Package 'eRm'

April 17, 2009

Type Package

Title Extended Rasch Modeling.

Version 0.10-2

Date 2009-03-05

Author Patrick Mair, Reinhold Hatzinger

Maintainer Patrick Mair <patrick.mair@wu-wien.ac.at>

Description eRm fits Rasch models (RM), linear logistic test models (LLTM), rating scale model (RSM), linear rating scale models (LRSM), partial credit models (PCM), and linear partial credit models (LPCM). Missing values are allowed in the data matrix. Additional features are the ML estimation of the person parameters, Andersen's LR-test, item-specific Wald test, itemfit and personfit statistics including infit and outfit measures, various ICC and related plots, automated stepwise item elimination, simulation module for various binary data matrices. An eRm platform is provided at R-forge (see URL).

License GPL

URL http://r-forge.r-project.org/projects/erm/

Imports graphics, stats, MASS

Depends R (>= 2.7.0), gtools, splines, ROCR

Revision 52

Repository CRAN

Date/Publication 2009-03-06 16:53:27

R topics documented:

`eRm-package` *extended Rasch modeling*

Description

This package estimates extended Rasch models, i.e. the ordinary Rasch model for dichotomous data (RM), the linear logistic test model (LLTM), the rating scale model (RSM) and its linear extension (LRSM), the partial credit model (PCM) and its linear extension (LPCM). The parameters are estimated by conditional maximum likelihood (CML). Missing values are allowed in the data matrix. Additional features are the estimation of the person parameters, LR-Model test, item-spefific Wald test, itemfit and personfit statistics, various ICC plots. An eRm platform is provided at http://r-forge.r-project.org/projects/erm/.

Details

Package:	eRm
Type:	Package
Version:	0.10-2
Date:	2009-03-05
License:	GPL

The basic input units for the functions are the person-item matrix X and the design matrix W. Missing values in X are coded with `NA`. By default, W is generated automatically, but it can be specified by the user as well. The function call of the basic models can be achieved through `RM(X,`

W), RSM(X, W), and PCM(X, W).

The linear extensions provide the possibility to fit a more restricted model than its basic complement, such as LLTM(X, W), LRSM(X, W),LPCM(X, W), but also a generalization by imposing repeated measurement designs and group contrasts. These models can be estimated by using, e.g.,
LLTM(X, W, mpoints = 2, groupvec = G),
LRSM(X, W, mpoints = 2, groupvec = G), and
LPCM(X, W, mpoints = 2, groupvec = G).
mpoints specifies the number of measurement or time points, G is a vector with the group membership for each subject ordered according to the rows of the data matrix.

RM produces an object belonging to the classes dRM, RM, and eRm. PCM and RSM produce objects belonging to the classes RM and eRm, whereas results of LLTM, LRSM, and LLTM are object of class eRm.

Author(s)

Patrick Mair, Reinhold Hatzinger

Maintainer: Patrick Mair <patrick.mair@wu-wien.ac.at>

References

Fischer, G. H., and Molenaar, I. (1995). Rasch Models - Foundations, Recent Developements, and Applications. Springer.

Mair, P., and Hatzinger, R. (2007). Extended Rasch modeling: The eRm package for the application of IRT models in R. Journal of Statistical Software, 20(9), 1-20.

Mair, P., and Hatzinger, R. (2007). CML based estimation of extended Rasch models with the eRm package in R. Psychology Science, 49, 26-43.

---

gofIRT *Various model tests and fit indices*

Description

This function computes various model tests and fit indices for objects of class ppar: Collapsed deviance, Casewise deviance, Rost's LR-test, Hosmer-Lemeshow test, R-Squared measures, confusion matrix, ROC analysis.

Usage

```
## S3 method for class 'ppar':
gofIRT(object, groups.hl = 10, cutpoint = 0.5)
```

Arguments

object	Object of class ppar (from person.parameter()).
groups.hl	Number of groups for Hosmer-Lemeshow test (see details).
cutpoint	Integer between 0 and 1 for computing the 0-1 model matrix from the estimated probabilities

Details

So far this test statistics are implemented only for dichotomous models without NA's. The Hosmer-Lemeshow test is computed by splitting the response vector into percentiles, e.g. `groups.hl = 10` corresponds to decile splitting.

Value

The function `gofIRT` returns an object of class `gof` containing:

`test.table`	Ouput for model tests.
`R2`	List with R-squared measures.
`classifier`	Confusion matrix, accuracy, sensitivity, specificity.
`AUC`	Area under ROC curve.
`Gini`	Gini coefficient.
`ROC`	FPR and TPR for different cutpoints.
`opt.cut`	Optimal cutpoint determined by ROC analysis.
`predobj`	Prediction output from ROC analysis (`ROCR` package)

References

Mair, P., Reise, S. P., and Bentler, P. M. (2008). IRT goodness-of-fit using approaches from logistic regression. UCLA Statistics Preprint Series.

See Also

`itemfit.ppar`,`personfit.ppar`,`LRtest`

Examples

```
#Goodness-of-fit for a Rasch model
data(raschdat1)
res <- RM(raschdat1)
pres <- person.parameter(res)
gof.res <- gofIRT(pres)
gof.res
summary(gof.res)
```

IC *Information criteria*

Description

Computation of information criteria such as AIC, BIC, and cAIC based on unconditional (joint), marginal, and conditional log-likelihood

Usage

```
## S3 method for class 'ppar':
IC(object)
```

Arguments

object	Object of class ppar (from person.parameter().

Details

The joint log-likelihood is established by summation of the logarithms of the estimated solving probabilities. The marginal log-likelihood can be computed directly from the conditional (see vignette for details).

Value

The function returns an object of class ICr containing:

ICtable	Matrix containing log-likelihood values, number of parameteres, IC's

See Also

LRtest.Rm

Examples

```
#IC's for Rasch model
data(raschdat2)
res <- RM(raschdat2)              #Rasch model
pres <- person.parameter(res)     #Person parameters
IC(pres)

#IC's for RSM
data(rsmdat)
res <- RSM(rsmdat)
pres <- person.parameter(res)
IC(pres)
```

`itemfit.ppar` *Residuals, Personfit and Itemfit Statistics*

Description

`pmat` computes the theoretical person-item matrix with solving probabilities for each category (except 0th). `residuals` computes the squared and standardized residuals based on the observed and the expected person-item matrix. Chi-square based itemfit and personfit statistics can be obtained by using `itemfit` and `personfit`.

Usage

```
## S3 method for class 'ppar':
pmat(object)
## S3 method for class 'ppar':
residuals(object,...)
## S3 method for class 'ppar':
itemfit(object)
## S3 method for class 'ppar':
personfit(object)
## S3 method for class 'ifit':
print(x, ...)
## S3 method for class 'pfit':
print(x, ...)
## S3 method for class 'resid':
print(x, ...)
```

Arguments

`object`	Object of class `ppar`, derived from `person.parameter`.
`x`	Object of class `ifit`, `pfit`, or `resid`.
`...`	Further arguments passed to or from other methods. They are ignored in this function.

