# Package 'equateIRT'

January 27, 2015

**Type** Package

**Title** Direct, Chain and Average Equating Coefficients with Standard Errors Using IRT Methods.

**Imports** statmod

**Version** 1.2

**Date** 2014-11-14

**Author** Michela Battauz

**Maintainer** Michela Battauz <michela.battauz@uniud.it>

**Description** This package computes direct, chain and average (bisector) equating coefficients with standard errors using Item Response Theory (IRT) methods for dichotomous items.

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-11-14 18:12:47

## R topics documented:

---

`equateIRT-package` *Direct, Chain and Average Equating Coefficients with Standard Errors Using IRT Methods.*

---

## Description

This package computes direct, chain and average (bisector) equating coefficients with standard errors using IRT methods for dichotomous items. The IRT models included are the three-parameter logistic model, the two-parameter logistic model, the one-parameter logistic model and the Rasch model.

## Details

| | |
|---|---|
| Package: | equateIRT |
| Type: | Package |
| Version: | 1.2 |
| Date: | 2014-11-14 |
| License: | GPL (>= 2) |

Direct equating coefficients and their standard errors between forms presenting common items can be computed using function `direc`. The equating methods implemented are "mean-mean", "mean-geometric mean", "mean-sigma", "Haebara" and "Stocking-Lord". Estimates of item parameters and their covariance matrix can be imported from the R packages `ltm` and `mirt` or from the IRT programs IRTPRO and flexMIRT using functions `import.ltm`, `import.mirt`, `import.irtpro` and `import.flexmirt`. Item parameter estimates from other software can be imported as well by the user. Data should be previously organized using function `modIRT`. Function `alldirec` computes all direct equating coefficients (with standard errors) between all pairs of a list of forms. Function `chainec` computes chain equating coefficients (and standard errors) given direct equating coefficients between forms directly linked. Average equating coefficients with standard errors can be calculated using function `bisectorec`, that implements the (weighted) bisector method. Three simulated datasets are available for illustrative purposes. These datasets contain item parameter coefficients and their covariance matrix. In particular, `est3pl` concerns a three-parameter logistic model, `est2pl` regards a two-parameter logistic model, and `estrasch` refers to a Rasch model. The estimates included in `est2pl` are obtained from the dataset `data2pl`, also contained in the package.

## Author(s)

Michela Battauz

Maintainer: Michela Battauz <michela.battauz@uniud.it>

## References

Battauz, M. (2013). IRT Test Equating in Complex Linkage Plans. *Psychometrika*, **78**, 464–480.

Cai L. (2013). *FlexMIRT version 2: Flexible Multilevel Multidimensional Item Analysis and Test Scoring [Computer Software]*. Chapel Hill, NC: Vector Psychometric Group.

Cai, L., du Toit, S. H. C., Thissen, D. (2011). *IRTPRO: Flexible, multidimensional, multiple categorical IRT modeling [Computer software]*. Chicago: Scientific Software International.

Chalmers, R. P. (2012). mirt: A Multidimensional Item Response Theory Package for the R Environment. *Journal of Statistical Software*, **48**, 1–29.

Holland, P.W. and Strawderman, W.E. (2011). How to average equating functions if you must. In A.A. von Davier (Ed.), *Statistical models for test equating, scaling, and linking* (pp. 89–107). New York: Springer.

Kolen, M.J. and Brennan, R.L. (2004). *Test equating, scaling, and linking: methods and practices*, 2nd ed., New York: Springer.

Ogasawara, H. (2000). Asymptotic standard errors of IRT equating coefficients using moments. *Economic Review (Otaru University of Commerce)*, **51**, 1–23.

Ogasawara, H. (2001). Standard Errors of Item Response Theory Equating/Linking by Response Function Methods. *Applied Psychological Measurement*, **25**, 53–67.

Rizopoulos, D. (2006). ltm: an R package for latent variable modelling and item response theory analyses. *Journal of Statistical Software*, **17**, 1–25.

---

`alldirec` *Direct Equating Coefficients Between All Pairs of a List of Forms*

---

## Description

Calculates direct equating coefficients and standard errors using IRT methods between all pairs of a list of forms.

