Package ‘PP’

October 14, 2026

Type Package

Title Person Parameter Estimation

Version 1.0.0

Date 2026-10-14

Description Estimation of person parameters (abilities) for the 1, 2, 3 and 4 parameter logistic model, the (generalized) partial credit model and tests that mix dichotomous and polytomous items, under the assumption that the item parameters are known and fixed: maximum likelihood, weighted likelihood (Warm, 1989 <doi:10.1007/BF02294627>), maximum and expected a posteriori and robust estimation (Schuster and Yuan, 2011 <doi:10.3102/1076998610396890>). Further features are jackknife estimates (Wainer and Wright, 1980 <doi:10.1007/BF02293910>), plausible values, the person fit statistics lz, lz* (Snijders, 2001 <doi:10.1007/BF02294437>), infit and outfit, and the simulation of response data. Models fitted with the 'eRm' package can be passed on directly. The package is useful, e.g., when items of an item bank are given to a new sample of persons.

URL https://github.com/manuelreif/PP, https://manuelreif.github.io/PP/

BugReports https://github.com/manuelreif/PP/issues

License GPL-3

LazyData true

VignetteBuilder knitr, rmarkdown

Depends R (>= 4.0)

Encoding UTF-8

Imports graphics, Rcpp (>= 1.0.0), stats, tinyplot (>= 0.8.0), utils

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), knitr, rmarkdown, eRm

NeedsCompilation yes

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

Author Manuel Reif [cre, aut],
Jan Steinfeld [aut] (ORCID: <https://orcid.org/0000-0001-9853-8260>)

Maintainer Manuel Reif <manuel.reif@gmail.com>

Repository CRAN

Date/Publication 2026-10-14 21:52:08 UTC

Contents

PP-package 2
findmodel 4
fourpl_df 5
JKpp 5
Pfit 8
pp-plots 12
PPall 14
PPass 18
ppeo-methods 21
PP_4pl 22
pp_amt 27
PP_gpcm 28
print.pv 31
sim_4pl 34
sim_gpcm 35
sim_mixed 36

Index **38**

PP-package *PP: Person Parameter Estimation*

Description

The PP package estimates person parameters (abilities) in item response theory when the item parameters are known and fixed, e.g. for items from an item bank given to a new sample of persons.

Details

Models the 1PL (Rasch), 2PL, 3PL and 4PL model (PP_4pl), the partial credit and generalized partial credit model (PP_gpcm) and tests that mix both (PPall).

Estimates maximum likelihood (MLE), weighted likelihood (WLE; Warm, 1989), maximum a posteriori (MAP), expected a posteriori (EAP) and a robust Huber-type estimate, each with a standard error and a status per person.

Further analyses jackknife estimates and standard errors (JKpp), plausible values (PV) and the person fit indices lz, lz*, infit and outfit (Pfit).

In one step `PPass` estimates and computes the person fit for a data.frame with item responses and other person variables, or for a model fitted with the eRm package (RM, LLTM, RSM, PCM, LRSM, LPCM).

Simulation `sim_4pl`, `sim_gpcm` and `sim_mixed`.

The numerical core is written in C++. Every person is estimated with a safeguarded Newton method, EAP estimates and plausible values come from the posterior on a grid. The vignettes give an introduction: `vignette("intro_pp", package = "PP")`, `vignette("intro_pfit", package = "PP")` and `vignette("ppass_erm", package = "PP")`.

Author(s)

Maintainer: Manuel Reif <manuel.reif@gmail.com>

Authors:

- Manuel Reif <manuel.reif@gmail.com>
- Jan Steinfeld <jan.d.steinfeld@gmail.com> (ORCID)

References

Barton, M. A., & Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item-response model. *ETS Research Report Series*, 1981(1), i–8.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & M. R. Novick (Eds.), *Statistical theories of mental test scores*. Reading, MA: Addison-Wesley.

Drasgow, F., Levine, M. V., & Williams, E. A. (1985). Appropriateness measurement with polychotomous item response models and standardized indices. *British Journal of Mathematical and Statistical Psychology*, **38**(1), 67–86.

Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement*, **16**, 159–176.

Schuster, C., & Yuan, K. H. (2011). Robust estimation of latent ability in item response models. *Journal of Educational and Behavioral Statistics*, **36**(6), 720–735.

Snijders, T. A. B. (2001). Asymptotic null distribution of person fit statistics with estimated person parameter. *Psychometrika*, **66**(3), 331–342.

Wainer, H., & Wright, B. D. (1980). Robust estimation of ability in the Rasch model. *Psychometrika*, **45**(3), 373–391.

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

See Also

Useful links:

- https://github.com/manuelreif/PP
- https://manuelreif.github.io/PP/
- Report bugs at https://github.com/manuelreif/PP/issues

Examples

```
# 12 items of a 2PL test with known parameters, 100 persons
set.seed(1522)
b  <- seq(-3, 3, length = 12)
sl <- round(runif(12, 0.5, 1.5), 2)
x  <- sim_4pl(beta = b, alpha = sl, lowerA = 0, upperA = 1, theta = rnorm(100))

res <- PP_4pl(respm = x, thres = b, slopes = sl, type = "wle")
head(res$estimates)

# estimates and person fit in one table
PPass(data.frame(id = 1:100, x), items = 2:13, thres = b, slopes = sl)
```

---

findmodel *The model of each item in a mixed test*

Description

This function shows which model `PPall()` uses for each item: an item with one threshold is a 4PL item, an item with several thresholds a GPCM item.

Usage

```
findmodel(thres)
```

Arguments

`thres`	A numeric matrix which contains the threshold parameter for each item. NA is allowed - in fact expected!

Details

Up to version 0.6.4-1 its result was passed to `PPall()` as `model2est`. This is no longer needed: `PPall()` derives the model of each item from `thres` in the same way.

Value

A character vector with one entry per item (column of `thres`): `"4PL"` for items with one threshold, `"GPCM"` for items with more.

Author(s)

Manuel Reif

See Also

PPall

Examples

```
thres <- rbind(0, c(-1, 0.5, -0.2), c(NA, 1.2, 0.4))
findmodel(thres)
```

`fourpl_df` *Simulated data set*

Description

This dataset contains the responses of 60 persons to 12 items of a 3PL test, simulated with the `sim_4pl()` function, together with an id and the sex of each person. The item parameters are stored as attributes.

Format

A data.frame with 60 rows and 14 columns:

`id` person id

`sex` sex of the person (`"m"` or `"w"`)

`Item1,...,Item12` the responses (0 or 1, `NA` = not answered)

The item parameters are the attributes `diffpar` (difficulties), `slopes`, `lowerA` (lower asymptotes) and `upperA` (upper asymptotes, all 1), e.g. `attr(fourpl_df, "diffpar")`.

Source

simulation

See Also

sim_4pl

`JKpp` *Run a jackknife*

Description

This function uses a jackknife approach to compute person parameters. The jackknife ability measure is based on primarily estimated models (`PP_4pl()`, `PP_gpcm()` or `PPall()`) - so the function is applied on the estimation objects, and jackknifed ability measures are returned.

Usage

```
JKpp(estobj, ...)

## S3 method for class 'ppeo'
JKpp(
  estobj,
  cmeth = "mean",
  maxsteps = NULL,
  exac = NULL,
  fullmat = FALSE,
  ...
)

## S3 method for class 'jk'
print(x, digits = 4, ...)

## S3 method for class 'jk'
summary(object, nrowmax = 15, digits = 4, ...)

## S3 method for class 'ppass'
JKpp(estobj, ...)
```

Arguments

estobj	An object which originates from using PP_gpcm(), PP_4pl() or PPall().
...	More input.
cmeth	Choose the centering method, to summarize the pseudovalues of a person to one single ability estimate. There are three valid entries: "mean", "median" and "AMT" (see Details for further description).
maxsteps	The maximum number of Newton steps per leave-one-out sample. If NULL (default), the value of estobj is used.
exac	How accurate are the leave-one-out estimates supposed to be? If NULL (default), the value of estobj is used.
fullmat	Default = FALSE. If TRUE, the function returns the whole jackknife matrix of leave-one-out estimates, which is the basis for the jackknife estimator.
x, object	An object of class jk, the result of JKpp().
digits	Number of significant digits to print.
nrowmax	summary(): how many persons to show. Default = 15.

Details

For each person, every answered item is left out once and the ability is estimated again from the remaining items, with the same estimation type, prior and range as in estobj (MLE, WLE, MAP and robust estimates start at the full estimate; EAP estimates use the posterior grid of the full estimate). Items a person did not answer are not left out: with L answered items and the leave-one-out estimates $\hat{\theta}_{(-j)}$, the pseudovalues are $L\hat{\theta} - (L-1)\hat{\theta}_{(-j)}$. The jackknife estimate is their mean,

median or AMT (cmeth), the standard error is $\sqrt{\sum_j (ps_j - \bar{ps})^2/(L(L-1))}$ (Wainer & Wright, 1980).

A leave-one-out sample may have no finite estimate, e.g. with type = "mle" when the only wrong answer is left out. Such samples (and samples at the bound of range or not converged) are left out of the person's jackknife, counted in the column excluded, and a warning is given; WLE avoids this in most cases. A person whose full estimate has no solution (status "extreme", "no_response", "boundary" or "not_converged" in estobj$estimates) or who has fewer than two usable leave-one-out samples (status "too_few_items") gets NA.