Value

`pmat`	Matrix of theoretical probabilities for each category except 0th (from function `pmat`).
`i.fit`	Chi-squared itemfit statistics (from function `itemfit`).
`i.df`	Degrees of freedom for itemfit statistics (from function `itemfit`).
`st.res`	Standardized residuals (from function `itemfit`).
`i.outfitMSQ`	Outfit mean-square statistics (from function `itemfit`).
`i.infitMSQ`	Infit mean-square statistics (from function `itemfit`).
`p.fit`	Chi-squared personfit statistics (from function `personfit`).

p.df	Degrees of freedom for personfit statistics (from function personfit).
st.res	Standardized residuals (from function personfit).
p.outfitMSQ	Outfit mean-square statistics (from function personfit).
p.infitMSQ	Infit mean-square statistics (from function personfit).

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Smith Jr., E. V., and Smith, R. M. (2004). Introduction to Rasch Measurement. JAM press.

Wright, B.D., and Masters, G.N. Computation of OUTFIT and INFIT Statistics. Rasch Measurement Transactions, 1990, 3:4 p.84-5

See Also

person.parameter

Examples

```
# Rasch model, estimation of item and person parameters
data(raschdat2)
res <- RM(raschdat2)
p.res <- person.parameter(res)

# Matrix with expected probabilities and corresponding residuals
pmat(p.res)
residuals(p.res)

#Itemfit
itemfit(p.res)

#Personfit
personfit(p.res)
```

LLTM *Estimation of linear logistic test models*

Description

This function computes the parameter estimates of a linear logistic test model (LLTM) for binary item responses by using CML estimation.

Usage

```
LLTM(X, W, mpoints = 1, groupvec = 1, se = TRUE, sum0 = TRUE,
   etaStart)
```

Arguments

X	Input 0/1 data matrix or data frame; rows represent individuals (N in total), columns represent items. Missing values have to be inserted as NA.
W	Design matrix for the LLTM. If omitted, the function will compute W automatically.
mpoints	Number of measurement points.
groupvec	Vector of length N which determines the group membership of each subject, starting from 1. If groupvec=1, no group contrasts are imposed.
se	If TRUE, the standard errors are computed.
sum0	If TRUE, the parameters are normalized to sum-0 by specifying an appropriate W. If FALSE, the first parameter is restricted to 0.
etaStart	A vector of starting values for the eta parameters can be specified. If missing, the 0-vector is used.

Details

Through appropriate definition of W the LLTM can be viewed as a more parsimonous Rasch model, on the one hand, e.g. by imposing some cognitive base operations to solve the items. One the other hand, linear extensions of the Rasch model such as group comparisons and repeated measurement designs can be computed. If more than one measurement point is examined, the item responses for the 2nd, 3rd, etc. measurement point are added column-wise in X.

If W is user-defined, it is nevertheless necessary to specify mpoints and groupvec. It is important that first the time contrasts and then the group contrasts have to be imposed.

Available methods for LLTM-objects are:
print, coef, model.matrix, vcov,summary, logLik, person.parameters.

Value

Returns on object of class eRm containing:

loglik	Conditional log-likelihood.
iter	Number of iterations.
npar	Number of parameters.
convergence	See code output in nlm.
etapar	Estimated basic item parameters.
se.eta	Standard errors of the estimated basic parameters.
betapar	Estimated item (easiness) parameters.
se.beta	Standard errors of item parameters.
hessian	Hessian matrix if se = TRUE.

W	Design matrix.
X	Data matrix.
X01	Dichotomized data matrix.
groupvec	Group membership vector.
call	The matched call.

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Fischer, G. H., and Molenaar, I. (1995). Rasch Models - Foundations, Recent Developements, and Applications. Springer.

Mair, P., and Hatzinger, R. (2007). Extended Rasch modeling: The eRm package for the application of IRT models in R. Journal of Statistical Software, 20(9), 1-20.

Mair, P., and Hatzinger, R. (2007). CML based estimation of extended Rasch models with the eRm package in R. Psychology Science, 49, 26-43.

See Also

LRSM,LPCM

Examples

```
#LLTM for 2 measurement points
#100 persons, 2*15 items, W generated automatically
data(lltmdat1)
res1 <- LLTM(lltmdat1, mpoints = 2)
print(res1)
summary(res1)

#Reparameterized Rasch model as LLTM (more pasimonious)
data(lltmdat2)
W <- matrix(c(1,2,1,3,2,2,2,1,1,1),ncol=2)              #design matrix
res2 <- LLTM(lltmdat2, W = W)
print(res2)
summary(res2)
```

LPCM *Estimation of linear partial credit models*

Description

This function computes the parameter estimates of a linear partial credit model (LRSM) for polytomuous item responses by using CML estimation.

Usage

```
LPCM(X, W , mpoints = 1, groupvec = 1, se = TRUE, sum0 = TRUE,
   etaStart)
```

Arguments

X	Input data matrix or data frame; rows represent individuals (N in total), columns represent items. Missing values are inserted as NA.
W	Design matrix for the LPCM. If omitted, the function will compute W automatically.
mpoints	Number of measurement points.
groupvec	Vector of length N which determines the group membership of each subject, starting from 1
se	If TRUE, the standard errors are computed.
sum0	If TRUE, the parameters are normalized to sum-0 by specifying an appropriate W. If FALSE, the first parameter is restricted to 0.
etaStart	A vector of starting values for the eta parameters can be specified. If missing, the 0-vector is used.

Details

Through appropriate definition of W the LPCM can be viewed as a more parsimonous PCM, on the one hand, e.g. by imposing some cognitive base operations to solve the items. One the other hand, linear extensions of the Rasch model such as group comparisons and repeated measurement designs can be computed. If more than one measurement point is examined, the item responses for the 2nd, 3rd, etc. measurement point are added column-wise in X.

If W is user-defined, it is nevertheless necessary to specify mpoints and groupvec. It is important that first the time contrasts and then the group contrasts have to be imposed.

Available methods for LPCM-objects are:
print, coef, model.matrix, vcov,summary, logLik, person.parameters.

Value

Returns on object of class `eRm` containing:

`loglik`	Conditional log-likelihood.
`iter`	Number of iterations.
`npar`	Number of parameters.
`convergence`	See `code` output in `nlm`.
`etapar`	Estimated basic item parameters.
`se.eta`	Standard errors of the estimated basic item parameters.
`betapar`	Estimated item (easiness) parameters.
`se.beta`	Standard errors of item parameters.
`hessian`	Hessian matrix if `se = TRUE`.
`W`	Design matrix.
`X`	Data matrix.
`X01`	Dichotomized data matrix.
`groupvec`	Group membership vector.
`call`	The matched call.

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Fischer, G. H., and Molenaar, I. (1995). Rasch Models - Foundations, Recent Developements, and Applications. Springer.

Mair, P., and Hatzinger, R. (2007). Extended Rasch modeling: The eRm package for the application of IRT models in R. Journal of Statistical Software, 20(9), 1-20.

Mair, P., and Hatzinger, R. (2007). CML based estimation of extended Rasch models with the eRm package in R. Psychology Science, 49, 26-43.

See Also

`LRSM`,`LLTM`

Examples

```
#LPCM for two measurement points and two subject groups
#20 subjects, 2*3 items
data(lpcmdat)
G <- c(rep(1,10),rep(2,10))                    #group vector
res <- LPCM(lpcmdat, mpoints = 2, groupvec = G)
print(res)
summary(res)
```

LRSM *Estimation of linear rating scale models*

Description

This function computes the parameter estimates of a linear rating scale model (LRSM) for polytomuous item responses by using CML estimation.