## Usage

```
alldirec(mods, method = "mean-mean", all = FALSE, quadrature = TRUE, nq = 30)
```

## Arguments

| | |
|---|---|
| `mods` | an object of the class `modIRT` containing item parameter coefficients and their covariance matrix of the forms to be equated. |
| `method` | the equating method to be used. This should be one of "`mean-mean`", "`mean-gmean`", "`mean-sigma`", "`Haebara`" or "`Stocking-Lord`". |
| `all` | logical; if FALSE forms that do not have common items will not appear in the output. |

| | |
|---|---|
| quadrature | logical; if TRUE the Gauss-Hermite quadrature is used to approximate the integral in the function that is minimized in the Haebara and Stocking-Lord methods. If FALSE the integral is replaced with a sum over 40 equally spaced values ranging from -4 to 4 with an increment of 0.05 and weights equal to one for all values. |
| nq | number of quadrature points used for the Gauss-Hermite quadrature if quadrature is TRUE |

## Value

An object of class eqclist consisting in a list with length equal to the number of pairs of forms equated. Each component of the list is an object of class eqc returned by function direc.

## Author(s)

Michela Battauz

## References

Kolen, M.J. and Brennan, R.L. (2004). *Test equating, scaling, and linking: methods and practices*, 2nd ed., New York: Springer

Ogasawara, H. (2000). Asymptotic standard errors of IRT equating coefficients using moments. *Economic Review (Otaru University of Commerce)*, **51**, 1–23.

Ogasawara, H. (2001). Standard Errors of Item Response Theory Equating/Linking by Response Function Methods. *Applied Psychological Measurement*, **25**, 53–67.

## See Also

direc, eqc, itm, modIRT, summary.eqclist

## Examples

```
# three-parameter logistic model
# direct equating coefficients using the "Stocking-Lord" method
data(est3pl)
test <- paste("test", 1:5, sep = "")
mod3pl <- modIRT(coef = est3pl$coef, var = est3pl$var, names = test, display = FALSE)
direclist3pl <- alldirec(mods = mod3pl, method = "Stocking-Lord")
summary(direclist3pl)
summary(direclist3pl$test1.test2)

# two-parameter logistic model
# direct equating coefficients using the "Haebara" method
data(est2pl)
test <- paste("test", 1:5, sep = "")
mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test, display = FALSE)
direclist2pl <- alldirec(mods = mod2pl, method = "Haebara")
summary(direclist2pl)
summary(direclist2pl$test1.test5)
```

```
# Rasch model
# direct equating coefficients using the "mean-mean" method
data(estrasch)
test <- paste("test", 1:5, sep = "")
modrasch <- modIRT(coef = estrasch$coef, var = estrasch$var, names = test,
display = FALSE)
direclistrasch <- alldirec(mods = modrasch, method = "mean-mean", all = TRUE)
summary(direclistrasch)
summary(direclistrasch$test5.test4)
```

---

`bisectorec` *Bisector Equating Coefficients*

---

## Description

Calculates average equating coefficients using the bisector method and standard errors given a set of direct and chain equating coefficients.

## Usage

```
bisectorec(ecall, mods, weighted = TRUE, unweighted = TRUE)
```

## Arguments

| | |
|---|---|
| `ecall` | list of objects of class `eqc` or `ceqc` returned by functions `direc` and `chainec`. |
| `mods` | an object of class `modIRT` containing item parameter coefficients and their covariance matrix of the forms to be equated. |
| `weighted` | logical; if TRUE weighted bisector coefficients are computed. |
| `unweighted` | logical; if TRUE unweighted bisector coefficients are computed. |

## Value

An object of class `meqc` with components

| | |
|---|---|
| `coef` | data frame containing `link`, `path`, coefficient `A`, coefficient `B`, standard error of coefficient A (`seA`), standard error of coefficient B (`seB`) and `weights` of direct, chain and bisector equating coefficients. |
| `method` | the equating method used. |

## Author(s)

Michela Battauz

## References

Battauz, M. (2013). IRT Test Equating in Complex Linkage Plans. *Psychometrika*, **78**, 464–480.

Holland, P.W. and Strawderman, W.E. (2011). How to average equating functions if you must. In A.A. von Davier (Ed.), *Statistical models for test equating, scaling, and linking* (pp. 89–107). New York: Springer.