Up to version 0.6.4 the standard error mixed the leave-one-out estimates with the formula for the pseudovalues and was too small by a factor of about $L-1$, all items counted as answered, and leave-one-out samples without a finite estimate were dropped silently.

AMT-robustified jackknife: When choosing cmeth = "AMT", the pseudovalues are combined as in scheme 4 of Wainer & Wright (1980): they are split into the pseudovalues of the right and of the wrong answers (polytomous items: answers at or above the expected score at $\hat{\theta}$ and below it), each group is summarised by Andrews' sine M-estimator (AMT), one of the winners in the Princeton Robustness Study (Andrews et al., 1972), and the two values are averaged with the group sizes as weights. Within a group, T solves $\sum_j \psi((ps_j - T)/s) = 0$ with $\psi(x) = \sin(x/2.1)$ for $|x| \leq 2.1\pi$ and 0 otherwise, where s is the median absolute deviation of the group's pseudovalues from their median; pseudovalues further than $2.1\pi s$ from T get no weight. Up to version 0.6.4 the AMT had a sign error, no scale and maximised the sum of the sines.

Value

An object of class jk, a list with

- **estimates** — A data.frame with one row per person (in the order of respm): estimate and SE of the jackknife, L the number of answered items, excluded the number of leave-one-out samples without a finite estimate and status ("ok", "too_few_items" or the status of the full estimate).
- **resjk** — The matrix of estimate and SE, as in earlier versions.
- **type, cmeth, call** — The estimation type of estobj, the centering method and the call.
- **jk_mat** — Only with fullmat = TRUE: the persons x items matrix of leave-one-out estimates, NA for items not answered and excluded samples.

Author(s)

Manuel Reif

References

Wainer, H., & Wright, B. D. (1980). Robust estimation of ability in the Rasch model. Psychometrika, 45(3), 373-391.

Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H., & Tukey, J. W. (1972). Robust estimates of location: Survey and advances. Princeton University Press.

See Also

PP_gpcm, PP_4pl, PPall

Examples

```
set.seed(1623)
b  <- seq(-3, 3, length = 12)
sl <- round(runif(12, 0.5, 1.5), 2)
x  <- sim_4pl(beta = b, alpha = sl, lowerA = 0, upperA = 1, theta = rnorm(20, sd = 1.5))

res <- PP_4pl(respm = x, thres = b, slopes = sl, type = "wle")
jk <- JKpp(res)
summary(jk)
# WLE and jackknife side by side
head(cbind(res$estimates[, c("estimate", "SE")], jk = jk$estimates[, c("estimate", "SE")]))

# robust centering: the AMT of Wainer & Wright (1980)
head(JKpp(res, cmeth = "AMT")$estimates)

# partial credit model, EAP
thres <- rbind(0, c(-1, -0.5, 0, 0.4), c(0.5, 0.8, 1, 1.5))
xg <- sim_gpcm(thres = thres, alpha = 1, theta = rnorm(20))
JKpp(PP_gpcm(respm = xg, thres = thres, type = "eap"))
```

`Pfit` *Person-Fit statistics*

Description

Compute the person fit statistics lz, lz*, infit and outfit for the 1-PL, 2-PL, 3-PL, 4-PL model, the GPCM and mixed tests, at the person estimates of `PP_4pl()`, `PP_gpcm()` or `PPall()`.

Usage

```
Pfit(pp, fitindices = c("lz", "lzstar", "infit", "outfit"), SE = FALSE)

## S3 method for class 'ppeo'
Pfit(pp, fitindices = c("lz", "lzstar", "infit", "outfit"), SE = FALSE)

## Default S3 method:
Pfit(pp, fitindices = c("lz", "lzstar", "infit", "outfit"), SE = FALSE)

## S3 method for class 'PPfit'
print(x, digits = 4, ...)

## S3 method for class 'PPfit'
summary(object, alpha = 0.05, digits = 4, ...)
```

Arguments

pp	An estimation object, the result of `PP_4pl()`, `PP_gpcm()` or `PPall()` with type `"mle"`, `"wle"` or `"map"`. The responses are taken from it.
fitindices	Character vector of the desired person fit statistics, any of `c("lz", "lzstar", "infit", "outfit")`.
SE	Logical: if `TRUE`, jackknife standard errors are computed.
x, object	An object of class `PPfit`.
digits	Number of significant digits to print.
...	Further arguments, passed to `print()`.
alpha	Significance level: persons with $\lvert z\rvert > z_{1-\alpha/2}$ (lz, lz*, `in_t`, `ou_t`) are counted as flagged.

Details

All statistics are computed at the estimate $\hat{\theta}$ of each person, over the items the person answered (missing responses are left out of every sum). With the observed response x_i, the expected score E_i, its variance V_i and fourth central moment C_i of item i, the residual is $Y_i = x_i - E_i$, and for the n answered items:

- **infit** (weighted mean square, Wright & Masters, 1982): $\sum Y_i^2 / \sum V_i$, with variance $q^2 = \sum(C_i - V_i^2)/(\sum V_i)^2$.
- **outfit** (unweighted mean square): $\sum(Y_i^2/V_i)/n$, with variance $q^2 = \sum(C_i/V_i^2)/n^2 - 1/n$.
- Both mean squares MS are standardised by the Wilson-Hilferty transformation $t = (MS^{1/3} - 1)(3/q) + q/3$. `chisq` is $\sum Y_i^2/V_i$ with `df` $= n - 1$ and the p-value of the χ^2 distribution.
- **lz** (Drasgow, Levine & Williams, 1985): the log-likelihood $l_0 = \sum \log P_i(x_i)$, standardised by its expectation and variance under the model, $(l_0 - \sum E \log P_i)/\sqrt{\sum Var \log P_i}$. `lz_unst` is l_0/n.
- **lz*** (Snijders, 2001) corrects lz for the estimated θ: with the item information I, $c_n = \sum Cov(\log P_i, s_i)/I$ (s_i the score of item i) and the term r_0 of the estimating equation (mle: 0, wle: $J/(2I)$, map: $(\mu - \theta)/\sigma^2$), lz* $= (l_0 - \sum E \log P_i + c_n r_0)/\sqrt{\sum Var \log P_i - c_n^2 I}$. For dichotomous items this is Snijders' lz*, for polytomous items and mixed tests the generalisation of Sinharay (2016).

Person fit needs an estimate of type `"mle"`, `"wle"` or `"map"`. Persons without an estimate (status other than `"ok"` in `pp$estimates`, e.g. an extreme score with mle) get `NA`. With a single answered dichotomous item lz* is not defined (the corrected variance is 0) and the χ^2 p-value has no degree of freedom; both are `NA`.

With `SE = TRUE` every statistic gets a jackknife standard error: for each answered item j, θ is estimated without item j (the leave-one-out estimates of `JKpp()`) and the statistic $f_{(-j)}$ is computed on the remaining items. With L answered items and m leave-one-out samples with a finite estimate, $SE = (L-1)\sqrt{\sum_j (f_{(-j)} - \bar{f})^2/(m(m-1))}$, which is $\sqrt{(L-1)/L \sum_j (f_{(-j)} - \bar{f})^2}$ when no sample is left out (Efron & Stein, 1981).

Up to version 0.6.4, lz* summed over all items also for persons with missing responses, the GPCM infit and outfit used the observed instead of the possible categories and ignored the slopes, mixed

tests were not supported, the jackknife standard error of lz used the prior SD as variance, and all results were rounded. The responses were passed again as first argument `respm`; since 1.0.0 they are taken from `pp`, and `respm` is gone: call `Pfit(res)` instead of `Pfit(respm = X, pp = res)`.

Value

An object of class `PPfit`: a list with one matrix per statistic (persons in rows, in the order of the response matrix):

- `lz`: `lz`, `lz_unst`
- `lzstar`: `lzstar`
- `infit`: `infit` (mean square), `in_t`, `in_chisq`, `in_df`, `in_pv`
- `outfit`: `outfit` (mean square), `ou_t`, `ou_chisq`, `ou_df`, `ou_pv`
- with `SE = TRUE` also `lz_se`, `lzs_se`, `in_se`, `ou_se`.

`summary()` gives, per statistic, the number of persons, mean, SD, range and the number of persons flagged at level `alpha`.

Author(s)

Jan Steinfeld, Manuel Reif

References

Armstrong, R. D., Stoumbos, Z. G., Kung, M. T. & Shi, M. (2007). On the performance of the lz person-fit statistic. *Practical Assessment, Research & Evaluation*, **12(16)**.

De La Torre, J., & Deng, W. (2008). Improving Person-Fit Assessment by Correcting the Ability Estimate and Its Reference Distribution. Journal of Educational Measurement, **45(2)**, 159-177.

Drasgow, F., Levine, M. V. & Williams, E. A. (1985) Appropriateness measurement with polychotomous item response models and standardized indices. *British Journal of Mathematical and Statistical Psychology*, **38(1)**, 67–86.

Efron, B., & Stein, C. (1981). The jackknife estimate of variance. *The Annals of Statistics*, **9(3)**, 586-596.

Karabatsos, G. (2003) Comparing the Aberrant Response Detection Performance of Thirty-Six Person-Fit Statistics. *Applied Measurement In Education*, **16(4)**, 277–298.

Magis, D., Raiche, G. & Beland, S. (2012) A didactic presentation of Snijders's lz* index of person fit with emphasis on response model selection and ability estimation. *Journal of Educational and Behavioral Statistics*, **37(1)**, 57–81.