Usage

```
LRSM(X, W , mpoints = 1, groupvec = 1, se = TRUE, sum0 = TRUE,
   etaStart)
```

Arguments

X	Input data matrix or data frame; rows represent individuals (N in total), columns represent items. Missing values are inserted as NA.
W	Design matrix for the LRSM. If omitted, the function will compute W automatically.
mpoints	Number of measurement points.
groupvec	Vector of length N which determines the group membership of each subject, starting from 1
se	If TRUE, the standard errors are computed.
sum0	If TRUE, the parameters are normalized to sum-0 by specifying an appropriate W. If FALSE, the first parameter is restricted to 0.
etaStart	A vector of starting values for the eta parameters can be specified. If missing, the 0-vector is used.

Details

Through appropriate definition of W the LRSM can be viewed as a more parsimonous RSM, on the one hand, e.g. by imposing some cognitive base operations to solve the items. One the other hand, linear extensions of the Rasch model such as group comparisons and repeated measurement designs can be computed. If more than one measurement point is examined, the item responses for the 2nd, 3rd, etc. measurement point are added column-wise in X.

If W is user-defined, it is nevertheless necessary to specify mpoints and groupvec. It is important that first the time contrasts and then the group contrasts have to be imposed.

Available methods for LRSM-objects are: print, coef, model.matrix, vcov,summary, logLik, person.parameters.

Value

Returns on object of class `eRm` containing:

`loglik`	Conditional log-likelihood.
`iter`	Number of iterations.
`npar`	Number of parameters.
`convergence`	See `code` output in `nlm`.
`etapar`	Estimated basic item parameters (item and category parameters).
`se.eta`	Standard errors of the estimated basic item parameters.
`betapar`	Estimated item (easiness) parameters.
`se.beta`	Standard errors of item parameters.
`hessian`	Hessian matrix if `se = TRUE`.
`W`	Design matrix.
`X`	Data matrix.
`X01`	Dichotomized data matrix.
`groupvec`	Group membership vector.
`call`	The matched call.

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Fischer, G. H., and Molenaar, I. (1995). Rasch Models - Foundations, Recent Developements, and Applications. Springer.

Mair, P., and Hatzinger, R. (2007). Extended Rasch modeling: The eRm package for the application of IRT models in R. Journal of Statistical Software, 20(9), 1-20.

Mair, P., and Hatzinger, R. (2007). CML based estimation of extended Rasch models with the eRm package in R. Psychology Science, 49, 26-43.

See Also

`LLTM`,`LPCM`

Examples

```
#LRSM for two measurement points
#20 subjects, 2*3 items, W generated automatically,
#first parameter set to 0, no standard errors computed.

data(lrsmdat)
res <- LRSM(lrsmdat, mpoints = 2, groupvec = 1, sum0 = FALSE, se = FALSE)
print(res)
```

LRtest *Computation of Andersen's LR-test.*

Description

This LR-test is based on subject subgroup splitting.

Usage

```
## S3 method for class 'Rm':
LRtest(object, splitcr = "median", se = FALSE)
## S3 method for class 'LR':
plotGOF(x, beta.subset = "all", main="Graphical Model Check",
   xlab = NULL, ylab = NULL, tlab = "item",
   ylim = c(-3, 3), xlim = c(-3, 3), type = "p", pos = "4",
   conf = NULL, ctrline = NULL, ...)
```

Arguments

object	Object of class Rm.
splitcr	Split criterion for subject raw score splitting. all.r corresponds to a full raw score split, median uses the median as split criterion, mean performs a mean-split. Optionally splitcr can also be a vector which assigns each person to a certain subgroup (e.g., following an external criterion). This vector can be numeric, character or a factor.
se	If TRUE standard errors for beta's are computed.
x	Object of class LR. Also used for visualizing the fit of single items.
beta.subset	If "all", all items are plotted. Otherwise numeric subset vector can be specified.
main	Main title of the plot.
xlab	Label on x-axis, default gives name of splitcr and level.
ylab	Label on y-axis, default gives name of splitcr and level.
tlab	Specification of item labels: "item" prints the item names, "number" gives integers corresponding to order of the beta parameters, if "none" no labels are printed. "identify" allows for an interactive labelling. Initially no labels are printed, after clicking close to an item point the corresponding label is added. The identification process is terminated by clicking the second button and selecting 'Stop' from the menu, or from the 'Stop' menu on the graphics window. For more information and basic operation see identify.
xlim	Limits on x-axis.
ylim	Limits on y-axis.
type	Plotting type.(see plot)
pos	Position of the item label (see text)

conf	for plotting confidence ellipses for the item parameters. If conf=NULL (the default) no ellipses are drawn. Otherwise, conf must be specified as a list with optional elements: gamma, is the confidence level (numeric), col and lty, colour and linetype (see par), and ia, logical, if the ellipses are to be drawn interactively (cf. tlab="identify" above). If conf is specified as a an empty list, the default values conf=list(gamma=0.95, col="red", lty="dashed", ia=FALSE) will be used. See example below. To use conf, the LR object x has to be generated using the option se=TRUE in LRtest().
ctrline	for plotting confidence bands (control lines, cf.eg.Wright and Stone, 1999). If ctrline=NULL (the default) no lines are drawn. Otherwise, ctrline must be specified as a list with optional elements: gamma, is the confidence level (numeric), col and lty, colour and linetype (see par). If ctrline is specified as ctrline=list(), the default values conf=list(gamma=0.95, col="blue", lty="solid") will be used. See examples below. To use ctrline, the LR object x has to be generated using the option se=TRUE in LRtest().
...	Additional parameters.

Details

If the data set contains missing values and mean or median is specified as splitcriterion, means or medians are calculated for each missing value subgroup and consequently used for raw score splitting.

When using interactive selection for both labelling of single points (tlab = "identify" and drawing confidence ellipses at certain points (ia = TRUE) then first all plotted points are labelled and afterwards all ellipses are generated. Both identification processes can be terminated by clicking the second (right) mouse button and selecting 'Stop' from the menu, or from the 'Stop' menu on the graphics window.

summary and print methods are available for objects of class LR.

Value

LRtest returns an object of class LR containing:

LR	LR-value.
df	Degrees of freedom of the test statistic.
Chisq	Chi-square value with corresponding df.
pvalue	P-value of the test.
likgroup	Log-likelihood values for the subgroups
betalist	List of beta parameters for the subgroups.
selist	List of standard errors of beta's.
etalist	List of eta parameters for the subgroups.
spl.gr	Names and levels for splitcr.
call	The matched call.

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Fischer, G. H., and Molenaar, I. (1995). Rasch Models - Foundations, Recent Developements, and Applications. Springer.

Mair, P., and Hatzinger, R. (2007). Extended Rasch modeling: The eRm package for the application of IRT models in R. Journal of Statistical Software, 20(9), 1-20.

Mair, P., and Hatzinger, R. (2007). CML based estimation of extended Rasch models with the eRm package in R. Psychology Science, 49, 26-43.

