m \psi_m z_{jm})\}$$

****Identification.**** At $z = 0$ the trait is $N(0, 1)$. A common threshold shift of all items is the same as a shift of the trait mean, so 'impact = "mean"' cannot be combined with 'b = "common"' or 'b = "free"' unless anchors are given. Likewise a common change of all discriminations and the trait variance differ only through the scaling of the thresholds, so 'a = "common"' or '"free"' together with 'impact = "var"' is only weakly identified.

****DIF.**** Fit the model without threshold effects and test them with [score_test()] (one fit, every item), then free the flagged items' shifts with the others as anchors: 'b = "free", anchor = ...'. Spike-and-slab screening ('prior("ssp")') is written in the syntax of [birt()] with 'engine = "jags"'.

****Priors**** (ELGM only; maximum likelihood has none): those of [birt()] ([dpriors()]): discriminations hierarchical lognormal, first threshold ' $N(0, 2^2)$ ', increments lognormal ' $(-0.5, 1)$ ', 'gamma $\sim N(0, 1)$ ', 'psi $\sim N(0, 1)$ '.

Value

A [birt()] fit ('fit\$model' holds the syntax). Fits with moderation also have 'dif' (one row per moderated item parameter), 'impact' and 'precision' (by moderator tercile); see also [ordinal_information()] and 'plot(fit, type = "dif")'.

Examples

```
if (requireNamespace("RTMB", quietly = TRUE)) {
  d <- sim_hgrm(N = 400, J = 6, seed = 1)
  fit <- hgrm("theta =~ y1 + y2 + y3 + y4 + y5 + y6",
    data = d, moderators = "logT",
    a = "common", impact = "mean", control = rtmb_control(method = "aghq"),
    progress = FALSE
  )
  summary(fit)
  fit$model # the same model in birt() syntax
  score_test(fit, d$logT, by_item = TRUE) # DIF along the time, item by item
}
```

Description

Which parameters change with which person covariates? ‘ipcr()’ computes each person’s individual parameter contributions (IPCs), $IPC_i = \hat{\theta} + Bs_i$, where s_i are the person’s casewise scores ([estfun()][estfun.birt_rtmb]) and B the bread (the inverse of the average information, [bread()][estfun.birt_rtmb]); the IPCs average to the estimates. Each parameter’s IPCs are then regressed on the covariates: a slope that differs from zero says the parameter varies with that covariate (DIF of an item parameter, a covariate effect on a structural parameter). All covariates enter jointly, so each slope is adjusted for the others (Arnold, Oberski, Brandmaier & Voelkle, 2020; Arnold, Brandmaier & Voelkle, 2021).

Usage

```
ipcr(
  object,
  covariates,
  parm = NULL,
  se = c("HC3", "model"),
  impact_check = TRUE
)

## S3 method for class 'birt_ipcr'
print(x, digits = 3, all = FALSE, alpha = 0.05, ...)
```

Arguments

object	A maximum likelihood fit of ‘birt(..., engine = "rtmb")’ (‘control = rtmb_control(method = "aghq")’); ELGM fits are refused, as in [score_test()].
covariates	Person covariates: a data frame (numeric columns, factors, logical or character columns, which become factors) or a single vector, one row per person, without missing values. Only the covariates, e.g. ‘d[c("x1", "x2")]’: columns named like indicators of the model are refused.
parm	Parameters to regress: ‘NULL’ (all), “measurement”, “rt”, “structural”, a regular expression on the labels, or column indices, as in [score_test()].
se	“HC3” (default): heteroskedasticity-consistent standard errors of the slopes (the residual variance of the IPCs differs between persons by construction); “model”: the usual OLS standard errors, as ‘ipcr::ipcr()’ reports them.
impact_check	Message when a covariate correlates with the factor scores but is not a covariate of the model (impact), as in [score_test()]: a covariate related to a latent variable shifts every intercept of its indicators, so its slopes look like DIF on all of them (e.g. on every response time when it is related to speed). Add it to the latent regression (‘speed ~ x1’) and run ‘ipcr()’ on that fit to see DIF beyond impact.
x	A ‘birt_ipcr’ object.

<code>digits</code>	Number of decimals.
<code>all</code>	Print every slope; by default only those with <code>'p_holm < alpha'</code> (all slopes are in <code>'x\$coefficients'</code>).
<code>alpha</code>	Level of the Holm-adjusted p-values for the slopes printed by default.
<code>...</code>	Unused.

Details

IPCs are a one-step linearization around the pooled estimates, so the slopes are attenuated towards zero: use `'ipcr()'` to screen, then fit the moderated model (e.g. `'E(y1) ~ z'` or `'V(t1) ~ z'` in the syntax) for the size of an effect. The iterated version of the `'ipcr'` package is not implemented. The parameters are on the internal scale of the fit, as in `[estfun()]`: log of the positive loadings (labelled `'log(f == y)'`), log SDs, atanh of the correlation, the first threshold and log increments of ordinal items. Parameters at a bound (no standard error) are left out.

Value

An object of class `'birt_ipcr'`: `'coefficients'`, a tibble with one row per parameter and regression term (`'parameter'`, `'term'`, `'est'`, `'se'`, `'statistic'` (t), `'p'`, and `'p_holm'`, the Holm-adjusted p-value over all slopes); `'ipcs'`, the persons x parameters matrix of IPCs; `'se'`, `'n'`. `'print()'` shows the slopes with `'p_holm < 0.05'`, sorted by p-value, and how to read a slope on the internal scale (`'print(x, all = TRUE)'` shows all of them).

References

Arnold, M., Oberski, D. L., Brandmaier, A. M., & Voelkle, M. C. (2020). Identifying heterogeneity in dynamic panel models with individual parameter contribution regression. **Structural Equation Modeling*, 27*, 613-628. doi:10.1080/10705511.2019.1667240

Arnold, M., Brandmaier, A. M., & Voelkle, M. C. (2021). Predicting differences in model parameters with individual parameter contribution regression using the R package ipcr. **Psych*, 3*, 360-385. doi:10.3390/psych3030027

See Also

`[score_test()]` for a test of one moderator along all its values.