Meijer, R. R. & Sijtsma, K. (2001) Methodology review: Evaluating person fit. *Applied Psychological Measurement*, **25(2)**, 107–135.

Molenaar, I. W. & Hoijtink, H. (1990) The many null distributions of person fit indices. *Psychometrika*, **55(1)**, 75–106.

Reise, S. P. (1990). A comparison of item-and person-fit methods of assessing model-data fit in IRT. *Applied Psychological Measurement*, **14(2)**, 127-137.

Sinharay, S. (2016). Asymptotically correct standardization of person-fit statistics beyond dichotomous items. *Psychometrika*, **81(4)**, 992–1013.

Snijders, T. A. B. (2001) Asymptotic null distribution of person fit statistics with estimated person parameter. *Psychometrika*, **66(3)**, 331–342.

Wright, B. D. & Masters, G. N. (1990). Computation of OUTFIT and INFIT Statistics. *Rasch Measurement Transactions*, 3:4, 84-85.

Wright, B. D., & Masters, G. N. (1982). *Rating Scale Analysis. Rasch Measurement.* MESA Press, 5835 S. Kimbark Avenue, Chicago, IL 60637.

See Also

PPall, PP_4pl, PP_gpcm, PPass, JKpp

Examples

```
## dichotomous items

set.seed(1337)
diffpar <- seq(-3, 3, length = 15)
sl      <- round(runif(15, 0.5, 1.5), 2)
la      <- round(runif(15, 0, 0.25), 2)
awm     <- sim_4pl(beta = diffpar, alpha = sl, lowerA = la, upperA = rep(1, 15),
                   theta = rnorm(100))

# 3PL, WLE
res3plwle <- PP_4pl(respm = awm, thres = diffpar, slopes = sl, lowerA = la, type = "wle")

# the responses are taken from the estimation object
fit3pl <- Pfit(res3plwle, fitindices = c("lz", "lzstar", "infit", "outfit"))
head(fit3pl$lzstar)
summary(fit3pl)

# with missing responses and jackknife standard errors
awm_na <- awm
awm_na[sample(length(awm), 300)] <- NA
res3plna <- PP_4pl(respm = awm_na, thres = diffpar, slopes = sl, lowerA = la, type = "wle")
head(Pfit(res3plna, fitindices = c("lz", "infit"), SE = TRUE)$lz)

## GPCM

thres <- rbind(0, c(-1, -0.5, 0), c(0.5, 0.8, 1), c(1.5, 2, 2.5))
slg   <- c(0.8, 1.2, 1)
respg <- sim_gpcm(thres = thres, alpha = slg, theta = rnorm(50))
resgwle <- PP_gpcm(respm = respg, thres = thres, slopes = slg, type = "wle")
summary(Pfit(resgwle))

## GPCM and 4PL mixed

THRESx <- matrix(c(-2, -1.23, 1.11, 3.48, 1, 2, -1, -0.2, 0.5, 1.3, -0.8, 1.5), nrow = 2)
THRESx[2, 1:3] <- NA
slm <- c(0.5, 1, 1.5, 1.1, 1, 0.98)
lam <- c(0.02, 0.1, 0, 0, 0, 0)
uam <- c(0.97, 0.91, 1, 1, 1, 1)
awmatrix <- matrix(c(1, 0, 1, 0, 1, 1, 1, 0, 0, 1,
```

```
                    2, 0, 0, 0, 0, 0, 0, 0, 0, 1,
                    1, 2, 2, 1, 1, 1, 1, 0, 0, 1), byrow = TRUE, nrow = 5)
resmixed <- PPall(respm = awmatrix, thres = THRESx, slopes = slm, lowerA = lam,
                  upperA = uam, type = "wle")
Pfit(resmixed, fitindices = c("lz", "lzstar"))
```

---

pp-plots *Plots of person estimates and person fit*

Description

Three plots for the results of PPass(), PP_4pl(), PP_gpcm() and PPall(): a person fit map, the measurement precision of the test, and the response profiles of single persons.

Usage

```
## S3 method for class 'ppass'
plot(
  x,
  which = c("fit", "precision", "profile"),
  person = NULL,
  index = NULL,
  alpha = 0.05,
  label = NULL,
  curve = NULL,
  ...
)

## S3 method for class 'ppeo'
plot(
  x,
  which = c("precision", "fit", "profile"),
  person = NULL,
  index = NULL,
  alpha = 0.05,
  curve = NULL,
  ...
)
```

Arguments

x	A result of PPass() (class ppass) or of PP_4pl(), PP_gpcm() or PPall() (class ppeo).
which	The plot: "fit" (person fit map), "precision" (measurement precision) or "profile" (person profiles). The default is "fit" for PPass results with person fit and "precision" otherwise.

person	For which = "profile": the persons to show, as row positions. The default are the (at most 4) flagged persons with the strongest misfit: the lowest values of lz and lz*, the highest of the infit and outfit t.
index	The person fit statistic used to flag persons: one of "lzstar", "lz", "in_t" and "ou_t". For PPass results the default is the first of these that was computed, for estimation results "lzstar"; for estimation results it is computed with Pfit(), which needs an estimate of type "mle", "wle" or "map".
alpha	Significance level: persons with $\|index\| > z_{1-\alpha/2}$ are flagged, and responses with $\|z\| > z_{1-\alpha/2}$ are unexpected.
label	For PPass results: the name of a column of x (e.g. an id) whose values label the persons. By default persons are labelled with their row position.
curve	For which = "precision": draw the standard error curve of all items? NULL (default) draws it when every person answered every item (and, for MAP and EAP, all persons have the same prior variance).
...	Further arguments for tinyplot::tinyplot(), e.g. main, xlab, ylab or palette.

Details

Person fit map (which = "fit"): the person fit statistic index against the estimate, one point per person with status "ok". The dashed lines are the critical values $\pm z_{1-\alpha/2}$; persons beyond them are flagged (the same rule as in summary() of a PPass result), and the at most 10 most extreme ones are labelled.

Measurement precision (which = "precision"): the upper panel shows the distribution of the estimates, the lower panel the standard error of each person against the estimate. Because the item parameters are known, the standard error of the whole test follows from the test information $I(\theta)$, the sum of the item informations: $1/\sqrt{I(\theta)}$, and $1/\sqrt{I(\theta) + 1/\sigma^2}$ for MAP and EAP estimates with the prior variance σ^2. This curve is drawn when every person answered every item (curve = NULL); in incomplete or adaptive designs each person has an own set of items, and the curve of all items holds for nobody. The ticks at the bottom mark the location of each item, the ability at which it gives the most information (for the 1PL and 2PL its difficulty). The plot shows where the test measures accurately and whether the persons are located there. For robust estimates the curve is the model standard error without the Huber weights; for EAP estimates the standard error is the posterior SD.

Person profile (which = "profile"): one panel per person in person. Each answered item is a point: the expected score at the estimate (x axis, relative to the maximum score of the item, easy items on the left) against the observed score (y axis, relative to the maximum score). Responses in line with the model lie close to the diagonal. Items with a standardised residual $|z| = |x - E|/\sqrt{V}$ above $z_{1-\alpha/2}$ are marked as unexpected and labelled with the item name: a wrong answer to an easy item at the bottom left, a right answer to a hard item at the top right. The profile shows which responses make a person conspicuous. The panel title gives the estimate, its standard error and the fit statistic index.

The plots are drawn with the tinyplot package, so a theme set with tinyplot::tinytheme() applies to them.

Value

The plotted data, invisibly: for "fit" a data.frame with one row per plotted person (person, label, estimate, value of index, flagged); for "precision" a list with the data.frames persons (person, estimate, SE), curve (theta, info, SE; NULL without curve) and items (item, location); for "profile" a data.frame with one row per answered item and person (person, item, response, max, expected, z, unexpected).

Author(s)

Manuel Reif

See Also

PPass, Pfit, PP_4pl, PP_gpcm, PPall

Examples

```
set.seed(1337)
diffpar <- seq(-3, 3, length = 20)
sl <- round(runif(20, 0.5, 1.5), 2)
awm <- sim_4pl(beta = diffpar, alpha = sl, lowerA = rep(0, 20), upperA = rep(1, 20),
               theta = rnorm(300))
# 10 persons answer at random
awm[1:10, ] <- matrix(rbinom(200, 1, 0.5), 10)
dat <- data.frame(id = sprintf("P%03d", 1:300), awm)

res <- PPass(dat, items = 2:21, thres = diffpar, slopes = sl)

# who does not fit?
plot(res, label = "id")

# where does the test measure accurately, and are the persons there?
plot(res, which = "precision")

# which responses make the flagged persons conspicuous?
plot(res, which = "profile", label = "id")

# the same plots for an estimation result
est <- PP_4pl(respm = awm, thres = diffpar, slopes = sl, type = "map")
plot(est)
plot(est, which = "profile", person = 1:2)
```

PPall — *Estimate Person Parameters*

Description

Compute person parameters for the 1,2,3,4-PL model and for the GPCM. Choose between ML, WL, MAP, EAP and robust estimation. Use this function if 4-PL items and GPCM items are mixed for each person.