Wright, B.D., and Stone, M.H. (1999). Measurement essentials. Wide Range Inc., Wilmington. (http://www.rasch.org/measess/me-all.pdf 28Mb).

See Also

Waldtest

Examples

```
# LR-test on dichotomous Rasch model with user-defined split
splitvec <- sample(1:3, 100, replace = TRUE)
data(raschdat1)
res <- RM(raschdat1)
lrres <- LRtest(res, splitcr = splitvec)
lrres
summary(lrres)

## Not run:
# goodness-of-fit plot with interactive labelling of items
plotGOF(lrres, tlab = "identify")
## End(Not run)

# LR-test with mean split, standard errors for beta's
lrres2 <- LRtest(res, split = "mean", se = TRUE)

# goodness-of-fit plot
# additional 95 percent control line with user specified style
plotGOF(lrres2, ctrline=list(gamma=0.95, col="red", lty="dashed"))

# goodness-of-fit plot for items 1, 14, 24, and 25
# additional 95 percent confidence ellipses, default style
plotGOF(lrres2, beta.subset=c(14,25,24,1), conf=list())
```

PCM *Estimation of partial credit models*

Description

This function computes the parameter estimates of a partial credit model for polytomous item responses by using CML estimation.

Usage

```
PCM(X, W, se = TRUE, sum0 = TRUE, etaStart)
```

Arguments

X	Input data matrix or data frame with item responses (starting from 0); rows represent individuals, columns represent items. Missing values are inserted as NA.
W	Design matrix for the PCM. If omitted, the function will compute W automatically.
se	If TRUE, the standard errors are computed.
sum0	If TRUE, the parameters are normed to sum-0 by specifying an appropriate W. If FALSE, the first parameter is restricted to 0.
etaStart	A vector of starting values for the eta parameters can be specified. If missing, the 0-vector is used.

Details

Through specification in W, the parameters of the categories with 0 responses are set to 0 as well as the first category of the first item. Available methods for PCM-objects are:
print, coef, model.matrix, vcov, plot, summary, logLik, person.parameters, plotICC, LRtest.

Value

Returns an object of class Rm, eRm containing.

loglik	Conditional log-likelihood.
iter	Number of iterations.
npar	Number of parameters.
convergence	See code output in nlm.
etapar	Estimated basic item parameters.
se.eta	Standard errors of the estimated basic item parameters.
betapar	Estimated item-category (easiness) parameters.
se.beta	Standard errors of item parameters.

hessian	Hessian matrix if se = TRUE.
W	Design matrix.
X	Data matrix.
X01	Dichotomized data matrix.
call	The matched call.

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Fischer, G. H., and Molenaar, I. (1995). Rasch Models - Foundations, Recent Developements, and Applications. Springer.

Mair, P., and Hatzinger, R. (2007). Extended Rasch modeling: The eRm package for the application of IRT models in R. Journal of Statistical Software, 20(9), 1-20.

Mair, P., and Hatzinger, R. (2007). CML based estimation of extended Rasch models with the eRm package in R. Psychology Science, 49, 26-43.

See Also

RM,RSM,LRtest

Examples

```
##PCM with 10 subjects, 3 items
data(pcmdat)
res <- PCM(pcmdat)
res
summary(res)                #eta and beta parameters with CI
thresholds(res)             #threshold parameters
```

---

person.parameter *Estimation of Person Parameters*

Description

Maximum likelihood estimation of the person parameters with spline interpolation for non-observed and 0/full responses. Extraction of information criteria such as AIC, BIC, and cAIC based on unconditional log-likelihood.

Usage

```
## S3 method for class 'eRm':
person.parameter(object)
## S3 method for class 'ppar':
summary(object, ...)
## S3 method for class 'ppar':
print(x, ...)
## S3 method for class 'ppar':
summary(object, ...)
## S3 method for class 'ppar':
plot(x, xlab = "Person Raw Scores",
   ylab = "Person Parameters (Theta)", main = NULL, ...)
## S3 method for class 'ppar':
logLik(object, ...)
## S3 method for class 'ppar':
confint(object, parm, level = 0.95, ...)
```

Arguments

object	Object of class eRm in person.parameter and object of class ppar in IC.
x	Object of class ppar.
xlab	Label of the x-axis.
ylab	Label of the y-axis.
main	Title of the plot.
...	Further arguments to be passed to or from other methods. They are ignored in this function.
parm	Parameter specification (ignored).
level	Alpha-level.

Details

If the data set contains missing values, person parameters are estimated for each missing value subgroup.

Value

The function person.parameter returns an object of class ppar containing:

loglik	Log-likelihood of the collapsed data (for faster estimation persons with the same raw score are collapsed).
npar	Number of parameters.
niter	Number of iterations.
thetapar	Person parameter estimates.
se.theta	Standard errors of the person parameters.
hessian	Hessian matrix.

theta.table	Matrix with person parameters (ordered according to original data) including NA pattern group.
pers.ex	Indices with persons excluded due to 0/full raw score
X.ex	Data matrix with persons excluded
gmemb	NA group membership vector (0/full persons excluded)
j.loglik	Unconditional (joint) log-likelihood value.
AIC	AIC criterion.
BIC	BIC criterion.
cAIC	Consistent AIC criterion.

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Fischer, G. H., and Molenaar, I. (1995). Rasch Models - Foundations, Recent Developements, and Applications. Springer.

Mair, P., and Hatzinger, R. (2007). Extended Rasch modeling: The eRm package for the application of IRT models in R. Journal of Statistical Software, 20(9), 1-20.

Mair, P., and Hatzinger, R. (2007). CML based estimation of extended Rasch models with the eRm package in R. Psychology Science, 49, 26-43.

See Also

itemfit.ppar,personfit.ppar

Examples

```
#Person parameter estimation of a rating scale model
data(rsmdat)
res <- RSM(rsmdat)
pres <- person.parameter(res)
print(pres)
summary(pres)
plot(pres)

#Person parameter estimation for a Rasch model with missing values
data(raschdat2)
res <- RM(raschdat2, se = FALSE) #Rasch model without standard errors
pres <- person.parameter(res)
print(pres)                      #person parameters
summary(pres)
logLik(pres)                     #log-likelihood of person parameter estimation
```

plotICC *ICC and Goodness-of-Fit Plots*

Description

Various plot functions for visualizing the item characteristic curves

Usage

```
## S3 method for class 'Rm':
plotICC(object, item.subset = "all", empICC = NULL, empCI = NULL,
   mplot = NULL, xlim = c(-4, 4), ylim = c(0, 1),
   xlab = "Latent Dimension", ylab = "Probability to Solve",
   col = NA, lty = 1, legpos = "left", ask = TRUE, ...)
## S3 method for class 'dRm':
plotjointICC(object, item.subset = "all", legend = TRUE,
   xlim = c(-4, 4), ylim = c(0, 1), xlab = "Latent Dimension",
   ylab = "Probability to Solve", lty = 1, legpos = "left",...)
```