## See Also

chainec, convert, direc, eqc, summary.meqc

## Examples

```
# three-parameter logistic model
# direct equating coefficients using the "Stocking-Lord" method
data(est3pl)
test <- paste("test", 1:5, sep = "")
mod3pl <- modIRT(coef = est3pl$coef, var = est3pl$var, names = test, display = FALSE)
direclist3pl <- alldirec(mods = mod3pl, method = "Stocking-Lord")
# compute chain equating coefficients for path 1,2,3,4
pth1 <- paste("test", 1:4, sep = "")
pth1 <- data.frame(t(pth1), stringsAsFactors = FALSE)
chainec1 <- chainec(direclist = direclist3pl, pths = pth1)
# compute chain equating coefficients for path 1,5,4
pth2 <- c(paste("test", c(1,5,4), sep = ""))
pth2 <- data.frame(t(pth2), stringsAsFactors = FALSE)
chainec2 <- chainec(direclist = direclist3pl, pths = pth2)
# compute chain equating coefficients for path 1,2,3,4,5
pth3 <- paste("test", 1:5, sep = "")
pth3 <- data.frame(t(pth3), stringsAsFactors = FALSE)
chainec3 <- chainec(direclist = direclist3pl, pths = pth3)
# create a list of objects of class eqc or ceqc
ecall <- c(chainec1, chainec2, chainec3, direclist3pl["test1.test5"])
# compute bisector and weighted bisector coefficients
allec <- bisectorec(ecall = ecall, mods = mod3pl, weighted = TRUE, unweighted = TRUE)
summary(allec)

# two-parameter logistic model
# direct equating coefficients using the "Haebara" method
data(est2pl)
test <- paste("test", 1:5, sep = "")
mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test, display = FALSE)
direclist2pl <- alldirec(mods = mod2pl, method = "Haebara")
# compute chain equating coefficients for path 1,2,3,4
pth1 <- paste("test", 1:4, sep = "")
pth1 <- data.frame(t(pth1), stringsAsFactors = FALSE)
chainec1 <- chainec(direclist = direclist2pl, pths = pth1)
# compute chain equating coefficients for path 1,5,4
pth2 <- c(paste("test", c(1,5,4), sep = ""))
pth2 <- data.frame(t(pth2), stringsAsFactors = FALSE)
chainec2 <- chainec(direclist = direclist2pl, pths = pth2)
# compute chain equating coefficients for path 1,2,3,4,5
pth3 <- paste("test", 1:5, sep = "")
pth3 <- data.frame(t(pth3), stringsAsFactors = FALSE)
chainec3 <- chainec(direclist = direclist2pl, pths = pth3)
# create a list of objects of class eqc or ceqc
ecall <- c(chainec1, chainec2, chainec3, direclist2pl["test1.test5"])
# compute bisector and weighted bisector coefficients
allec <- bisectorec(ecall = ecall, mods = mod2pl, weighted = TRUE, unweighted = TRUE)
summary(allec)
```

```
# Rasch model
# direct equating coefficients using the "mean-mean" method
data(estrasch)
test <- paste("test", 1:5, sep = "")
modrasch <- modIRT(coef = estrasch$coef, var = estrasch$var, names = test,
display = FALSE)
direclistrasch <- alldirec(mods = modrasch, method = "mean-mean", all = TRUE)
# compute chain equating coefficients for path 1,2,3,4
pth1 <- paste("test", 1:4, sep = "")
pth1 <- data.frame(t(pth1), stringsAsFactors = FALSE)
chainec1 <- chainec(direclist = direclistrasch, pths = pth1)
# compute chain equating coefficients for path 1,5,4
pth2 <- c(paste("test", c(1,5,4), sep = ""))
pth2 <- data.frame(t(pth2), stringsAsFactors = FALSE)
chainec2 <- chainec(direclist = direclistrasch, pths = pth2)
# compute chain equating coefficients for path 1,2,3,4,5
pth3 <- paste("test", 1:5, sep = "")
pth3 <- data.frame(t(pth3), stringsAsFactors = FALSE)
chainec3 <- chainec(direclist = direclistrasch, pths = pth3)
# create a list of objects of class eqc or ceqc
ecall <- c(chainec1, chainec2, chainec3, direclistrasch["test1.test5"])
# compute bisector and weighted bisector coefficients
allec <- bisectorec(ecall = ecall, mods = mod3pl, weighted = TRUE, unweighted = TRUE)
summary(allec)
```

---

`chainec` *Chain Equating Coefficients*

---

## Description

Calculates chain (indirect) equating coefficients and standard errors using IRT methods.