Usage

```
PPall(
  respm,
  thres,
  slopes = NULL,
  lowerA = NULL,
  upperA = NULL,
  theta_start = NULL,
  mu = NULL,
  sigma2 = NULL,
  type = "wle",
  model2est = NULL,
  maxsteps = 100,
  exac = 0.001,
  H = 1,
  ctrl = list(),
  range = c(-6, 6),
  verbose = FALSE
)
```

Arguments

respm	An integer matrix (or a data.frame), which contains the responses of the examinees. A persons x items matrix is expected. Missing responses are NA.
thres	A numeric matrix which contains the threshold parameter for each item. If the first row of the matrix is not set to zero (only zeroes in the first row) - then a row-vector with zeroes is added by default.
slopes	A numeric vector, which contains the slope parameters for each item - one parameter per item is expected. A single value is used for all items. If NULL (default), all slopes are 1.
lowerA	A numeric vector, which contains the lower asymptote parameters (kind of guessing parameter) for each item. It applies to the 4PL items only; a value other than 0 for a GPCM item is ignored with a warning. A single value is used for all items. If NULL (default), all lower asymptotes are 0.
upperA	numeric vector, which contains the upper asymptote parameters for each item. It applies to the 4PL items only; a value other than 1 for a GPCM item is ignored with a warning. A single value is used for all items. If NULL (default), all upper asymptotes are 1.
theta_start	A vector which contains a starting value for each person. If NULL is submitted, the starting values are set automatically. If a scalar is submitted, this start value is used for each person.
mu	A numeric vector of location parameters for each person, the normal prior for MAP and EAP estimation and for PV. A single value is used for all persons. If nothing is submitted this is set to 0 for each person. MLE, WLE and robust estimation have no prior: a value given there is ignored with a warning, and the result stores NA.

sigma2 — A numeric vector of variance parameters for each person, the normal prior for MAP and EAP estimation and for PV. A single value is used for all persons. If nothing is submitted this is set to 1 for each person. MLE, WLE and robust estimation have no prior: a value given there is ignored with a warning, and the result stores NA.

type — Which maximization should be applied? There are five valid entries possible: "mle", "wle", "map", "eap" and "robust". To choose between the methods, or just to get a deeper understanding the papers mentioned below are quite helpful. The default is "wle" which is a good choice in many cases. With "eap" the estimate is the mean and the SE the standard deviation of the posterior, computed on a grid of ability values (see PV). Plausible values (PV) need the prior of "eap" or "map".

model2est — No longer needed: the model of each item follows from thres. A value given is ignored with a warning. (A dichotomous item declared as "GPCM" is the same as a 4PL item with lower asymptote 0 and upper asymptote 1.)

maxsteps — The maximum number of Newton steps per person. Default = 100.

exac — How accurate are the estimates supposed to be? The iteration of a person stops when the step is smaller than exac. Default is 0.001.

H — In case type = "robust" a Huber ability estimate is performed, and H modulates how fast the downweighting takes place (for more Details read Schuster & Yuan 2011).

ctrl — No longer used. Its options killdupli and skipcheck of version 0.6.4 saved time with the old estimation routine; since 1.0.0 every person is estimated separately, and options given are ignored with a warning.

range — The interval c(lower, upper) in which mle, wle, map and robust estimates are searched. If the estimating equation has no root inside, the estimate is set to the bound it points to and its status is "boundary". Default c(-6, 6).

verbose — If TRUE, a message names the model and the estimation type. Default FALSE.

Details

For a test with both: dichotomous and polytomous items which have been scaled under 1/2/3/4-PL model or the GPCM, use this function to estimate the person ability parameters. The model of each item follows from thres: an item with one threshold is a 4PL item (with slopes, lowerA and upperA), an item with several thresholds a GPCM item (see findmodel).

Please note, that robust estimation with (Huber ability estimate) polytomous items is still experimental!

Value

An object of class ppeo (and fourpl, gpcm or gpcm4pl), a list with

estimates A data.frame with one row per person, in the order of respm: estimate, SE, nsteps (Newton steps of the person; 0 for eap) and status, a factor with the levels

- "ok": converged inside range,

- "extreme": mle or robust estimate of a person with all answers in the lowest or highest category, the estimate is -Inf or Inf and the SE NA,
- "boundary": the estimating equation has no root inside range, the estimate is the bound,
- "no_response": the person answered no item, estimate and SE are NA (eap: prior mean and SD),
- "not_converged": no convergence within maxsteps.

A warning reports persons with status "boundary" or "not_converged".

type, call The estimation type and the call.

model The checked and prepared inputs.

resPP The result in the layout of earlier versions: resPP$resPP is the persons x 2 matrix of estimates and SEs, resPP$nsteps the largest number of steps.

Use coef() for the estimates, as.data.frame() for the table and plot() for the measurement precision, the person fit map or person profiles (see pp-plots).

Author(s)

Manuel Reif

References

Baker, Frank B., and Kim, Seock-Ho (2004). Item Response Theory - Parameter Estimation Techniques. CRC-Press.

Barton, M. A., & Lord, F. M. (1981). An Upper Asymptote for the Three-Parameter Logistic Item-Response Model.

Magis, D. (2013). A note on the item information function of the four-parameter logistic model. Applied Psychological Measurement, 37(4), 304-315.

Muraki, Eiji (1992). A Generalized Partial Credit Model: Application of an EM Algorithm. Applied Psychological Measurement, 16, 159-176.

Muraki, Eiji (1993). Information Functions of the Generalized Partial Credit Model. Applied Psychological Measurement, 17, 351-363.

Samejima, Fumiko (1993). The bias function of the maximum likelihood estimate of ability for the dichotomous response level. Psychometrika, 58, 195-209.

Samejima, Fumiko (1993). An approximation of the bias function of the maximum likelihood estimate of a latent variable for the general case where the item responses are discrete. Psychometrika, 58, 119-138.

Schuster, C., & Yuan, K. H. (2011). Robust estimation of latent ability in item response models. Journal of Educational and Behavioral Statistics, 36(6), 720-735.

Wang, S. and Wang, T. (2001). Precision of Warm's Weighted Likelihood Estimates for a Polytomous Model in Computerized Adaptive Testing. Applied Psychological Measurement, 25, 317-331.

Warm, Thomas A. (1989). Weighted Likelihood Estimation Of Ability In Item Response Theory. Psychometrika, 54, 427-450.

Yen, Y.-C., Ho, R.-G., Liao, W.-W., Chen, L.-J., & Kuo, C.-C. (2012). An empirical evaluation of the slip correction in the four parameter logistic models with computerized adaptive testing. Applied Psychological Measurement, 36, 75-87.

See Also

PPass, PP_gpcm, PP_4pl, JKpp, PV

Examples

```
# items 1 to 3 are 4PL items (one threshold), items 4 to 6 GPCM items
thres <- matrix(c(-2, -1.23, 1.11, 3.48, 1, 2, -1, -0.2, 0.5, 1.3, -0.8, 1.5), nrow = 2)
thres[2, 1:3] <- NA
sl <- c(0.5, 1, 1.5, 1.1, 1, 0.98)
la <- c(0.02, 0.1, 0, 0, 0, 0)
ua <- c(0.97, 0.91, 1, 1, 1, 1)
set.seed(1811)
x <- sim_mixed(thres = thres, alpha = sl, lowerA = la, upperA = ua, theta = rnorm(30))

res <- PPall(respm = x, thres = thres, slopes = sl, lowerA = la, upperA = ua, type = "wle")
summary(res)

res_eap <- PPall(respm = x, thres = thres, slopes = sl, lowerA = la, upperA = ua, type = "eap")
head(res_eap$estimates)
```

PPass *Person assessment: estimates and person fit in one step*

Description

PPass (**P**erson **P**arameter **ass**essment) estimates the person parameters and computes the person fit in one call, and returns both in one table, next to the other person variables. It takes a data.frame or matrix with item responses and known item parameters, or a model fitted with the eRm package.