Arguments

object	object of class Rm or dRm
item.subset	Subset of items to be plotted. Either a numeric vector indicating the column in X or a character vector indiciating the column name. If "all" (default), all items are plotted.
empICC	Plotting the empirical ICCs for objects of class dRm. If empICC=NULL (the default) the empirical ICC is not drawn. Otherwise, empICC must be specified as a list where the first element must be one of "raw", "loess", "tukey", "kernel". The other optional elements are smooth (numeric), type (line type for empirical ICCs, useful values are "p" (default), "l", and "b", see graphics parameter type in plot.default), pch, col, and lty, plotting 'character', colour and linetype (see par). See details and examples below.
empCI	Plotting confidence intervals for the the empirical ICCs. If empCI=NULL (the default) no confidence intervals are drawn. Otherwise, by specifying empCI as a list gives 'exact' confidence intervals for each point of the empirical ICC. The optional elements of this list are gamma, the confidence level, col, colour, and lty, line type. If empCI is specified as an empty list, the default values empCI=list(gamma=0.95,col="red",lty="dotted") will be used.
mplot	if NULL the default setting is in effect. For models of class dRm this is mplot = TRUE, i.e., the ICCs for up to 4 items are plotted in one figure. For Rm models the default is FALSE (each item in one figure) but may be set to TRUE.
xlab	Label of the x-axis.
ylab	Label of the y-axis.
xlim	Range of person parameters.

ylim	Range for probability to solve.
legend	If TRUE, legend is provided, otherwise the ICCs are labeled.
col	If NA, line color is determined automatically.
lty	Line type.
legpos	Position of the legend with possible values "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center". If FALSE no legend is displayed.
ask	If TRUE (the default) and the R session is interactive the user is asked for input, before a new figure is drawn. FALSE is only useful if automated figure export is in effect, e.g., when using Sweave.
...	Additional plot parameters.

Details

Empirical ICCs for objects of class dRm can be plotted using the option empICC, a list where the first element specifies the type of calculation of the empirical values. If empICC=list("raw", other specifications) relative frequencies of the positive responses are calculated for each rawscore group and plotted at the position of the corresponding person parameter. The other options use the default versions of various smoothers: "tukey" (see smooth), "loess" (see loess), and "kernel" (see ksmooth). For "loess" and "kernel" a further element, smooth, may be specified to control the span (default is 0.75) or the bandwith (default is 0.5), respectively. For example, the specification could be empirical = list("loess", smooth=0.9) or empirical = list("kernel",smooth=2). Higher values result in smoother estimates of the empirical ICCs.

The optional confidence intervals are obtained by a procedure first given in Clopper and Pearson (1934) based on the beta distribution (see binom.test).

Author(s)

Patrick Mair, Reinhold Hatzinger

See Also

plotGOF

Examples

```
# Rating scale model, ICC plot for all items
data(rsmdat)
rsm.res <- RSM(rsmdat)
thresholds(rsm.res)
plotICC(rsm.res)

# now items 1 to 4 in one figure without legends
plotICC(rsm.res, item.subset = 1:4, mplot = TRUE, legpos = FALSE)

# Rasch model for items 1 to 8 from raschdat1
# empirical ICCs displaying relative frequencies (default settings)
```

```
data(raschdat1)
rm8.res <- RM(raschdat1[,1:8])
plotICC(rm8.res, empICC=list("raw"))

# the same but using different plotting styles
plotICC(rm8.res, empICC=list("raw",type="b",col="blue",lty="dotted"))

# kernel-smoothed empirical ICCs using bandwidth = 2
plotICC(rm8.res, empICC = list("kernel",smooth=3))

# raw empirical ICCs with confidence intervals
# displaying only items 2,3,7,8
plotICC(rm8.res, item.subset=c(2,3,7,8), empICC=list("raw"), empCI=list())

# Joint ICC plot for items 2, 6, 8, and 15 for a Rasch model
data(raschdat1)
res <- RM(raschdat1)
plotjointICC(res, item.subset = c(2,6,8,15), legpos = "left")
```

`plotPImap` *Person-Item Map*

Description

A person-item map displays the location of item (and threshold) parameters as well as the distribution of person parameters.along the latent dimension. Person-item maps are useful to compare the range and position of the item measure distribution (lower panel) to the range and position of the person measure distribution (upper panel). Items should ideally be located along the whole scale to meaningfully measure the 'ability' of all persons.

Usage

```
plotPImap(object, item.subset = "all", sorted = FALSE,
   main = "Person-Item Map", latdim = "Latent Dimension",
   pplabel = "Person\nParameter\nDistribution", cex.gen = 0.7,
   xrange = NULL, warn.ord = TRUE)
```

Arguments

`object`	Object of class `Rm` or `dRm`
`item.subset`	Subset of items to be plotted. Either a numeric vector indicating the column in `X` or a character vector indicating the column name. If `"all"`, all items are plotted. The number of items to be plotted must be > 1.
`sorted`	If `TRUE`, the items are sorted in increasing order according to their location on the latent dimension.
`main`	Main title of the plot.
`latdim`	Label of the x-axis, i.e., the latent dimension.

pplabel	Title for the upper panel displaying the person parameter distribution
cex.gen	cex as a graphical parameter specifies a numerical value giving the amount by which plotting text and symbols should be magnified relative to the default. Here cex.gen applies to all text labels. The default is 0.7.
xrange	Range for the x-axis
warn.ord	If TRUE (the deault) asterisks are displayed in the right margin of the lower panel to indicate nonordinal threshold locations for polytomous items.

Details

Item locations are displayed with bullets, threshold locations with circles.

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Bond, T.G., and Fox Ch.M. (2007) Applying the Rasch Model. Fundamental Measurement in the Human Sciences. 2nd Edition. Lawrence Erlbaum Associates.

Examples

```
data(pcmdat)
res<-PCM(pcmdat)
plotPImap(res, sorted=TRUE)
```

---

predict.ppar *Predict methods*

Description

Returns data matrix based on model probabilites. So far implemented for dichotomous models only.

Usage

```
## S3 method for class 'ppar':
predict(object, cutpoint = "randomized", ...)
```

Arguments

object	Object of class ppar (from person.parameter()).
cutpoint	Either single integer value between 0 and 1 or "randomized" for randomized 0-1 assignment (see details)
...	Additional arguments ignored

Details

A randomized assignment implies that for each cell an additional random number is drawn. If the model probability is larger than this value, the person gets 1 on this particular item, if smaller, 0 is assigned. Alternatively, a numeric probability cutpoint can be assigned and the 0-1 scoring is carried out according to the same rule.

Value

Returns data matrix based on model probabilities

Author(s)

Patrick Mair, Reinhold Hatzinger

See Also

gofIRT.ppar

Examples

```
#Model-based data matrix for RSM
data(raschdat2)
res <- RM(raschdat2)
pres <- person.parameter(res)
predict(pres)
```

---

print.eRm *Methods for extended Rasch models*

Description

Several methods for objects of class eRm.

Usage

```
## S3 method for class 'eRm':
print(x, ...)
## S3 method for class 'eRm':
summary(object, ...)
## S3 method for class 'eRm':
coef(object, ...)
## S3 method for class 'eRm':
model.matrix(object, ...)
## S3 method for class 'eRm':
vcov(object, ...)
## S3 method for class 'eRm':
```

```
logLik(object, ...)
## S3 method for class 'eRm':
confint(object, parm, level = 0.95, ...)
```

Arguments

x	Object of class eRm.
object	Object of class eRm.
parm	Either "eta" or "beta".
level	Alpha-level.
...	Further arguments to be passed to or from other methods. They are ignored in this function.