Examples

```
d <- sim_rtirt(N = 400, K = 6, seed = 1)
model <- rtirt_syntax(paste0("y", 1:6), paste0("t", 1:6))
fit <- birt(model, d, control = rtmb_control(method = "aghq"))
ipcr(fit, d[c("x1", "x2")])
```

jags_code

JAGS code of a model

Description

JAGS code of a model

Usage

```
jags_code(
  x,
  data = NULL,
  binary = NULL,
  family = NULL,
  dp = dpriors(),
  ssp_p = NULL,
  ordered = NULL,
  itemtype = "grm"
)
```

Arguments

x A fitted ‘birt’ object, or model syntax for [birt()].
data, binary, family, dp, ssp_p, ordered, itemtype
 As in [birt()] (when ‘x’ is syntax).

Value

The JAGS model code (character), printed with ‘cat()’.

ordinal_information

Test information of the ordinal indicators

Description

For fits with ordinal indicators (graded response or partial credit), the test information of those indicators as a function of the latent variable, at chosen values of the moderators.

Usage

```
ordinal_information(object, at = NULL, theta = seq(-3, 3, by = 0.1))
```

Arguments

object	A [birt()] fit with ordinal indicators (either engine).
at	Named list of moderator values (default: the 10th, 50th and 90th percentile of the first moderator, the others at their mean).
theta	Grid of values of the latent variable.

Value

Data frame with ‘theta’, the moderator values and the test information (point estimates of the item parameters), printed with 3 decimals (class ‘birt_tbl’, see [birt_print]).

plot.birt	<i>Plots of a JAGS fit</i>
-----------	----------------------------

Description

For fits with the default engine (‘engine = "rtmb"’) see [plot.birt_rtmb()].

Usage

```
## S3 method for class 'birt'
plot(
  x,
  type = c("loadings", "trace", "dif", "information"),
  pars = NULL,
  items = NULL,
  ...
)
```

Arguments

x	A ‘birt’ object.
type	“loadings”, “trace”, or for fits with ordinal indicators “dif” (expected item scores at low and high values of the first moderator, for the flagged items or ‘items’) and “information” (test information, see [ordinal_information()]).
pars	Regular expression of JAGS node names (“trace”).
items	Items for ‘type = "dif"’.
...	Passed to [graphics::plot()].

Details

‘type = "loadings"’: posterior means and 95 all free loadings), with spike-and-slab inclusion probabilities when present. “trace”: trace plots of the parameters with the largest R-hat (or those matching ‘pars’).

plot.birt_rtmb	<i>Plots of an RTMB fit</i>
----------------	-----------------------------

Description

Plots of a fit from [birt()] with the default engine ('engine = "rtmb"', ELGM or maximum likelihood). 'plot(fit)' draws the quantile effects when the model has effects that change with the quantile level p ('V()' statements or 'shash()' residuals); otherwise it draws the person scores ('type = "scores"'). For JAGS fits see [plot.birt()].

Usage

```
## S3 method for class 'birt_rtmb'
plot(
  x,
  type = c("quantiles", "scores", "dif", "information"),
  p = seq(0.05, 0.95, by = 0.05),
  targets = NULL,
  predictors = NULL,
  ald = NULL,
  max_panels = 16,
  items = NULL,
  ...
)
```

Arguments

x	A 'birt_rtmb' fit.
type	"quantiles": quantile effects against the level p (one panel per target and predictor whose effect can vary with p; the default for models with such effects); "scores": EAP scores of the latent variables (a histogram for one, a scatter plot for two; the default for other models); "information": test information about the first latent variable, from all indicators, the accuracy indicators (binary and ordinal) and the response times (see [cond_reliability()]); for fits with ordinal indicators only: the test information of those, by the first moderator, see [ordinal_information()]; "dif": ordinal indicators, the expected item scores at low and high values of the first moderator.
p	Quantile levels.
targets, predictors	Restrict the panels (names of latent variables or indicators / of predictors).
ald	Optional 'birt_quantile' fit ('engine = "jags"', [ald()]) at several p) whose coefficients are overlaid (points with 95% intervals).
max_panels	Maximum number of panels.
items	Items for 'type = "dif"'.
...	Passed to [graphics::plot()].

Value

Invisibly, the plotted values.

Examples

```
d <- sim_rtirt(N = 300, K = 6, seed = 1)
fit <- birt(rtirt_syntax(paste0("y", 1:6), paste0("t", 1:6)), d, progress = FALSE)
plot(fit) # no quantile effects that change with p: the scores of ability and speed
plot(fit, type = "information")
```

 ppc

Posterior predictive checks

Description

Replicates the data from posterior draws (person parameters included) and compares: item proportions correct, the sum-score distribution, means and SDs of the continuous indicators, and the conditional dependence between each binary item and its paired continuous indicator (correlation of accuracy with the standardized residual; Bolsinova & Tijmstra, 2016). A continuous indicator is paired with a binary item when its name ends with the item name (e.g. 'lrt_ME1' and 'ME1'), or by position when the numbers of binary and continuous indicators are equal.

Usage

```
ppc(object, ...)

## S3 method for class 'birt'
ppc(object, n_draws = 500, seed = 1, ...)
```

Arguments

object	A 'birt' object.
...	Unused.
n_draws	Number of posterior draws.
seed	Random seed.

Value

An object of class 'birt_ppc' (list of tables); PPP values outside (.025, .975) are flagged.

quantile_information *Information of each quantile regression of the response times*

Description

For a model fitted with ‘engine = "rtmb"’: how much the τ -th quantile regression of the response times on the latent variables (e.g. the cross relation ‘t ~ speed + ability’, with cross-loadings ‘ability ~ t1’ and scale or skew moderation ‘V(t1) ~ ability’) says about the first latent variable, and which τ says the most. Speed is a nuisance and is projected out, as in [cond_reliability()].