Usage

```
PPass(...)

## Default S3 method:
PPass(
  respdf,
  items = "all",
  fitindices = c("lz", "lzstar", "infit", "outfit"),
  SE = FALSE,
  ...
)

## S3 method for class 'eRm'
PPass(
  RMobj,
  fitindices = c("lz", "lzstar", "infit", "outfit"),
  SE = FALSE,
  ...
```

```
)

## S3 method for class 'ppass'
print(x, n = 10, digits = 3, ...)

## S3 method for class 'ppass'
summary(object, alpha = 0.05, digits = 3, ...)

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

Arguments

`respdf`	A data.frame or matrix with one row per person, which contains the item responses and possibly other person variables.
`items`	The item columns of `respdf`: `"all"` (default), their positions or their names.
`fitindices`	The person fit indices to compute, any of `"lz"`, `"lzstar"`, `"infit"` and `"outfit"` (default all).
`SE`	If `TRUE`, jackknife standard errors of the fit indices are added (see `Pfit()`).
`RMobj`	A model fitted with the `eRm` package (`RM()`, `LLTM()`, `RSM()`, `PCM()`, `LRSM()` or `LPCM()`).
`x`, `object`	A result of `PPass()`.
`n`	Number of persons to print.
`digits`	Number of significant digits to print.
`alpha`	Significance level for flagging persons: $\lvert z\rvert > z_{1-\alpha/2}$ for lz*, lz, `in_t` or `ou_t` (the first one computed).
`row.names`, `optional`	Ignored.
`...`	Arguments for the estimation: the item parameters (`PPass.default` only) and `type`, `mu`, `sigma2`, `theta_start`, `range`, `maxsteps`, `exac`, `H` and `verbose`, see `PP_4pl()`, `PP_gpcm()` and `PPall()`. The default estimate is WLE.

Details

Data with item parameters (`PPass.default`): the columns `items` of `respdf` are the responses, the other columns are kept as person variables. The item parameters are passed in `...` (`thres`, `slopes`, `lowerA`, `upperA`), and the model follows from them: `thres` alone gives the 1PL model, with `slopes` the 2PL, with `lowerA` the 3PL and with `upperA` the 4PL model, estimated with `PP_4pl()`. A threshold matrix with more than one threshold for some item gives the partial credit model (all slopes 1) or the generalized partial credit model, estimated with `PP_gpcm()`. With such polytomous items and `lowerA` or `upperA`, the test is a mixed test (items with one threshold are 4PL items), estimated with `PPall()`. The arguments `mod` and `model2est` of earlier versions are no longer needed; they are ignored with a warning.

eRm models (`PPass.eRm`): all models of the `eRm` package are supported, the Rasch model (`RM`), the linear logistic test model (`LLTM`), the rating scale model (`RSM`), the partial credit model (`PCM`) and the linear models `LRSM` and `LPCM`, also with several time points (`mpoints`) and groups (`groupvec`).

The responses and item parameters come from the fitted object. eRm stores easiness parameters per category, $\beta_{ih} = -\sum_{k \le h} \delta_{ik}$; they are turned into the thresholds $\delta_{ih} = \beta_{i,h-1} - \beta_{ih}$, so for the dichotomous models the difficulty is $-\beta_i$. With time points or groups every combination of item, time point and group is a virtual item with its own parameters; a person answers the virtual items of their group, and gets one ability over all time points. The person parameters are estimated with `PP_4pl()` (RM, LLTM) or `PP_gpcm()` with all slopes 1. The maximum likelihood estimates (`type = "mle"`) equal those of `eRm::person.parameter()`; unlike `eRm`, `PPass` also gives estimates for models with group contrasts, and WLE, MAP, EAP and robust estimates, person fit, jackknife and plausible values. The `eRm` package is not needed to run `PPass` on such an object.

Person fit is computed by `Pfit()` and needs an estimate of type `"mle"`, `"wle"` or `"map"`. For `type = "eap"` and `"robust"` `PPass` returns the estimates only, with a message.

Value

A data.frame of class `ppass` with one row per person: the person variables (the columns of `respdf` that are not items; for eRm models with groups the column `group`), `estimate`, `SE` and `status` of the estimate (see `PP_4pl()`) and the columns of the chosen person fit indices (see `Pfit()`). It can be used like any data.frame. The estimation object is kept as attribute, so `JKpp()` and `PV()` can be applied to the result directly; `print()` and `summary()` show an overview with the persons flagged by the person fit, `plot()` draws the person fit map, the measurement precision or person profiles (see pp-plots), and `as.data.frame()` returns a plain data.frame.

Author(s)

Manuel Reif, Jan Steinfeld

See Also

PP_4pl, PP_gpcm, PPall, Pfit, JKpp, PV, pp-plots

Examples

```
## data with item parameters

set.seed(1337)
# difficulties and slopes of 15 items
diffpar <- seq(-3, 3, length = 15)
sl <- round(runif(15, 0.5, 1.5), 2)
# responses of 200 persons, plus an id and a group variable
awm <- sim_4pl(beta = diffpar, alpha = sl, lowerA = rep(0, 15), upperA = rep(1, 15),
               theta = rnorm(200))
dat <- data.frame(id = 1:200, group = rep(c("a", "b"), 100), awm)

# 2PL, WLE estimates and all person fit indices; id and group are kept
res <- PPass(dat, items = 3:17, thres = diffpar, slopes = sl)
res
summary(res)

# person fit map, flagged persons labelled with their id
plot(res, label = "id")
```

```
# a plain data.frame for further use
head(as.data.frame(res))

# jackknife directly from the result
jk <- JKpp(res)


## real data (adaptive test, 1PL)

data(pp_amt)
d <- pp_amt$daten_amt
rd_res <- PPass(d, items = 8:ncol(d), thres = pp_amt$betas[, 2], fitindices = "lz")
head(rd_res)


## models fitted with eRm

if (requireNamespace("eRm", quietly = TRUE)) {

  # Rasch model
  my_rm <- eRm::RM(eRm::raschdat1)
  res_rm <- PPass(my_rm)
  summary(res_rm)

  # the maximum likelihood estimates equal eRm::person.parameter()
  res_mle <- PPass(my_rm, type = "mle", fitindices = "lzstar")

  # partial credit model
  set.seed(2751)
  thres <- rbind(0, matrix(c(-2, -1, 0.5, -1.2, 0.3, NA, -0.5, 1, NA, 0, 1.5, 2), 3))
  simdat <- sim_gpcm(thres = thres, alpha = rep(1, 4), theta = rnorm(300))
  res_pcm <- PPass(eRm::PCM(simdat))

  # linear partial credit model with two time points and two groups
  simdat2 <- sim_gpcm(thres = thres, alpha = rep(1, 4), theta = rnorm(300, 0.5))
  my_lpcm <- eRm::LPCM(cbind(simdat, simdat2), mpoints = 2, groupvec = rep(1:2, 150))
  res_lpcm <- PPass(my_lpcm)
  head(res_lpcm)

  # plausible values need an EAP or MAP estimate
  pv <- PV(PPass(my_rm, type = "map"))
}
```

`ppeo-methods` *Methods for estimation results*

Description

Print, summarise and extract the person estimates of `PP_4pl()`, `PP_gpcm()` and `PPall()` (objects of class ppeo). The plots are described in pp-plots.

Usage

```
## S3 method for class 'ppeo'
print(x, digits = 4, ...)

## S3 method for class 'ppeo'
summary(object, nrowmax = 15, digits = 4, ...)

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

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

Arguments

`x`, `object`	An object of class `ppeo`.
`digits`	Number of significant digits to print.
`...`	Further arguments, passed to `print()`.
`nrowmax`	When printing the estimates - how many rows should be shown? Default = 15.
`row.names`, `optional`	Ignored; the rows are the persons in the order of the response matrix.

Value

`coef()` returns the estimates as a vector (persons in the order of the response matrix), `as.data.frame()` the data.frame `$estimates` (estimate, SE, nsteps, status). `print()` and `summary()` return the object invisibly.

See Also

PP_4pl, PP_gpcm, PPall, pp-plots

PP_4pl *Estimate Person Parameters for the 4-PL model*

Description

Compute Person Parameters for the 1/2/3/4-PL model and choose between five common estimation techniques: ML, WL, MAP, EAP and a robust estimation. All item parameters are treated as fixed.

Usage

```
PP_4pl(
  respm,
  thres,
  slopes = NULL,
  lowerA = NULL,
  upperA = NULL,
  theta_start = NULL,
  mu = NULL,
  sigma2 = NULL,
  type = "wle",
  maxsteps = 100,
  exac = 0.001,
  H = 1,
  ctrl = list(),
  range = c(-6, 6),
  verbose = FALSE
)
```

Arguments

respm	An integer matrix (or a data.frame), which contains the responses of the examinees. A persons x items matrix is expected. Missing responses are NA.
thres	A numeric vector or a numeric matrix which contains the threshold parameter (also known as difficulty parameter or beta parameter) for each item. A matrix with one column per item is also accepted; its first row of zeros (as in PP_gpcm) is optional.
slopes	A numeric vector, which contains the slope parameters for each item - one parameter per item is expected. A single value is used for all items. If NULL (default), all slopes are 1.
lowerA	A numeric vector, which contains the lower asymptote parameters (kind of guessing parameter) for each item. A single value is used for all items. If NULL (default), all lower asymptotes are 0.
upperA	numeric vector, which contains the upper asymptote parameters for each item. A single value is used for all items. If NULL (default), all upper asymptotes are 1.
theta_start	A vector which contains a starting value for each person. If NULL is submitted, the starting values are set automatically. If a scalar is submitted, this start value is used for each person.
mu	A numeric vector of location parameters for each person, the normal prior for MAP and EAP estimation and for PV. A single value is used for all persons. If nothing is submitted this is set to 0 for each person. MLE, WLE and robust estimation have no prior: a value given there is ignored with a warning, and the result stores NA.
sigma2	A numeric vector of variance parameters for each person, the normal prior for MAP and EAP estimation and for PV. A single value is used for all persons. If

nothing is submitted this is set to 1 for each person. MLE, WLE and robust estimation have no prior: a value given there is ignored with a warning, and the result stores NA.

`type` Which maximization should be applied? There are five valid entries possible: "mle", "wle", "map", "eap" and "robust". To choose between the methods, or just to get a deeper understanding the papers mentioned below are quite helpful. The default is `"wle"` which is a good choice in many cases. With `"eap"` the estimate is the mean and the SE the standard deviation of the posterior, computed on a grid of ability values (see PV). Plausible values (PV) need the prior of `"eap"` or `"map"`.

`maxsteps` The maximum number of Newton steps per person. Default = 100.

`exac` How accurate are the estimates supposed to be? The iteration of a person stops when the step is smaller than `exac`. Default is 0.001.

`H` In case `type = "robust"` a Huber ability estimate is performed, and `H` modulates how fast the downweighting takes place (for more Details read Schuster & Yuan 2011).

`ctrl` No longer used. Its options `killdupli` and `skipcheck` of version 0.6.4 saved time with the old estimation routine; since 1.0.0 every person is estimated separately, and options given are ignored with a warning.