Details

vcov returns the variance-covariance matrix of the parameter estimates, coef the estimates of the basic parameters, print the value of the log-likelihood and the parameter estimates and their standard errors, model.matrix the design matrix, logLik the log-likelihood value.

Author(s)

Patrick Mair, Reinhold Hatzinger

Examples

```
data(raschdat1)
res <- RM(raschdat1)
print(res)
summary(res)
coef(res)
vcov(res)
model.matrix(res)
```

---

raschdat1 *Data for Computing Extended Rasch Models*

Description

Artificial data sets for computing extended Rasch models.

Usage

```
data(raschdat1)
```

Format

Numeric matrices with subjects as rows, items as columns, missing values as NA.

Examples

```
data(raschdat1)
data(raschdat2)
data(lltmdat1)
data(lltmdat2)
data(pcmdat)
data(pcmdat2)
data(lpcmdat)
data(rsmdat)
data(lrsmdat)
```

RM *Estimation of Rasch Models*

Description

This function computes the parameter estimates of a Rasch model for binary item responses by using CML estimation.

Usage

```
RM(X, W, se = TRUE, sum0 = TRUE, etaStart)
```

Arguments

X	Input 0/1 data matrix or data frame; rows represent individuals, columns represent items. Missing values are inserted as NA.
W	Design matrix for the Rasch model. If omitted, the function will compute W automatically.
se	If TRUE, the standard errors are computed.
sum0	If TRUE, the parameters are normed to sum-0 by specifying an appropriate W. If FALSE, the first parameter is restricted to 0.
etaStart	A vector of starting values for the eta parameters can be specified. If missing, the 0-vector is used.

Details

For estimating the item parameters the CML method is used. Available methods for RM-objects are:
print, coef, model.matrix, vcov, summary, logLik, person.parameter, LRtest, Waldtest, plotICC, plotjointICC.

Value

Returns an object of class `dRm, Rm, eRm` and contains the log-likelihood value, the parameter estimates and their standard errors.

`loglik`	Conditional log-likelihood.
`iter`	Number of iterations.
`npar`	Number of parameters.
`convergence`	See `code` output in `nlm`.
`etapar`	Estimated basic item parameters.
`se.eta`	Standard errors of the estimated basic item parameters.
`betapar`	Estimated item (easiness) parameters.
`se.beta`	Standard errors of item parameters.
`hessian`	Hessian matrix if `se = TRUE`.
`W`	Design matrix.
`X`	Data matrix.
`X01`	Dichotomized data matrix.
`call`	The matched call.

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Fischer, G. H., and Molenaar, I. (1995). Rasch Models - Foundations, Recent Developements, and Applications. Springer.

Mair, P., and Hatzinger, R. (2007). Extended Rasch modeling: The eRm package for the application of IRT models in R. Journal of Statistical Software, 20(9), 1-20.

Mair, P., and Hatzinger, R. (2007). CML based estimation of extended Rasch models with the eRm package in R. Psychology Science, 49, 26-43.

See Also

`RSM`,`PCM`, `LRtest`, `Waldtest`

Examples

```
# Rasch model with beta.1 restricted to 0
data(raschdat1)
res <- RM(raschdat1, sum0 = FALSE)
print(res)
summary(res)
res$W                                    #generated design matrix
```

```
# Rasch model with sum-0 beta restriction; no standard errors computed
res <- RM(raschdat1, se = FALSE, sum0 = TRUE)
print(res)
summary(res)
res$W                                          #generated design matrix

#Rasch model with missing values
data(raschdat2)
res <- RM(raschdat2)
print(res)
summary(res)
```

RSM *Estimation of rating scale models*

Description

This function computes the parameter estimates of a rating scale model for polytomous item responses by using CML estimation.

Usage

```
RSM(X, W, se = TRUE, sum0 = TRUE, etaStart)
```

Arguments

X	Input data matrix or data frame with item responses (starting from 0); rows represent individuals, columns represent items. Missing values are inserted as NA.
W	Design matrix for the RSM. If omitted, the function will compute W automatically.
se	If TRUE, the standard errors are computed.
sum0	If TRUE, the parameters are normed to sum-0 by specifying an appropriate W. If FALSE, the first parameter is restricted to 0.
etaStart	A vector of starting values for the eta parameters can be specified. If missing, the 0-vector is used.

Details

The design matrix approach transforms the RSM into a partial credit model and estimates the corresponding basic parameters by using CML. Available methods for RSM-objects are `print`, `coef`, `model.matrix`, `vcov`, `summary`, `logLik`, `person.parameters`, `plotICC`, `LRtest`.

Value

Returns an object of class `Rm, eRm` and contains the log-likelihood value, the parameter estimates and their standard errors.

`loglik`	Conditional log-likelihood.
`iter`	Number of iterations.
`npar`	Number of parameters.
`convergence`	See `code` output in `nlm`.
`etapar`	Estimated basic item parameters (item and category parameters).
`se.eta`	Standard errors of the estimated basic item parameters.
`betapar`	Estimated item-category (easiness) parameters.
`se.beta`	Standard errors of item parameters.
`hessian`	Hessian matrix if `se = TRUE`.
`W`	Design matrix.
`X`	Data matrix.
`X01`	Dichotomized data matrix.
`call`	The matched call.

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Fischer, G. H., and Molenaar, I. (1995). Rasch Models - Foundations, Recent Developements, and Applications. Springer.

Mair, P., and Hatzinger, R. (2007). Extended Rasch modeling: The eRm package for the application of IRT models in R. Journal of Statistical Software, 20(9), 1-20.

Mair, P., and Hatzinger, R. (2007). CML based estimation of extended Rasch models with the eRm package in R. Psychology Science, 49, 26-43.

See Also

`RM`,`PCM`,`LRtest`

Examples

```
##RSM with 10 subjects, 3 items
data(rsmdat)
res <- RSM(rsmdat)
res
summary(res)                              #eta and beta parameters with CI
thresholds(res)                           #threshold parameters
```

`sim.2pl` *Simulation of 2-pl data*

Description

This utility function returns a 0-1 matrix violating the parallel ICC assumption in the Rasch model.

Usage

```
sim.2pl(persons, items, discrim = 0.25, seed = NULL,
   cutpoint = "randomized")
```

Arguments

`persons`	Either a vector of person parameters or an integer indicating the number of persons (see details).
`items`	Either a vector of item parameters or an integer indicating the number of items (see details).
`discrim`	Standard deviation on the log scale.
`seed`	A seed for the random number generated can be set.
`cutpoint`	Either `"randomized"` for a randomized tranformation of the model probability matrix into the model 0-1 matrix or an integer value between 0 and 1 (see details).

Details

If `persons` or `items` is an integer value, the corresponding parameter vector is drawn from N(0,1). The `cutpoint` argument refers to the transformation of the theoretical probabilities into a 0-1 data matrix. A randomized assingment implies that for each cell an additional random number is drawn. If the model probability is larger than this value, the person gets 1 on this particular item, if smaller, 0 is assigned. Alternatively, a numeric probability cutpoint can be assigned and the 0-1 scoring is carried out according to the same rule.