Usage

```
quantile_information(
  object,
  p = seq(0.05, 0.95, by = 0.05),
  theta = c(-2, 0, 2),
  at = NULL
)

## S3 method for class 'birt_qinfo'
print(x, ...)

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

Arguments

object	A fit of ‘birt(..., engine = "rtmb")’ with continuous indicators.
p	Quantile levels τ .
theta	Values of the first latent variable.
at	Covariate and moderator values (default: their means), as in [cond_reliability()].
x	A ‘birt_qinfo’ object.
...	Unused.

Details

The τ -th regression of item i is estimated by the check loss, whose estimating function is $\psi_\tau = \tau - 1\{t_{ij} \leq q_{i\tau}\}$. Its information about the latent variable (Godambe, 1960) is

$$I_{i\tau}(\theta) = \frac{f_i(q_{i\tau})^2 \beta_i(\tau)^2}{\tau(1-\tau)}, \quad \beta_i(\tau) = \partial q_{i\tau} / \partial \theta,$$

a signal (how far the quantile moves with θ) times the precision of the quantile, the inverse of its asymptotic variance $\tau(1-\tau)/f(q_\tau)^2$ (Koenker, 2005). It equals the Fisher information of the indicator "faster than $q_{i\tau}$ ". The items are summed (with the speed direction, item by item rank one) and speed is projected out.

The values for different τ are not additive: each is the information of one regression on its own, and at most the information of the whole response-time distribution ('info_rt', as in [cond_reliability()]); 'efficiency' is the ratio. For one item with a normal residual and a location effect (a cross-loading) the median regression is best and keeps $2/\pi = .64$ of the item's information; for a scale effect ('V(t) ~ ability') the median carries nothing and the best single quantiles are $\tau = .058$ and $.942$, which keep $.30$ (maximizing $z_\tau^2 \phi(z_\tau)^2 / (\tau(1 - \tau))$); on the choice of sample quantiles, Ogawa, 1951). With several items and the projection of speed, the efficiencies of the curve differ somewhat from these item values.

The quantities are evaluated at the parameter estimates (ML or ELGM), at each 'theta' with speed at its conditional median and the covariates at 'at'; the effects are conditional on θ , not the average marginal effects of [quantiles()]. The response-time distribution is that of the fitted model without censoring.

Value

A list of class 'birt_qinfo': * 'curve': 'theta', 'p', 'info' (all items, speed projected out), 'info_rt' (the whole RT distribution), 'efficiency'; * 'items': 'theta', 'item', 'p', 'q' (the quantile, model scale), 'beta' ($\partial q / \partial \theta$), 'density' ($f(q)$), 'info' (the item's $I_{i\tau}$, speed held fixed); * 'best': for each 'theta', the τ of 'p' with the largest 'info'.

References

- Godambe, V. P. (1960). An optimum property of regular maximum likelihood estimation. *The Annals of Mathematical Statistics, 31*, 1208-1211. doi:10.1214/aoms/1177705693
- Koenker, R. (2005). *Quantile regression*. Cambridge University Press. doi:10.1017/CBO9780511754098
- Ogawa, J. (1951). Contributions to the theory of systematic statistics, I. *Osaka Mathematical Journal, 3*, 175-213.

Examples

```
d <- sim_rtirt(N = 500, K = 6, seed = 1)
m <- paste("theta =~ y1 + y2 + y3 + y4 + y5 + y6",
  "speed =~ -1*t1 + -1*t2 + -1*t3 + -1*t4 + -1*t5 + -1*t6",
  "V(t1 + t2 + t3 + t4 + t5 + t6) ~ theta",
  sep = "\n"
)
fit <- birt(m, d, progress = FALSE)
qi <- quantile_information(fit)
qi
plot(qi)
```

Description

For a maximum likelihood fit of `[birt()]` with `'engine = "rtmb"'`: the casewise influence function of each quantile effect of `[quantiles()]` (e.g. the cross-relation `'t ~ speed + ability'` of a log response time at $p = 0.9$). A quantile effect is a smooth function $\beta(p) = g_p(\vartheta)$ of the parameters, so to first order

$$\hat{\beta}(p) - \beta(p) \approx \frac{1}{N} \sum_j \text{IF}_j, \quad \text{IF}_j = N \nabla g_p^\top \hat{V} s_j,$$

with s_j the person's score (`[estfun.birt_rtmb()]`) and $\hat{V}$ the inverse observed information: the influence function of a function of an M-estimator (Hampel, 1974), obtained by the delta method. It gives * robust standard errors, $\{\sum_j \text{IF}_j^2\}^{1/2}/N$ (the sandwich of Huber, 1967, and White, 1982, valid when the SHASH or normal distribution is misspecified), next to the model-based ones of `[quantiles()]`; * the influence of each person on an effect: removing person j changes the estimate by about $-\text{IF}_j/N$ (as in Cook, 1977); * score-based tests of the invariance of an effect along a moderator, `'score_test(qs, z)'` (Zeileis & Hornik, 2007; Merkle & Zeileis, 2013).

Usage

```
quantile_score(
  object,
  p = object$control$p,
  type = c("conditional", "unconditional")
)