`range` The interval `c(lower, upper)` in which mle, wle, map and robust estimates are searched. If the estimating equation has no root inside, the estimate is set to the bound it points to and its status is `"boundary"`. Default `c(-6, 6)`.

`verbose` If `TRUE`, a message names the model and the estimation type. Default `FALSE`.

Details

With this function you can estimate:

- **1-PL model** (Rasch model) by submitting: the data matrix, item difficulties and **nothing else**, since the 1-PL model is merely a 4-PL model with: any slope = 1, any lower asymptote = 0 and any upper asymptote = 1!
- **2-PL model** by submitting: the data matrix, item difficulties and slope parameters. Lower and upper asymptotes are automatically set to 0 and 1 respectively.
- **3-PL model** by submitting anything except the upper asymptote parameters
- **4-PL model** by submitting all parameters.

The probability function of the 4-PL model is:

$$P(x_{ij} = 1|\hat{\alpha}_i, \hat{\beta}_i, \hat{\gamma}_i, \hat{\delta}_i, \theta_j) = \hat{\gamma}_i + (\hat{\delta}_i - \hat{\gamma}_i)\frac{exp(\hat{\alpha}_i(\theta_j - \hat{\beta}_i))}{1 + exp(\hat{\alpha}_i(\theta_j - \hat{\beta}_i))}$$

In our case θ is to be estimated, and the four item parameters are assumed as fixed (usually these are estimates of a former scaling procedure).

The 3-PL model is the same, except that $\delta_i = 1, \forall i$.

In the 2-PL model $\delta_i = 1, \gamma_i = 0, \forall i$.

In the 1-PL model $\delta_i = 1, \gamma_i = 0, \alpha_i = 1, \forall i$.

The **robust** estimation method, applies a Huber-type estimator (Schuster & Yuan, 2011), which downweights responses to items which provide little information for the ability estimation. First a residuum is estimated and on this basis, the weight for each observation is computed.

residuum:

$$r_i = \alpha_i(\theta - \beta_i)$$

weight:

$$w(r_i) = 1 \rightarrow if\, |r_i| \leq H$$

$$w(r_i) = H/|r| \rightarrow if\, |r_i| > H$$

Value

An object of class `ppeo` (and `fourpl`, `gpcm` or `gpcm4pl`), a list with

`estimates` A data.frame with one row per person, in the order of `respm`: `estimate`, `SE`, `nsteps` (Newton steps of the person; 0 for eap) and `status`, a factor with the levels

- `"ok"`: converged inside `range`,
- `"extreme"`: mle or robust estimate of a person with all answers in the lowest or highest category, the estimate is `-Inf` or `Inf` and the SE `NA`,
- `"boundary"`: the estimating equation has no root inside `range`, the estimate is the bound,
- `"no_response"`: the person answered no item, estimate and SE are `NA` (eap: prior mean and SD),
- `"not_converged"`: no convergence within `maxsteps`.

A warning reports persons with status `"boundary"` or `"not_converged"`.

`type`, `call` The estimation type and the call.

`model` The checked and prepared inputs.

`resPP` The result in the layout of earlier versions: `resPP$resPP` is the persons x 2 matrix of estimates and SEs, `resPP$nsteps` the largest number of steps.

Use `coef()` for the estimates, `as.data.frame()` for the table and `plot()` for the measurement precision, the person fit map or person profiles (see pp-plots).

Author(s)

Manuel Reif

References

Baker, Frank B., and Kim, Seock-Ho (2004). Item Response Theory - Parameter Estimation Techniques. CRC-Press.

Barton, M. A., & Lord, F. M. (1981). An Upper Asymptote for the Three-Parameter Logistic Item-Response Model.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In Lord, F.M. & Novick, M.R. (Eds.), Statistical theories of mental test scores. Reading, MA: Addison-Wesley.

Magis, D. (2013). A note on the item information function of the four-parameter logistic model. Applied Psychological Measurement, 37(4), 304-315.

Samejima, Fumiko (1993). The bias function of the maximum likelihood estimate of ability for the dichotomous response level. Psychometrika, 58, 195-209.

Samejima, Fumiko (1993). An approximation of the bias function of the maximum likelihood estimate of a latent variable for the general case where the item responses are discrete. Psychometrika, 58, 119-138.

Schuster, C., & Yuan, K. H. (2011). Robust estimation of latent ability in item response models. Journal of Educational and Behavioral Statistics, 36(6), 720-735.

Warm, Thomas A. (1989). Weighted Likelihood Estimation Of Ability In Item Response Theory. Psychometrika, 54, 427-450.

Yen, Y.-C., Ho, R.-G., Liao, W.-W., Chen, L.-J., & Kuo, C.-C. (2012). An empirical evaluation of the slip correction in the four parameter logistic models with computerized adaptive testing. Applied Psychological Measurement, 36, 75-87.

See Also

PPass, PPall, PP_gpcm, JKpp, PV

Examples

```
## real data: an adaptive intelligence test with known item difficulties (1PL)
data(pp_amt)
d <- as.matrix(pp_amt$daten_amt[, -(1:7)])
res <- PP_4pl(respm = d, thres = pp_amt$betas[, 2], type = "wle")
summary(res)

## simulated data: 2PL, 3PL and 4PL
set.seed(1522)
b  <- seq(-3, 3, length = 12)
sl <- round(runif(12, 0.5, 1.5), 2)
la <- round(runif(12, 0, 0.25), 2)
ua <- round(runif(12, 0.8, 1), 2)
x  <- sim_4pl(beta = b, alpha = sl, lowerA = la, upperA = ua, theta = rnorm(50))

res2pl <- PP_4pl(respm = x, thres = b, slopes = sl, type = "mle")
res3pl <- PP_4pl(respm = x, thres = b, slopes = sl, lowerA = la, type = "wle")
res4pl <- PP_4pl(respm = x, thres = b, slopes = sl, lowerA = la, upperA = ua, type = "map")

# persons with all answers right (or wrong) have no finite ML estimate
table(res2pl$estimates$status)
head(res3pl$estimates)
coef(res4pl)[1:5]

## EAP with person-specific priors
res_eap <- PP_4pl(respm = x, thres = b, slopes = sl, type = "eap",
```

```
                    mu = rep(c(-0.5, 0.5), 25), sigma2 = 0.8)

## robust estimation: the example of Schuster & Yuan (2011)
AWM <- matrix(0, 7, 7)
diag(AWM) <- 1
coef(PP_4pl(respm = AWM, thres = -3:3, type = "mle"))
coef(PP_4pl(respm = AWM, thres = -3:3, type = "robust"))
```

---

pp_amt *Adaptive Matrices Test data*

Description

This dataset contains real data from the 'Adaptive Matrices Test' (AMT), which is a computer-administered test. This power test assesses logical reasoning as an indicator of general intelligence. The ability to identify regularities and draw logical conclusions is a very good predictor of long-term success at work. The dataset is sparse, because the test tailors a specific set of items for each examinee's ability level. (More information about adaptive testing: https://en.wikipedia.org/wiki/Computerized_adaptive_testing)

Format

A list with two data.frames. The first data.frame 'daten_amt' contains 298 columns and 710 rows. Each row contains responses from one examinee. The second data.frame 'betas' contains the difficulty parameter (1PL) (These parameters came with the raw-score extraction.).

Details

The data are provided from the University of Vienna, Faculty of Psychology, Department of Psychological Assessment. Thanks to Schuhfried.

Source

Division of Psychological Assessment and Applied Psychometrics, Faculty of Psychology, University of Vienna

- ID: id of person
- AGE: age in years (with ages below 18 and above 34 are collapsed)
- TE_GA:
  - TE: self-assessment. To pass a psychological assessment course, the students have to complete several hours self assessment on a bunch of tests, to get familiar with them.
  - GA: testing for an assessment report. The students also have to test other people (not psychologists nor psychology students) in order to write an assessment report.
- FORM: There are several different versions of this test, which differ in test length, duration etc ...
- TIME1: start time

- TIME2: end time
- REL: reliability for each person
- i: items

References

- Hornke, L. F., Etzel, S., & Rettig, K. (2003). Manual Adaptive Matrices Test (AMT). *Moedling: SCHUHFRIED GmbH*.

See Also

PPass

`PP_gpcm` *Estimate Person Parameters for the GPCM*

Description

Compute person parameters for the GPCM and choose between five common estimation techniques: ML, WL, MAP, EAP and a robust estimation. All item parameters are treated as fixed.

Usage

```
PP_gpcm(
  respm,
  thres,
  slopes = NULL,
  theta_start = NULL,
  mu = NULL,
  sigma2 = NULL,
  type = "wle",
  maxsteps = 100,
  exac = 0.001,
  H = 1,
  ctrl = list(),
  range = c(-6, 6),
  verbose = FALSE
)
```

Arguments

`respm`	An integer matrix (or a data.frame), which contains the responses of the examinees. A persons x items matrix is expected. Missing responses are `NA`.
`thres`	A numeric matrix which contains the threshold parameter for each item. If the first row of the matrix is not set to zero (only zeroes in the first row) - then a row-vector with zeroes is added by default.

`slopes` A numeric vector, which contains the slope parameters for each item - one parameter per item is expected. A single value is used for all items. If `NULL` (default), all slopes are 1.

`theta_start` A vector which contains a starting value for each person. If NULL is submitted, the starting values are set automatically. If a scalar is submitted, this start value is used for each person.