The `discrim` argument can be specified as a vector of length `items` indicating the item discrimination parameters in the 2-PL. If it is specified as integer, the discrimination parameters are drawn from a lognormal distribution with `meanlog = 0`. The degree of violation is steered by means of the specified value in `discrim` which refers to the standard deviation on the log-scale. The larger the values, the stronger the degree of Rasch violation. Values up to 0.5 should considered. If 0, the data are Rasch homogeneous.

References

Suárez-Falcón, J. C., & Glas, C. A. W. (2003). Evaluation of global testing procedures for item fit to the Rasch model. British Journal of Mathematical and Statistical Society, 56, 127-143.

See Also

`sim.rasch`, `sim.locdep`, `sim.xdim`

Examples

```
#simulating 2-PL data
#500 persons, 10 items, sdlog = 0.30, randomized cutpoint
X <- sim.2pl(500, 10, discrim = 0.30)

#item and discrimination parameters from uniform distribution,
#cutpoint fixed
dpar <- runif(50, 0, 2)
ipar <- runif(50, -1.5, 1.5)
X <- sim.2pl(500, ipar, dpar, cutpoint = 0.5)
```

`sim.locdep` *Simulation locally dependent items*

Description

This utility function returns a 0-1 matrix violating the local independence assumption.

Usage

```
sim.locdep(persons, items, it.cor = 0.25, seed = NULL,
   cutpoint = "randomized")
```

Arguments

`persons`	Either a vector of person parameters or an integer indicating the number of persons (see details).
`items`	Either a vector of item parameters or an integer indicating the number of items (see details).
`it.cor`	Either a single correlation value between 0 and 1 or a positive semi-definite VC matrix.
`seed`	A seed for the random number generated can be set.
`cutpoint`	Either `"randomized"` for a randomized tranformation of the model probability matrix into the model 0-1 matrix or an integer value between 0 and 1 (see details).

Details

If `persons` or `items` is an integer value, the corresponding parameter vector is drawn from N(0,1). The `cutpoint` argument refers to the transformation of the theoretical probabilities into a 0-1 data matrix. A randomized assingment implies that for each cell an additional random number is drawn. If the model probability is larger than this value, the person gets 1 on this particular item, if smaller, 0 is assigned. Alternatively, a numeric probability cutpoint can be assigned and the 0-1 scoring is carried out according to the same rule.

The argument `it.cor` reflects the pair-wise inter-item correlation. If this should be constant across the items, a single value between 0 (i.e. Rasch model) and 1 (strong violation) can be specified. Alternatively, a symmetric VC-matrix of dimension number of items can be defined.

References

Jannarone, R. J. (1986). Conjunctive item response theory kernels. Psychometrika, 51, 357-373.

Suárez-Falcón, J. C., & Glas, C. A. W. (2003). Evaluation of global testing procedures for item fit to the Rasch model. British Journal of Mathematical and Statistical Society, 56, 127-143.

See Also

sim.rasch, sim.2pl, sim.xdim

Examples

```
#simulating locally-dependent data
#500 persons, 10 items, inter-item correlation of 0.5
X <- sim.locdep(500, 10, it.cor = 0.5)

#500 persons, 4 items, correlation matrix specified
sigma <- matrix(c(1,0.2,0.2,0.3,0.2,1,0.4,0.1,0.2,0.4,1,0.8,0.3,0.1,0.8,1),
   ncol = 4)
X <- sim.locdep(500, 4, it.cor = sigma)
```

---

sim.rasch *Simulation of Rasch homogeneous data*

Description

This utility function returns a 0-1 matrix which fits the Rasch model.

Usage

```
sim.rasch(persons, items, seed = NULL, cutpoint = "randomized")
```

Arguments

persons	Either a vector of person parameters or an integer indicating the number of persons (see details)
items	Either a vector of item parameters or an integer indicating the number of items (see details)
seed	A seed for the random number generated can be set.
cutpoint	Either "randomized" for a randomized tranformation of the model probability matrix into the model 0-1 matrix or an integer value between 0 and 1 (see details)

Details

If `persons` or `items` is an integer value, the corresponding parameter vector is drawn from N(0,1). The `cutpoint` argument refers to the transformation of the theoretical probabilities into a 0-1 data matrix. A randomized assingment implies that for each cell an additional random number is drawn. If the model probability is larger than this value, the person gets 1 on this particular item, if smaller, 0 is assigned. Alternatively, a numeric probability cutpoint can be assigned and the 0-1 scoring is carried out according to the same rule.

References

Suárez-Falcón, J. C., & Glas, C. A. W. (2003). Evaluation of global testing procedures for item fit to the Rasch model. British Journal of Mathematical and Statistical Society, 56, 127-143.

See Also

`sim.xdim`, `sim.locdep`, `sim.2pl`

Examples

```
#simulating Rasch homogenous data
#100 persons, 10 items, parameter drawn from N(0,1)
X <- sim.rasch(100, 10)

#person parameters drawn from uniform distribution, fixed cutpoint
ppar <- runif(100,-2,2)
X <- sim.rasch(ppar, 10, cutpoint = 0.5)
```

---

sim.xdim *Simulation of multidimensional binary data*

Description

This utility function simulates a 0-1 matrix violating the unidimensionality assumption in the Rasch model.

Usage

```
sim.xdim(persons, items, Sigma, weightmat, seed = NULL,
   cutpoint = "randomized")
```

Arguments

`persons`	Either a matrix (each column corresponds to a dimension) of person parameters or an integer indicating the number of persons (see details).
`items`	Either a vector of item parameters or an integer indicating the number of items (see details).

Sigma	A positive-definite symmetric matrix specifying the covariance matrix of the variables.
weightmat	Matrix for item-weights for each dimension (columns).
seed	A seed for the random number generated can be set.
cutpoint	Either "randomized" for a randomized tranformation of the model probability matrix into the model 0-1 matrix or an integer value between 0 and 1 (see details).

Details

If persons is specified as matrix, Sigma is ignored. If items is an integer value, the corresponding parameter vector is drawn from N(0,1). The cutpoint argument refers to the transformation of the theoretical probabilities into a 0-1 data matrix. A randomized assingment implies that for each cell an additional random number is drawn. If the model probability is larger than this value, the person gets 1 on this particular item, if smaller, 0 is assigned. Alternatively, a numeric probability cutpoint can be assigned and the 0-1 scoring is carried out according to the same rule.

If weightmat is not specified, a random indicator matrix is generated where each item is a measurement of only one dimension. For instance, the first row for a 3D-model could be (0,1,0) which means that the first item measures the second dimension only. This corresponds to the between-item multidimensional model presented by Adams et al. (1997).

Sigma reflects the VC-structure for the person parameters drawn from a multivariate standard normal distribution. Thus, the diagonal elements are typically 1 and the lower the covariances in the off-diagonal, the stronger the model violation.

References

Adams, R. J., Wilson, M., & Wang, W. C. (1997). The multidimensional random coefficients multinomial logit model. Applied Psychological Measurement, 21, 1-23.

Glas, C. A. W. (1992). A Rasch model with a multivariate distribution of ability. In M. Wilson (Ed.), Objective Measurement: Foundations, Recent Developments, and Applications (pp. 236-258). Norwood, NJ: Ablex.