## S3 method for class 'birt_qscore'
print(x, ...)

## S3 method for class 'birt_qscore'
score_test(
  object,
  moderator,
  test = c("auto", "DM", "CvM", "maxLM", "LM"),
  ...
)
```

Arguments

<code>object</code>	A maximum likelihood fit of <code>'birt(..., engine = "rtmb")'</code> (<code>'rtmb_control(method = "aghq")'</code>).
<code>p</code>	Quantile levels.
<code>type</code>	<code>"conditional"</code> (covariates fixed) or <code>"unconditional"</code> .
<code>x</code>	A <code>'birt_qscore'</code> object.
<code>...</code>	Unused.
<code>moderator</code>	One value per person: numeric, or a factor (logical and character are converted).
<code>test</code>	<code>"auto"</code> (<code>"DM"</code> for a numeric moderator, <code>"LM"</code> for a factor), <code>"DM"</code> , <code>"CvM"</code> , <code>"maxLM"</code> or <code>"LM"</code> , as in <code>[score_test()]</code> .

Details

****Conditional or unconditional.**** The effects are averages over the persons' covariates (average marginal effects). 'type = "conditional"' treats the covariates as fixed, as [quantiles()] does. 'type = "unconditional"' also counts their sampling: IF_j gains $h_j - \hat{\beta}(p)$, the deviation of person j 's own marginal effect from the average (Graubard & Korn, 1999). The two differ only when the effect depends on covariates.

****ELGM fits.**** 'quantile_score()' is a maximum likelihood tool and refuses an ELGM fit (the default of [birt()]): refit the same model with 'control = rtmb_control(method = "aghq")'. Influence functions and score tests need case-wise scores that sum to zero at the estimate; at the ELGM posterior mode they sum to minus the gradient of the log prior, the reported ELGM estimates are posterior means rather than the mode, and MAP versions of the score tests need centred scores, with error rates that depend on the covariance estimate (Debelak, Pawel, Strobl & Merkle, 2022). The Bayesian counterparts: [quantiles()] of the ELGM fit gives posterior means, SDs and intervals of the quantile effects, [quantile_test()] works on ELGM fits, and the invariance of an effect along a numeric 'z' is assessed by fitting the ELGM model with and without the moderation (e.g. 'V(t1) ~ z') and comparing the marginal likelihoods with [compare()] (a Bayes factor, which depends on the prior of the effect).

'score_test(qs, moderator)' orders each effect's influence function by the moderator and compares its cumulative sum, standardized by its variance, with a Brownian bridge. Under parameter invariance the partial sums of any fixed linear combination of the scores converge to a Brownian bridge (Zeileis & Hornik, 2007), so the test is valid; it detects changes of the parameters in the direction that moves the effect (to first order), one effect at a time, with Holm-adjusted p-values. It needs 'type = "conditional"': the covariate term of the unconditional influence function varies with the covariates by construction.

Value

A persons x effects matrix of influence functions (columns 'target|predictor|p') of class 'birt_qscore', with attribute 'effects': 'est', 'se' (model-based), 'se.robust' (sandwich).

References

- Debelak, R., Pawel, S., Strobl, C., & Merkle, E. C. (2022). Score-based measurement invariance checks for Bayesian maximum-a-posteriori estimates in item response theory. *British Journal of Mathematical and Statistical Psychology*, 75*(3), 728-752. doi:10.1111/bmsp.12275
- Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69*, 383-393.
- Huber, P. J. (1967). The behavior of maximum likelihood estimates under nonstandard conditions. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1*, 221-233.
- White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, 50*, 1-25.
- Cook, R. D. (1977). Detection of influential observation in linear regression. *Technometrics*, 19*, 15-18.
- Graubard, B. I., & Korn, E. L. (1999). Predictive margins with survey data. *Biometrics*, 55*, 652-659. doi:10.1111/j.0006341X.1999.00652.x

Zeileis, A., & Hornik, K. (2007). Generalized M-fluctuation tests for parameter instability. **Statistica Neerlandica*, 61*, 488-508. doi:10.1111/j.14679574.2007.00371.x

quantile_test

Does a quantile effect change with the quantile level?

Description

For a ‘birt_rtmb’ fit: for each target and predictor of [quantiles()] (e.g. the cross-relation ‘t ~ speed + ability’ of a log response time, or ‘speed ~ ability + x’), a test of whether the effect changes with the quantile level p, and the difference of each level from the median.

Usage

```
quantile_test(object, p = object$control$p, ...)
```

```
## S3 method for class 'birt_qtest'
print(x, ...)
```

Arguments

object	A ‘birt_rtmb’ fit.
p	Quantile levels (at least two).
...	Unused.
x	A ‘birt_qtest’ object.

Details

In the model the location of the target is linear in the predictor and its scale and skewness may depend on it, so the p-quantile effect changes with p exactly when the predictor moderates the scale ($V(t) \sim k'$) or the skewness of the target; equal effects at every level is the null hypothesis of no heteroscedasticity, as in the test of equal slopes across regression quantiles (Koenker & Bassett, 1982). The test is therefore the Wald test of these moderation parameters (df = their number). A test of the differences of the effects across p would be singular at this null (the effects of the shape parameters vanish when the moderation is 0). Curves that change with p without such a parameter (e.g. a SHASH latent target correlated with the predictor) are tested by their differences from the median (noted); curves constant by construction are reported as such.

Value

A list of class ‘birt_qtest’: ‘test’ (per curve: the tested ‘parameters’, ‘chisq’, ‘df’, ‘p.value’) and ‘differences’ (per curve and level: ‘diff’ = $\beta(p) - \beta(\text{median})$, ‘se’, ‘z’, ‘p.value’). With ELGM the statistic uses the posterior mean and covariance (a normal approximation).

References

Koenker, R., & Bassett, G. (1982). Robust tests for heteroscedasticity based on regression quantiles. **Econometrica*, 50*(1), 43-61. doi:10.2307/1912528

quantiles

*Quantile effects of an RTMB fit***Description**

For a model fitted with `'engine = "rtmb"'`: the effect of each predictor on the conditional p-quantile of the second latent variable (e.g. `'speed ~ theta + x'`) and of each continuous indicator (e.g. the cross-relation of `'theta'` with a log response time), averaged over the model-implied distribution of the moderators (average marginal effects), with delta-method standard errors (with `'method = "elgm"'`: posterior means, SDs and 95% credible intervals). With normal residuals and no moderators the effects are the same at every level (the mean-model coefficients); `'shash()'` and `'scale' / 'skew'` moderators let them change with p. These are the counterparts of the coefficients of `'ald(p)'` fits (engine `"jags"`), from one fit for all p.

Usage

```
quantiles(object, ...)

## S3 method for class 'birt_rtmb'
quantiles(object, p = object$control$p, ...)
```

Arguments

<code>object</code>	A <code>'birt_rtmb'</code> fit.
<code>...</code>	Unused.
<code>p</code>	Quantile levels.

Details

`'band.lower'` / `'band.upper'` form a simultaneous 95% band over the levels `'p'` of each target-predictor curve (sup-t band; Montiel Olea & Plagborg-Moller, 2019): every effect of the curve lies in its band with probability 0.95, so the levels where the band excludes 0 can be read off together. The critical value is the 0.95 quantile of $\max |Z|$ for Z normal with the correlation of the curve's estimates (the joint delta-method covariance; with ELGM, the posterior covariance). `[quantile_test()]` tests whether a curve changes with p.

Value

A data frame with `'target'`, `'predictor'`, `'p'`, `'est'`, `'se'`, `'ci.lower'`, `'ci.upper'` (pointwise), `'band.lower'`, `'band.upper'` (simultaneous over `'p'`); attribute `'vcov'`: the joint covariance of `'est'`. It prints with 3 decimals (class `'birt_tbl'`, see `[birt_print]`).

References

Montiel Olea, J. L., & Plagborg-Moller, M. (2019). Simultaneous confidence bands: Theory, implementation, and an application to SVARs. *Journal of Applied Econometrics*, 34*(1), 1-17.
[doi:10.1002/jae.2656](https://doi.org/10.1002/jae.2656)

rt_information	<i>Where response times carry information about the first latent variable</i>
----------------	---

Description

For a model fitted with ‘engine = "rtmb"’: how the information that the response times carry about the first latent variable (e.g. ability, through cross-loadings ‘ability $\sim$ t1’ and scale or skew moderation ‘V(t1) $\sim$ ability’) is spread over the quantile levels of the response-time distribution, and how much of it, and of the reliability, a time limit keeps. Speed is a nuisance and is projected out, as in [cond_reliability()].

Usage