`mu` A numeric vector of location parameters for each person, the normal prior for MAP and EAP estimation and for `PV`. A single value is used for all persons. If nothing is submitted this is set to 0 for each person. MLE, WLE and robust estimation have no prior: a value given there is ignored with a warning, and the result stores `NA`.

`sigma2` A numeric vector of variance parameters for each person, the normal prior for MAP and EAP estimation and for `PV`. A single value is used for all persons. If nothing is submitted this is set to 1 for each person. MLE, WLE and robust estimation have no prior: a value given there is ignored with a warning, and the result stores `NA`.

`type` Which maximization should be applied? There are five valid entries possible: "mle", "wle", "map", "eap" and "robust". To choose between the methods, or just to get a deeper understanding the papers mentioned below are quite helpful. The default is `"wle"` which is a good choice in many cases. With `"eap"` the estimate is the mean and the SE the standard deviation of the posterior, computed on a grid of ability values (see PV). Plausible values (`PV`) need the prior of `"eap"` or `"map"`.

`maxsteps` The maximum number of Newton steps per person. Default = 100.

`exac` How accurate are the estimates supposed to be? The iteration of a person stops when the step is smaller than `exac`. Default is 0.001.

`H` In case `type = "robust"` a Huber ability estimate is performed, and `H` modulates how fast the downweighting takes place (for more Details read Schuster & Yuan 2011).

`ctrl` No longer used. Its options `killdupli` and `skipcheck` of version 0.6.4 saved time with the old estimation routine; since 1.0.0 every person is estimated separately, and options given are ignored with a warning.

`range` The interval `c(lower, upper)` in which mle, wle, map and robust estimates are searched. If the estimating equation has no root inside, the estimate is set to the bound it points to and its status is `"boundary"`. Default `c(-6, 6)`.

`verbose` If `TRUE`, a message names the model and the estimation type. Default `FALSE`.

Details

Please note, that `robust` estimation with (Huber ability estimate) polytomous items is still experimental!

The probability choosing the k-th category is as follows:

$$P(x_{ij} = k|\hat{\alpha}_i, \hat{\beta}_{iv}, \theta_j) = \frac{exp(\sum_{v=0}^{(k-1)} \hat{\alpha}_i(\theta_j - \hat{\beta}_{iv}))}{\sum_{c=0}^{m_i-1} exp(\sum_{v=0}^{c} \hat{\alpha}_i(\theta_j - \hat{\beta}_{iv})))}$$

In our case θ is to be estimated. The item parameters are assumed as fixed (usually these are estimates of a former scaling procedure).

The model simplifies to the Partial Credit Model by setting $\alpha_i = 1, \forall i$.

Value

An object of class `ppeo` (and `fourpl`, `gpcm` or `gpcm4pl`), a list with

`estimates` A data.frame with one row per person, in the order of `respm`: `estimate`, `SE`, `nsteps` (Newton steps of the person; 0 for eap) and `status`, a factor with the levels

- `"ok"`: converged inside `range`,
- `"extreme"`: mle or robust estimate of a person with all answers in the lowest or highest category, the estimate is `-Inf` or `Inf` and the SE `NA`,
- `"boundary"`: the estimating equation has no root inside `range`, the estimate is the bound,
- `"no_response"`: the person answered no item, estimate and SE are `NA` (eap: prior mean and SD),
- `"not_converged"`: no convergence within `maxsteps`.

A warning reports persons with status `"boundary"` or `"not_converged"`.

`type`, `call` The estimation type and the call.

`model` The checked and prepared inputs.

`resPP` The result in the layout of earlier versions: `resPP$resPP` is the persons x 2 matrix of estimates and SEs, `resPP$nsteps` the largest number of steps.

Use `coef()` for the estimates, `as.data.frame()` for the table and `plot()` for the measurement precision, the person fit map or person profiles (see pp-plots).

Author(s)

Manuel Reif

References

Baker, Frank B., and Kim, Seock-Ho (2004). Item Response Theory - Parameter Estimation Techniques. CRC-Press.

Masters, G. N. (1982). A Rasch model for partial credit scoring. Psychometrika, 47(2), 149-174.

Muraki, Eiji (1992). A Generalized Partial Credit Model: Application of an EM Algorithm. Applied Psychological Measurement, 16, 159-176.

Muraki, Eiji (1993). Information Functions of the Generalized Partial Credit Model. Applied Psychological Measurement, 17, 351-363.

Samejima, Fumiko (1993). The bias function of the maximum likelihood estimate of ability for the dichotomous response level. Psychometrika, 58, 195-209.

Samejima, Fumiko (1993). An approximation of the bias function of the maximum likelihood estimate of a latent variable for the general case where the item responses are discrete. Psychometrika, 58, 119-138.

Schuster, C., & Yuan, K. H. (2011). Robust estimation of latent ability in item response models. Journal of Educational and Behavioral Statistics, 36(6), 720-735.

Wang, S. and Wang, T. (2001). Precision of Warm's Weighted Likelihood Estimates for a Polytomous Model in Computerized Adaptive Testing. Applied Psychological Measurement, 25, 317-331.

Warm, Thomas A. (1989). Weighted Likelihood Estimation Of Ability In Item Response Theory. Psychometrika, 54, 427-450.

See Also

PPass, PPall, PP_4pl, JKpp, PV

Examples

```
# six items with three categories: one column of thresholds per item
thres <- matrix(c(-2, -1.23, 1.11, 3.48, 1, 2, -1, -0.2, 0.5, 1.3, -0.8, 1.5), nrow = 2)
sl <- c(0.5, 1, 1.5, 1.1, 1, 0.98)
set.seed(1750)
x <- sim_gpcm(thres = thres, alpha = sl, theta = rnorm(30))

# GPCM, weighted likelihood
resgpcm <- PP_gpcm(respm = x, thres = thres, slopes = sl, type = "wle")
summary(resgpcm)

# partial credit model: all slopes 1 (the default), EAP
respcm <- PP_gpcm(respm = x, thres = thres, type = "eap")
head(respcm$estimates)

# items with a different number of categories: NA below the last threshold
thres4 <- rbind(thres, c(NA, NA, NA, NA, 1.7, 1))
x4 <- sim_gpcm(thres = thres4, alpha = sl, theta = rnorm(30))
head(PP_gpcm(respm = x4, thres = thres4, slopes = sl, type = "map")$estimates)
```

`print.pv` *Draw Plausible values*

Description

This function draws npv plausible values for each person from their posterior density.

Usage

```
## S3 method for class 'pv'
print(x, digits = 4, ...)

## S3 method for class 'pv'
summary(object, nrowmax = 15, digits = 4, ...)

## S3 method for class 'ppass'
PV(estobj, ...)

PV(estobj, ...)
```

```
## S3 method for class 'ppeo'
PV(estobj, npv = 10, approx = TRUE, ...)
```

Arguments

x, object	An object of class pv, the result of PV().
digits	Number of significant digits to print.
...	Not used. thinning, burnin and mult of version 0.6.4 give a warning and are ignored.
nrowmax	summary(): how many persons to show. Default = 15.
estobj	An object which originates from using PP_4pl(), PP_gpcm() or PPall() with type = "eap" or type = "map".
npv	The number of plausible values per person - default is 10.
approx	If TRUE (default), the plausible values are drawn from a normal approximation of the posterior. If FALSE, they are drawn from the posterior itself.

Details

Plausible values need a prior, so estobj must come from an estimation with type = "eap" or type = "map". The posterior of each person combines the likelihood of the answered items with the normal prior N(mu, sigma2) of the person, as given to PP_4pl(), PP_gpcm() or PPall() (default N(0, 1)). It is computed on a grid of ability values, the same as for type = "eap": one grid for all persons, and a finer grid around the posterior mode for persons whose posterior is very narrow or lies outside the range of the prior. Both types give the same plausible values, since they share the posterior; the point estimate itself is not used. MLE, WLE and robust estimates have no prior, and PV() stops with an error for them. Missing responses are skipped; a person without any response gets draws from the prior.

With approx = FALSE each value is drawn from the posterior on the grid: a grid point is chosen with its posterior probability, then the value is drawn uniformly within its cell. The draws are independent. With approx = TRUE (default) the draws come from the normal distribution with the posterior mean and the posterior SD of the person (the EAP estimate and its SE).

Up to version 0.6.4 PV() accepted any estimation type, approx = FALSE used a Metropolis-Hastings algorithm and approx = TRUE drew from the normal distribution around the estimate of estobj with its SE. For an MLE or WLE estimate this is not a posterior and the plausible values were too widely spread. The arguments thinning, burnin and mult of the sampler are no longer used.

Value

An object of class pv, a list with pvdraws, a matrix with one row per person (in the order of respm) and npv columns, and call.

Author(s)

Manuel Reif

References

Mislevy, R. J. (1991). Randomization-based inference about latent variables from complex samples. Psychometrika, 56(2), 177-196.

Von Davier, M., Gonzalez, E., & Mislevy, R. (2009). What are plausible values and why are they useful. IERI monograph series, 2, 9-36.