See Also

sim.rasch, sim.locdep, sim.2pl

Examples

```
# 500 persons, 10 items, 3 dimensions, random weights.
Sigma <- matrix(c(1, 0.01, 0.01, 0.01, 1, 0.01, 0.01, 0.01, 1), 3)
X <- sim.xdim(500, 10, Sigma)

#500 persons, 10 items, 2 dimensions, weights fixed to 0.5
itemvec <- runif(10, -2, 2)
Sigma <- matrix(c(1, 0.05, 0.05, 1), 2)
weights <- matrix(0.5, ncol = 2, nrow = 10)
X <- sim.xdim(500, itemvec, Sigma, weightmat = weights)
```

`stepwiseIt` *Stepwise item elimination*

Description

This function eliminates items stepwise according to one of the following criteria: itemfit, Wald test, Andersen's LR-test

Usage

```
## S3 method for class 'eRm':
stepwiseIt(object, criterion = list("itemfit"), alpha = 0.05, verbose = TRUE, maxst
```

Arguments

`object`	Object of class `eRm`.
`criterion`	List with either `"itemfit"`, `"Waldtest"` or `"LRtest"` as first element. Optionally, for the Waldtest and LRtest a second element containing the split criterion can be specified (see details).
`alpha`	Significance level.
`verbose`	If `TRUE` intermediate results are printed out.
`maxstep`	Maximum number of elimination steps. If `NA` the procedure stops when the itemset is Rasch homogeneous.

Details

If `criterion = list("itemfit")` the elimination stops when none of the p-values in itemfit is significant. Within each step the item with the largest chi-squared itemfit value is excluded.

If `criterion = list("Waldtest")` the elimination stops when none of the p-values resulting from the Wald test is significant. Within each step the item with the largest z-value in Wald test is excluded.

If `criterion = list("LRtest")` the elimination stops when Andersen's LR-test is not significant. Within each step the item with the largest z-value in Wald test is excluded.

Value

The function returns an object of class `step` containing:

`X`	Reduced data matrix (bad items eliminated)
`fit`	Object of class `eRm` with the final item parameter elimination
`it.elim`	Vector contaning the names of the eliminated items
`res.wald`	Elimination results for Wald test criterion
`res.itemfit`	Elimination results for itemfit criterion
`res.LR`	Elimination results for LR-test criterion
`nsteps`	Number of elimination steps

See Also

LRtest.Rm, Waldtest.Rm, itemfit.ppar

Examples

```
## 2pl-data, 100 persons, 10 items
set.seed(123)
X <- sim.2pl(500, 10, 0.4)
res <- RM(X)

## elimination according to itemfit
stepwiseIt(res, criterion = list("itemfit"))

## Wald test based on mean splitting
stepwiseIt(res, criterion = list("Waldtest","mean"))

## Andersen LR-test based on random split
set.seed(123)
groupvec <- sample(1:3, 500, replace = TRUE)
stepwiseIt(res, criterion = list("LRtest",groupvec))
```

---

thresholds *Computation of item-category treshold parameters.*

Description

This function transforms the beta parameters into threshold parameters. These can be interpreted by means of log-odds as visualized in ICC plots.

Usage

```
## S3 method for class 'eRm':
thresholds(object)
## S3 method for class 'threshold':
print(x, ...)
## S3 method for class 'threshold':
summary(object, ...)
## S3 method for class 'threshold':
confint(object, parm, level = 0.95, ...)
```

Arguments

Arguments for thresholds:

object	Object of class eRm.
x	Object of class threshold.

parm	Parameter specification (ignored).
level	Alpha-level.
...	Further arguments to be passed to methods. They are ignored.

Details

For dichotomous models (i.e., RM and LLTM) threshold parameters are not computed. The print method returns a location parameter for each item which is the mean of the corresponding threshold parameters. For LPCM and LRSM the thresholds are computed for each design matrix block (i.e., measurement point/group) separately (PCM and RSM have only 1 block).

Value

The function thresholds returns an object of class threshold containing:

threshpar	Vector with threshold parameters.
se.thresh	Vector with standard errors.
threshtable	Data frame with location and threshold parameters.

References

Andrich, D. (1978). Application of a psychometric rating model to ordered categories which are scored with successive integers. Applied Psychological Measurement, 2, 581-594.

See Also

plotICC.Rm

Examples

```
#Threshold parameterization for a rating scale model
data(rsmdat)
res <- RSM(rsmdat)
th.res <- thresholds(res)
th.res
confint(th.res)
summary(th.res)

#Threshold parameters for a PCM with ICC plot
data(pcmdat)
res <- PCM(pcmdat)
th.res <- thresholds(res)
th.res
plotICC(res)

#Threshold parameters for a LPCM:
#Block 1: t1, g1; Block 2: t1, g2; ...; Block 6: t2,g3

data(lpcmdat)
```

```
G <- c(rep(1,7),rep(2,7),rep(3,6)) # group vector for 3 groups
res <- LPCM(lpcmdat, mpoints = 2, groupvec = G)
th.res <- thresholds(res)
th.res
```

`Waldtest` *Item-Specific Wald Test*

Description

Performs a Wald test on item-level by splitting subjects into subgroups.

Usage

```
## S3 method for class 'Rm':
Waldtest(object, splitcr = "median")
## S3 method for class 'wald':
print(x,...)
```

Arguments

`object`	Object of class `RM`.
`splitcr`	Split criterion for subject raw score splitting. `median` uses the median as split criterion, `mean` performs a mean-split. Optionally `splitcr` can also be a dichotomous vector which assigns each person to a certain subgroup (e.g., following an external criterion). This vector can be numeric, character or a factor.
`x`	Object of class `wald`.
`...`	Further arguments passed to or from other methods. They are ignored in this function.

Details

Items are eliminated if they not have the same number of categories in each subgroup. To avoid this problem, for RSM and PCM it is considered to use a random or another user-defined split. If the data set contains missing values and `mean` or `median` is specified as splitcriterion, means or medians are calculated for each missing value subgroup and consequently used for raw score splitting.

Value

Returns an object of class `wald` containing:

`coef.table`	Data frame with test statistics, z- and p-values.
`betapar1`	Beta parameters for first subgroup
`se.beta1`	Standard errors for first subgroup

betapar2	Beta parameters for second subgroup
se.beta2	Standard errors for second subgroup
se.beta2	Standard errors for second subgroup
spl.gr	Names and levels for splitcr.
call	The matched call.

Author(s)

Patrick Mair, Reinhold Hatzinger

References

Fischer, G. H., and Molenaar, I. (1995). Rasch Models - Foundations, Recent Developements, and Applications. Springer.

Fischer, G. H., and Scheiblechner, H. (1970). Algorithmen und Programme fuer das probabilistische Testmodell von Rasch [Algorithms and programs for Rasch's probabilistic test model]. Psychologische Beitraege, 12, 23-51.

See Also

LRtest

Examples

```
#Wald test for Rasch model with user-defined subject split
data(raschdat2)
res <- RM(raschdat2)
splitvec <- sample(1:2,25,replace=TRUE)
Waldtest(res, splitcr = splitvec)

#Wald test for RSM eliminates 4 items (with median split)
data(rsmdat)
res <- RSM(rsmdat)
Waldtest(res)
```

Index