```
rt_information(
  object,
  theta = c(-2, 0, 2),
  bands = c(0, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 1),
  limits = list(slow = c(0.75, 0.9, 0.95, 0.99), fast = c(0.01, 0.05, 0.1)),
  at = NULL,
  k = 60
)

## S3 method for class 'birt_rt_info'
print(x, ...)

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

Arguments

object	A fit of ‘birt(..., engine = "rtmb")’ with continuous indicators.
theta	Values of the first latent variable.
bands	Breaks of the quantile bands.
limits	Quantile levels of the limits; ‘slow’ for time limits, ‘fast’ for fast responses.
at	Covariate and moderator values (default: their means), as in [cond_reliability()].
k	Gauss-Legendre nodes per interval.
x	A ‘birt_rt_info’ object.
...	Unused.

Details

* ‘bands’: the information of the response times in each band of quantile levels (all items at their own quantiles). The bands partition the RT information of [cond_reliability()] (‘info_rt’); they show where it lies, not what a design would keep. * ‘limits’: the information and reliability when every response time beyond a limit is censored (only known to exceed it): ‘side = "slow"’, a time limit

at the item's quantile u (as with a time limit); 'side = "fast"', responses faster than the quantile u known only as fast (as when rapid responses are not modelled). Censoring keeps the information that a response was beyond the limit, so a slow tail can hold much of the partition yet lose little under a time limit. Discarding the responses instead (truncation) loses more and biases a model that ignores it.

The decomposition of Fisher information over an ordered sample follows Zheng & Gastwirth (2000); censored information Efron & Johnstone (1990) and Escobar & Meeker (1998). The curves describe the fitted model: fast responses that are rapid guesses (Wise & Kong, 2005) violate it, and their apparent information is then an artefact. The response-time distribution is that of the fitted model without censoring; for a fit with 'censor =', [cond_reliability()] gives the information under the fit's own limits. Binary and ordinal indicators enter 'rel' as in [cond_reliability()], and moderated loadings or shifts are evaluated at the moderator values 'at'.

Value

A list of class 'birt_rt_info': 'density' ('theta', 'u', 'info': the information density over u), 'bands' ('theta', 'band', 'info', 'share'), 'limits' ('theta', 'side', 'u', 'info_rt', 'kept' (share of the RT information), 'rel' (reliability with the accuracy items), 'rel_full'), and 'check' (the RT information of the partition and of [cond_reliability()]).

References

- Zheng, G., & Gastwirth, J. L. (2000). Where is the Fisher information in an ordered sample? **Statistica Sinica*, 10*, 1267-1280.
- Efron, B., & Johnstone, I. M. (1990). Fisher's information in terms of the hazard rate. **The Annals of Statistics*, 18*, 38-62. doi:10.1214/aos/1176347492
- Escobar, L. A., & Meeker, W. Q. (1998). Fisher information matrices with censoring, truncation, and explanatory variables. **Statistica Sinica*, 8*, 221-237.
- Wise, S. L., & Kong, X. (2005). Response time effort: A new measure of examinee motivation in computer-based tests. **Applied Measurement in Education*, 18*, 163-183. doi:10.1207/s15324818ame1802_2

rtirt_syntax

Model syntax for joint response accuracy and response time models

Description

Writes the lavaan-style syntax of the usual RT-IRT models from the item and time column names, to pass to [birt()] or to edit.

Usage

```
rtirt_syntax(
  items,
  times,
  cross = c("none", "free", "ssp"),
```

```

    structure = NULL,
    cov = NULL,
    cov_on = c("speed", "ability", "both"),
    speed_var = c("free", "fixed"),
    ability = "ability",
    speed = "speed"
  )

```

Arguments

items	Names of the binary item columns.
times	Names of the (log) response time columns, in the order of ‘items’.
cross	Cross-loadings of ability on the response times: “none”, “free”, or “ssp” (spike-and-slab priors; engine “jags”).
structure	Relation of ability and speed: “cor” (correlated, van der Linden, 2007), “path” (‘speed ~ ability’), or “none” (independent). The default is “cor” without cross-loadings and “none” with them (a common cross-loading is the same as the correlation or the path, so they are not identified together).
cov	Covariates of the latent regression(s).
cov_on	Which latent variables are regressed on ‘cov’: “speed”, “ability” or “both”.
speed_var	“free” (default; the speed loadings are fixed at -1, so the speed variance is estimated) or “fixed” (‘speed ~ 1*speed’, as in the dissertation’s Chapter 5; usually too restrictive for log seconds).
ability, speed	Names of the latent variables.

Value

The syntax (a character string of class ‘birt_syntax’, printed with ‘cat’).

Examples

```

items <- paste0("y", 1:5)
times <- paste0("t", 1:5)
rtirt_syntax(items, times) # van der Linden (2007)
rtirt_syntax(items, times, cross = "ssp") # cross-loadings, ssp priors
rtirt_syntax(items, times, structure = "path", cov = c("x1", "x2"))

```

rtmb_code

RTMB code of a model

Description

The counterpart of [jags_code()] for ‘engine = “rtmb”’: writes a standalone R script with the model written out item by item (the joint log density of one person given the standard-normal latent values), the Laplace fit and the AGHQ refinement. It expects the data frame as ‘data’; parameter names and starting values are those of [birt()], so ‘fit\$rtmb\$par’ can be plugged in; from a fit, the script starts at its estimates. As in birt, the SHASH shapes are bounded and the full model starts from the normal one (birt also halves the step when an AGHQ round does not improve).