See Also

PP_gpcm, PP_4pl, PPall, JKpp

Examples

```
## 2PL model

set.seed(1522)
# difficulties and slopes
diffpar <- seq(-3, 3, length = 12)
sl      <- round(runif(12, 0.5, 1.5), 2)

# response matrix
awm <- sim_4pl(beta = diffpar, alpha = sl, lowerA = rep(0, 12), upperA = rep(1, 12),
               theta = rnorm(10))

# plausible values need a prior: EAP (or MAP) estimation, here with the default N(0, 1)
res2pl <- PP_4pl(respm = awm, thres = diffpar, slopes = sl, type = "eap")

# 10 plausible values per person, normal approximation of the posterior
res_pv <- PV(res2pl)
summary(res_pv)

# 200 draws from the posterior itself; their means approximate the EAP
res_pv2 <- PV(res2pl, npv = 200, approx = FALSE)
cbind(pv_mean = rowMeans(res_pv2$pvdraws),
      eap = res2pl$estimates$estimate)


## GPCM model with person-specific priors

THRES  <- matrix(c(-2,-1.23,1.11,3.48,1
                   ,2,-1,-0.2,0.5,1.3,-0.8,1.5),nrow=2)
sl     <- c(0.5,1,1.5,1.1,1,0.98)
awmatrix <- matrix(c(1,0,2,0,1,1,1,0,0,1
                     ,2,0,0,0,0,0,0,0,0,1,1,2,2,1,1,1,1,0,0,1),byrow=TRUE,nrow=5)

resgpcm <- PP_gpcm(respm = awmatrix, thres = THRES, slopes = sl, type = "eap",
                   mu = c(0, 0, 1, 1, 1), sigma2 = 0.5)
PV(resgpcm, npv = 5, approx = FALSE)


## GPCM and 4PL items mixed
```

```
THRESx <- THRES
THRESx[2,1:3] <- NA
la     <- c(0.02,0.1,0,0,0,0)
ua     <- c(0.97,0.91,1,1,1,1)
awmatrix <- matrix(c(1,0,1,0,1,1,1,0,0,1
                     ,2,0,0,0,0,0,0,0,0,1
                     ,1,2,2,1,1,1,1,0,0,1),byrow=TRUE,nrow=5)

respcm <- PPall(respm = awmatrix, thres = THRESx, slopes = sl, lowerA = la, upperA = ua,
                type = "eap")
res_mixedpv <- PV(respcm, npv = 200, approx = FALSE)

# quantiles of the draws per person
apply(res_mixedpv$pvdraws, 1, quantile)
```

`sim_4pl` *Simulate data for 1/2/3/4-pl model*

Description

This function returns a dichotomous matrix of simulated responses under given item and person parameters.

Usage

```
sim_4pl(beta, alpha, lowerA, upperA, theta)
```

Arguments

`beta`	A numeric vector which contains the difficulty parameters for each item.
`alpha`	A numeric vector, which contains the slope parameters for each item.
`lowerA`	A numeric vector, which contains the lower asymptote parameters (kind of guessing parameter) for each item.
`upperA`	numeric vector, which contains the upper asymptote parameters for each item.
`theta`	A numeric vector which contains person parameters.

Details

`alpha`, `lowerA` and `upperA` may also be single numbers, which are then used for every item. A person with `NA` as person parameter gets a row of `NA`.

Value

An integer matrix with one row per person and one column per item.

Author(s)

Manuel Reif

See Also

sim_gpcm, sim_mixed, PP_4pl

Examples

```
set.seed(1700)
theta <- rnorm(200)
b <- seq(-2, 2, length.out = 7)

# 1PL (Rasch) model: slopes 1, no asymptotes
x1 <- sim_4pl(beta = b, alpha = 1, lowerA = 0, upperA = 1, theta = theta)
colMeans(x1)

# 4PL model
x4 <- sim_4pl(beta = b, alpha = c(1, 1.1, 0.9, 0.9, 1.5, 1.1, 1),
              lowerA = c(0, 0, 0.2, 0.15, 0.04, 0, 0.21),
              upperA = c(0.9, 0.98, 0.97, 1, 1, 1, 0.97), theta = theta)
head(x4)
```

---

sim_gpcm *Simulate data for the gpcm model*

Description

This function returns an integer matrix of simulated responses under given item and person parameters.

Usage

```
sim_gpcm(thres, alpha, theta)
```

Arguments

thres	A numeric matrix which contains threshold parameters for each item. The first row may contain zeroes only (as in PP_gpcm); otherwise a row of zeroes is added.
alpha	A numeric vector, which contains the slope parameters - one parameter per item is expected.
theta	A numeric vector which contains person parameters.

Details

alpha may also be a single number, which is then used for every item. A person with NA as person parameter gets a row of NA.

Value

An integer matrix with one row per person and one column per item.

See Also

sim_4pl, sim_mixed, PP_gpcm

Examples

```
set.seed(1750)
# thresholds of six items with three categories (the row of zeroes is optional)
thres <- rbind(0, matrix(c(-2, -1.23, 1.11, 3.48, 1, 2, -1, -0.2, 0.5, 1.3, -0.8, 1.5),
                         nrow = 2))
x <- sim_gpcm(thres = thres, alpha = c(0.5, 1, 1.5, 1.1, 1, 0.98), theta = rnorm(100))
head(x)

# the last two items get a fourth category
thres4 <- rbind(thres, c(NA, NA, NA, NA, 1.7, 1))
apply(sim_gpcm(thres = thres4, alpha = 1, theta = rnorm(100)), 2, max)
```

---

sim_mixed *Simulate data for mixed tests (4PL and GPCM items)*

Description

This function returns an integer matrix of simulated responses for a test with dichotomous (1/2/3/4-PL) and polytomous (GPCM) items, with the item parameters in the layout of PPall.

Usage

```
sim_mixed(thres, alpha = NULL, lowerA = NULL, upperA = NULL, theta)
```

Arguments

thres	A numeric matrix with the thresholds of each item in its columns, as in PPall: for a 4PL item the difficulty, for a GPCM item its thresholds, NA below the last one. Items with one threshold are 4PL items, the others GPCM items. The first row may contain zeroes only; otherwise a row of zeroes is added.
alpha	A numeric vector of slope parameters. If NULL (default), all slopes are 1.
lowerA	A numeric vector of lower asymptotes (4PL items). If NULL (default), 0.
upperA	A numeric vector of upper asymptotes (4PL items). If NULL (default), 1.
theta	A numeric vector which contains person parameters.

Details

alpha, lowerA and upperA may also be single numbers, which are then used for every item. lowerA and upperA only apply to the 4PL items. A person with NA as person parameter gets a row of NA. The data of a pure 4PL or GPCM test are the same as those of sim_4pl or sim_gpcm with the same seed.

Value

An integer matrix with one row per person and one column per item.

See Also

sim_4pl, sim_gpcm, PPall

Examples

```
set.seed(1800)
# four 3PL items and two GPCM items with three categories
thres <- rbind(0, c(-1, -0.3, 0.4, 1.2, -0.5, 0.2), c(NA, NA, NA, NA, 0.8, 1.4))
x <- sim_mixed(thres = thres, alpha = c(1, 1.2, 0.8, 1.1, 1, 0.9),
               lowerA = c(0.2, 0.1, 0.15, 0.2, 0, 0), theta = rnorm(300))
head(x)

# and back: estimate the persons with the same parameters
res <- PPall(x, thres = thres, slopes = c(1, 1.2, 0.8, 1.1, 1, 0.9),
             lowerA = c(0.2, 0.1, 0.15, 0.2, 0, 0))
summary(res$estimates$estimate)
```

Index

∗ **4pl**
PP_4pl, 22
∗ **GPCM**
PP_gpcm, 28
∗ **Infit-Outfit**
Pfit, 8
∗ **LZ-Index**
Pfit, 8
∗ **Parameters**
findmodel, 4
PP_4pl, 22
PP_gpcm, 28
∗ **Person**
findmodel, 4
Pfit, 8
PP_4pl, 22
PP_gpcm, 28
∗ **fit**
Pfit, 8

as.data.frame.ppass *(PPass)*, 18
as.data.frame.ppeo *(ppeo-methods)*, 21

coef.ppeo *(ppeo-methods)*, 21

findmodel, 4, *16*
fourpl_df, 5

JKpp, *2*, 5, *11*, *18*, *20*, *26*, *31*, *33*

Pfit, *2*, 8, *14*, *20*
plot.ppass *(pp-plots)*, 12
plot.ppeo *(pp-plots)*, 12
PP *(PP-package)*, 2
PP-package, 2
pp-plots, 12, *17*, *20–22*, *25*, *30*
PP_4pl, *2*, *8*, *11*, *14*, *18*, *20*, *22*, 22, *31*, *33*, *35*
pp_amt, 27
PP_gpcm, *2*, *8*, *11*, *14*, *18*, *20*, *22, 23*, *26*, 28, *33*, *35, 36*
PPall, *2*, *4*, *8*, *11*, *14*, 14, *20*, *22*, *26*, *31*, *33*, *36, 37*
PPass, *3*, *11*, *14*, *18*, 18, *26*, *28*, *31*
ppeo-methods, 21
print.jk *(JKpp)*, 5
print.ppass *(PPass)*, 18
print.ppeo *(ppeo-methods)*, 21
print.PPfit *(Pfit)*, 8
print.pv, 31
PV, *2*, *15, 16*, *18*, *20*, *23, 24*, *26*, *29*, *31*
PV *(print.pv)*, 31

sim_4pl, *3*, *5*, 34, *36, 37*
sim_gpcm, *3*, *35*, 35, *36, 37*
sim_mixed, *3*, *35, 36*, 36
summary.jk *(JKpp)*, 5
summary.ppass *(PPass)*, 18
summary.ppeo *(ppeo-methods)*, 21
summary.PPfit *(Pfit)*, 8
summary.pv *(print.pv)*, 31