Usage

```
rtmb_code(
  x,
  data = NULL,
  binary = NULL,
  family = NULL,
  log_rt = FALSE,
  ordered = NULL,
  itemtype = "grm",
  censor = NULL,
  estimate = TRUE,
  k = NULL,
  file = NULL
)
```

Arguments

x A ‘birt_rtmb’ fit, or model syntax.

data, binary, family, log_rt, ordered, itemtype, censor As in [birt()] (when ‘x’ is syntax).

estimate Include the fitting part (Laplace and AGHQ); ‘FALSE’ gives the model only.

k AGHQ nodes per latent dimension (default as in [rtmb_control()]).

file Optional file to write the script to.

Value

The script (character), printed with ‘cat()’.

Examples

```
if (requireNamespace("RTMB", quietly = TRUE)) {
  d <- sim_rtirt(K = 5, seed = 1)
  f <- birt(rtirt_syntax(paste0("y", 1:5), paste0("t", 1:5), cross = "free"), d,
    control = rtmb_control(method = "aghq"), progress = FALSE
  )
  path <- file.path(tempdir(), "model_rtmb.R")
  rtmb_code(f, file = path) # then: data <- d; source(path)
}
```

rtmb_control

Control settings of the RTMB engine

Description

Control settings of the RTMB engine

Usage

```
rtmb_control(
  method = c("elgm", "aghq", "laplace"),
  k = NULL,
  max_rounds = 8,
  tol = 0.001,
  p = c(0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95),
  trace = FALSE,
  inner = c("aghq", "laplace"),
  hyper = "auto",
  k_hyper = NULL,
  cond_modes = TRUE
)
```

Arguments

method	"elgm" (default): approximate Bayesian inference for the model as an extended latent Gaussian model (Stringer, Brown & Stafford, 2023) with the priors of [dpriors()] and 'prior()'. The latent Gaussian field W holds the person latents only; they are integrated out per person ('inner'), and the posterior of the remaining parameters is integrated by an outer adaptive quadrature: 'k_hyper' nodes per dimension on the hyperparameters 'hyper', a Gaussian (Laplace) treatment of the other directions. See *Details*. "aghq": maximum likelihood with adaptive Gauss-Hermite quadrature over the person latents (as GLMMadaptive or mirt; needed by [score_test()], [quantile_score()], [dif_tree()]), also written "ml" (or "ML", "mle"); "laplace": maximum likelihood, Laplace.
k	Quadrature nodes per latent dimension (default 21 for one latent variable, 9 for two).
max_rounds, tol	AGHQ: recompute the quadrature centres at most 'max_rounds' times, until -2 log L changes by less than 'tol'.
p	Default quantile levels of [quantiles()].
trace	Print the progress of the optimization.
inner	'method = "elgm"': the integral over the person latents, "aghq" (default; the field is separable by person, so per-person AGHQ is cheap and more accurate) or "laplace" (the Laplace approximation of the original ELGM).
hyper	'method = "elgm"': parameters with quadrature in the outer layer. "auto" (default): the SHASH shape (and scale / skew moderators) of the second latent variable when it is 'shash()', otherwise 'c("cor", "path", "sd")'. Keywords "cor" (latent correlation), "path" (latent regression path), "sd" (free latent SDs), "shape", "cross" (cross-loadings), "resid" (residual log SDs), or names of internal parameters (names(fit\$rtmb\$x)). 'character()' treats every direction as Gaussian (posterior mode + Laplace). "auto" uses at most 3 directions (the SHASH skewness and tail weight first, then skew and scale moderators, then correlation, path and SDs) and says which ones it treats as Gaussian: with 4 directions the $5^4 = 625$ nodes took 5 times the outer time of 3 directions for posterior means within 0.034 SD and SDs within 2.1% (a SHASH speed with

	skew and scale moderators, N = 1000, 20 items); name them in ‘hyper’ for quadrature on all.
k_hyper	Outer nodes per hyperparameter (default 11, 9, 5, 5, 3, 3 for 1 to 6 hyperparameters). Each quadrature direction is scaled separately on both sides from the drop of the log posterior at +-2 SD (as in INLA’s grid), so skewed posteriors are covered.
cond_modes	‘method = "elgm"’: at each outer node, move the Gaussian directions to their conditional posterior mode (default ‘TRUE’; Laplace over them). ‘FALSE’ places them on the Gaussian conditional-mean line of the posterior mode, which is faster but thins the tails when the conditional modes move nonlinearly (e.g. SHASH skewness).

Details

****ELGM.**** With ‘method = "elgm"’ the fit is the posterior mode of $\log p(y \mid \theta) + \log p(\theta)$, with $\log p(y \mid \theta)$ from per-person AGHQ (or Laplace), and the outer posterior $p(\theta \mid y)$ is integrated on a mixed grid: the ‘hyper’ directions come first in the Cholesky factor of the inverse Hessian, the other parameters sit at their conditional posterior modes (‘cond_modes’) and contribute their conditional (Laplace) variance. Posterior means, SDs and 95% intervals (from the quadrature moments, on the log scale for SDs and the Fisher-z scale for correlations) are reported for every parameter and for [quantiles()]; EAP scores are mixtures over the outer nodes; ‘fit_indices()’ gives the log marginal likelihood (for Bayes factors in [compare()]). Item parameters are deliberately kept out of W: with items in W the inner mode is a penalized joint maximum likelihood (incidental parameter bias).

Value

A list used by [birt()] (‘engine = "rtmb"’).

References

Stringer, A., Brown, P., & Stafford, J. (2023). Fast, scalable approximations to posterior distributions in extended latent Gaussian models. *Journal of Computational and Graphical Statistics*, 32*, 84-98.

score_test	<i>Score-based tests of parameter invariance</i>
------------	--

Description

Tests whether item or structural parameters change along a person-level moderator (DIF, measurement invariance, item position effects) from a single maximum likelihood fit (‘birt(..., engine = "rtmb"’), also with ‘ordered = TRUE’): the casewise scores ([estfun.birt_rtmb()]) are ordered by the moderator, cumulated after decorrelation and compared with a Brownian bridge (Merkle & Zeileis, 2013).

Usage

```
score_test(object, ...)

## S3 method for class 'birt_rtmb'
score_test(
  object,
  moderator,
  parm = "measurement",
  by_item = FALSE,
  test = c("auto", "DM", "CvM", "maxLM", "LM"),
  decorrelate = c("block", "full"),
  cluster = NULL,
  impact_check = TRUE,
  ...
)
```

Arguments

object	A maximum likelihood fit of 'birt(..., engine = "rtmb")' ('rtmb_control(method = "aghq")').
...	Passed to the method.
moderator	One value per person: numeric (continuous or ordered) or a factor (logical and character are turned into factors).
parm	Parameters to test: "measurement" (loadings and intercepts or thresholds of the binary and ordinal items, plus their moderation effects; default), "rt" (parameters of the continuous indicators, including cross-loadings on them), "structural", a regular expression on the parameter labels, or column indices.
by_item	'TRUE': also test the parameters of each indicator separately, with Holm-adjusted p-values.
test	"auto" ("DM" for numeric moderators, "LM" for factors), "DM" (double maximum), "CvM" (Cramér-von Mises), "maxLM" (supremum LM, 10% trimming) or "LM".
decorrelate	"block" (within the tested parameters) or "full" (all parameters, then the tested components; not valid for moderators built from the response times).
cluster	Optional cluster identifiers (see Details).
impact_check	Message when the moderator correlates with the factor scores but is not a covariate of the model.

Details

****Moderators derived from the response times.**** The marginal distribution of the continuous indicators (response times) does not involve the parameters of the binary or ordinal items, so in a joint model their scores have mean zero given any function of the times (for example the total log time): the measurement parameters can be tested along such a moderator, provided the scores are decorrelated within the tested block ('decorrelate = "block", the default). The parameters of the continuous indicators cannot be tested along a moderator built from them.

****DIF and impact.**** A moderator related to the latent variable (impact) shifts all intercepts and is flagged as instability. To test DIF beyond impact, add the moderator to the latent regression (`'f ~ z'` in the syntax, or `'E(f) ~ z'` for ordinal items) and test that fit; `'score_test()'` gives a message when the moderator correlates with the factor scores and is not a covariate of the model.

****Clustered data.**** With `'cluster'`, the scores are summed within clusters, which is exact for a moderator that is constant within clusters; a person-level moderator in clustered data is refused.

****ELGM fits.**** `'score_test()'` is a maximum likelihood tool and refuses an ELGM fit (the default of `[birt()]`): refit the same model with `'control = rtmb_control(method = "aghq")'`. The tests need case-wise scores that sum to zero at the estimate; at the ELGM posterior mode they sum to minus the gradient of the log prior, the reported ELGM estimates are posterior means rather than the mode, and MAP versions of the score tests need centred scores, with error rates that depend on the covariance estimate (Debelak, Pawel, Strobl & Merkle, 2022). The Bayesian route for one effect is to fit the ELGM model with and without it (e.g. `'E(y1) ~ z'` for uniform DIF of `'y1'` along a numeric `'z'`) and compare the marginal likelihoods with `[compare()]` (a Bayes factor, which depends on the prior of the effect, `'dpriors(beta = )'`).

`birtRcpp` has a function of the same name for its models; with both packages attached, the one attached last is found first, and `'birt::score_test()'` dispatches to `birtRcpp` for its models.

Value

A data frame of class `'birt_score_test'`: `'parameters'`, `'n'`, `'statistic'`, `'p.value'` and, with `'by_item'`, `'p.holm'`.

References

- Merkle, E. C., & Zeileis, A. (2013). Tests of measurement invariance without subgroups: A generalization of classical methods. *Psychometrika*, 78*(1), 59-82. doi:10.1007/s1133601293024
- Debelak, R., Pawel, S., Strobl, C., & Merkle, E. C. (2022). Score-based measurement invariance checks for Bayesian maximum-a-posteriori estimates in item response theory. *British Journal of Mathematical and Statistical Psychology*, 75*(3), 728-752. doi:10.1111/bmsp.12275

Examples

```
d <- sim_rtirt(N = 400, K = 6, seed = 1)
f <- birt(rtirt_syntax(paste0("y", 1:6), paste0("t", 1:6)), d,
  control = rtmb_control(method = "aghq"), progress = FALSE
)
score_test(f, d$x1 > 0, by_item = TRUE) # a moderator unrelated to the items
score_test(f, rowSums(d[paste0("t", 1:6)])) # along the total log time
```

Description

For `birt(..., engine = "rtmb")`: replaces the normal residual of the second latent variable (e.g. `'speed'` in `'speed ~ theta + x'`) or of continuous indicators (e.g. log response times, `'family = list(continuous = shash(skew = "theta"))'`) by the sinh-arcsinh distribution of Jones and Pewsey (2009): `'e = sinh((asinh(z) + eps) / delta)'` with `'z ~ N(0, 1)'`, where `'eps'` sets the skewness (`'eps > 0'`: right-skewed) and `'delta'` the tail weight (`'delta < 1'`: heavier tails than the normal). Moderators make the skewness (`'skew'`) linear in latent variables or covariates, and `'V(y) ~ z'` in the model syntax the log scale, so that the predictor effects on the conditional quantiles can change with the level `p`; see `[quantiles()]`. This is the likelihood-based counterpart of fitting `[ald()]` at several quantiles with `'engine = "jags"'`: one fit gives every quantile, and the quantile curves cannot cross.

Usage

`shash(skew = NULL)`

Arguments

`skew` Moderators of the skewness `'eps'`.

Value

An object of class `'birt_family'`.

Examples

`shash(skew = "theta")`

sim_hgrm	<i>Simulate Likert responses moderated by a total response time</i>
----------	---

Description

GRM with `'J'` items and 4 categories; the log total time `'logT'` lowers the discrimination of every item (person precision, `'alpha'`), shifts the thresholds of items 1 and 2 (`'beta'`), and is related to the trait mean (`'gamma'`).

Usage

`sim_hgrm(N = 800, J = 10, alpha = 0.25, beta = 0.4, gamma = 0.2, seed = NULL)`

Arguments

<code>N, J</code>	Persons, items.
<code>alpha</code>	Common effect of standardized <code>'logT'</code> on log discrimination.
<code>beta</code>	Threshold shift of items 1 and 2 per SD of <code>'logT'</code> .
<code>gamma</code>	Effect of <code>'logT'</code> on the trait mean.
<code>seed</code>	Random seed for these data; <code>'NULL'</code> (default) draws from the current random number stream, as in <code>[sim_rtirt()]</code> .

Value

A data frame with 'y1..yJ' (1..4), 'Time' and 'logT'.

sim_rtirt	<i>Simulate joint response accuracy and response time data</i>
-----------	--

Description

2PL accuracy and lognormal response times with item-specific cross-loadings of ability on log response time, optionally an effect of ability on the log residual SD of the log response time (scale), and speed depending on two covariates.

Usage

```
sim_rtirt(
  N = 500,
  K = 10,
  rho = rep(c(0, 0.3), length.out = K),
  scale = 0,
  sd_speed = 0.3,
  beta = c(0.1, 0),
  seed = NULL
)
```

Arguments

N	Persons.
K	Items.
rho	Cross-loadings of ability on log RT (length 1 or K).
scale	Effects of ability on the log residual SD of log RT (length 1 or K; 0: none, as in 'V(t) ~ ability').
sd_speed	SD of the speed residual.
beta	Effects of covariates 'x1', 'x2' on speed.
seed	Random seed for these data (the random number stream of the session is left as it was); 'NULL' (default) draws from the current stream, as 'rnorm()' does, so that 'set.seed()' before the call, or repeated calls in a simulation, give new data each time.

Value

A data frame with 'y1..yK' (0/1), 't1..tK' (log RT), 'x1', 'x2'.

subset_chains	<i>Keep a subset of the MCMC chains</i>
---------------	---

Description

Recomputes the parameter table, scores, reliability and item tables from the kept chains, e.g. after a chain was found in a separate posterior mode (with free cross-loadings, ability can rotate into a second speed factor in one chain). The marginal-likelihood criteria are dropped; recompute them with `[add_fit_indices()]`.

Usage

```
subset_chains(object, chains)
```

Arguments

object	A 'birt' fit.
chains	Indices of the chains to keep.

Value

The fit with the reduced draws.

Examples

```
if (requireNamespace("rjags", quietly = TRUE)) {
  fit <- birt("ability =~ y1 + y2 + y3 + y4 + y5", sim_rtirt(N = 200, K = 5, seed = 1),
    engine = "jags", n_chains = 3, n_iter = 1500, n_burn = 500, fit_indices = FALSE,
    progress = FALSE
  )
  # e.g. chain 3 in another mode: compare per-chain means, then drop it
  sapply(1:3, function(ch) mean(fit$draws[[ch]][, 1]))
  fit <- subset_chains(fit, c(1, 2))
}
```

Index

- * **DIF**
 - ipcr, 29
 - score_test, 46
- * **measurement invariance**
 - ipcr, 29
 - score_test, 46
- * **response time models**
 - ipcr, 29
 - score_test, 46
- add_fit_indices, 3
- ald, 4
- algorithm, 5
- as.character.birt_equations (equations), 25
- as.data.frame.birt_summary (birt_summary), 15
- as.data.frame.birt_values (birt_print), 12
- birt, 6
- birt-class, 11
- birt_print, 12
- birt_quantile, 13
- birt_rtmb, 14
- birt_summary, 15
- bread.birt_rtmb (estfun.birt_rtmb), 26
- check_sbc, 16
- coef.birt (birt-class), 11
- coef.birt_quantile (birt_quantile), 13
- coef.birt_rtmb (birt_rtmb), 14
- compare, 18
- cond_reliability, 19
- convergence, 20
- convergence.birt_rtmb (birt_rtmb), 14
- dif_tree, 21
- dpriors, 23
- equations, 25
- estfun.birt_rtmb, 26
- estimates (birt-class), 11
- estimates.birt_quantile (birt_quantile), 13
- estimates.birt_rtmb (birt_rtmb), 14
- fit_indices (birt-class), 11
- fit_indices.birt_quantile (birt_quantile), 13
- fit_indices.birt_rtmb (birt_rtmb), 14
- format.birt_tbl (birt_print), 12
- format.birt_values (birt_print), 12
- hgrm, 27
- ipcr, 29
- jags_code, 31
- logLik.birt_rtmb (birt_rtmb), 14
- normal (ald), 4
- ordinal_information, 31
- plot.birt, 32
- plot.birt_cond_rel (cond_reliability), 19
- plot.birt_qinfo (quantile_information), 35
- plot.birt_quantile (birt_quantile), 13
- plot.birt_rt_info (rt_information), 41
- plot.birt_rtmb, 33
- plot.birt_sbc (check_sbc), 16
- pointwise_loglik (birt-class), 11
- posterior_draws (birt-class), 11
- ppc, 34
- print.birt (birt-class), 11
- print.birt_equations (equations), 25
- print.birt_ipcr (ipcr), 29

`print.birt_qinfo`
 (`quantile_information`), 35
`print.birt_qscore` (`quantile_score`), 36
`print.birt_qtest` (`quantile_test`), 39
`print.birt_quantile` (`birt_quantile`), 13
`print.birt_rt_info` (`rt_information`), 41
`print.birt_rtmb` (`birt_rtmb`), 14
`print.birt_sbc` (`check_sbc`), 16
`print.birt_summary` (`birt_summary`), 15
`print.birt_table` (`birt_summary`), 15
`print.birt_tbl` (`birt_print`), 12
`print.birt_values` (`birt_print`), 12

`quantile_information`, 35
`quantile_score`, 36
`quantile_test`, 39
`quantiles`, 40

`reliability` (`birt-class`), 11
`reliability.birt_quantile`
 (`birt_quantile`), 13
`reliability.birt_rtmb` (`birt_rtmb`), 14
`rt_information`, 41
`rtirt_syntax`, 42
`rtmb_code`, 43
`rtmb_control`, 44

`score_test`, 46
`score_test.birt_qscore`
 (`quantile_score`), 36
`scores` (`birt-class`), 11
`scores.birt_rtmb` (`birt_rtmb`), 14
`shash`, 48
`sim_hgrm`, 49
`sim_rtirt`, 50
`subset_chains`, 51
`summary.birt` (`birt-class`), 11
`summary.birt_quantile` (`birt_quantile`),
 13
`summary.birt_rtmb` (`birt_rtmb`), 14

`vcov.birt_rtmb` (`birt_rtmb`), 14