## Usage

```
chainec(r = NULL, direclist, f1 = NULL, f2 = NULL, pths = NULL)
```

## Arguments

| | |
|---|---|
| `r` | length of the chain, that is the number of forms used for equating including extremes. It should be at least 3. It does not need to be specified if argument `pths` is not NULL. |
| `direclist` | an object of the class `eqclist` return by function `alldirec` containing direct equating coefficients between pairs of forms. |
| `f1` | the name of the first form of the chain. |
| `f2` | the name of the last form of the chain. |
| `pths` | data frame containing the path used for equating. The number of columns is equal to `r` and the number of rows is equal to the equatings that have to be performed. If `NULL` all the chain equating coefficients of length `r` will be computed. |

## Details

Equating coefficients perform the conversion from the scale of the first form to the scale of the last form of the path.

## Value

An object of class `ceqclist` consisting in a list with length equal to the number of chain equating coefficients computed. Each component of the list is an object of class `ceqc` with components

| | |
|---|---|
| `tab1` | item parameters of the first form. |
| `tab2` | item parameters of the last form. |
| `tab` | Data frame containing item names (`Item`), item parameters of the first form (e.g. `test1`), item parameters of the last form (e.g. `test3`), and item parameters of the first form converted in the scale of the last form (e.g. `test1.as.test3`). |
| `varAll` | covariance matrix of item parameters of all forms used in the chain. |
| `partial` | partial derivatives of equating coefficients A and B with respect to item parameters. |
| `A` | equating coefficient A. |
| `B` | equating coefficient B. |
| `varAB` | covariance matrix of the equating coefficients. |
| `commonitem` | list of length r-1 containing the names of common item parameters between adjacent forms. |
| `ni` | vector containing number of common items between pairs of adjacent forms. |
| `forms` | names of equated forms. |
| `method` | the equating method used. |
| `itmp` | number of item parameters of the IRT model. |

## Author(s)

Michela Battauz

## References

Battauz, M. (2013). IRT Test Equating in Complex Linkage Plans. *Psychometrika*, **78**, 464–480.

Kolen, M.J. and Brennan, R.L. (2004). *Test equating, scaling, and linking: methods and practices*, 2nd ed., New York: Springer

## See Also

`alldirec`, `eqc`, `itm`, `summary.ceqc`, `summary.ceqclist`

## Examples

```
# three-parameter logistic model
# direct equating coefficients using the "Stocking-Lord" method
data(est3pl)
test<-paste("test", 1:5, sep = "")
mod3pl <- modIRT(coef = est3pl$coef, var = est3pl$var, names = test, display = FALSE)
direclist3pl <- alldirec(mods = mod3pl, method = "Stocking-Lord")
# compute all chain equating coefficients of length 4
chainec4 <- chainec(r = 4, direclist = direclist3pl)
summary(chainec4)
summary(chainec4$test1.test2.test3.test4)
# compute all chain equating coefficients of length 4
# where the first form is test1
chainec4.1 <- chainec(r = 4, direclist = direclist3pl, f1 = "test1")
summary(chainec4.1)
# compute all chain equating coefficients of length 4
# where the first form is test1 and the last form is test4
chainec4.14 <- chainec(r = 4, direclist = direclist3pl, f1 = "test1", f2 = "test4")
summary(chainec4.14)


# two-parameter logistic model
# direct equating coefficients using the "Haebara" method
data(est2pl)
test<-paste("test", 1:5, sep = "")
mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test, display = FALSE)
direclist2pl <- alldirec(mods = mod2pl, method = "Haebara")
# compute chain equating coefficients of a given path
pth <- c(paste("test", c(1,5,4), sep = ""))
pth <- data.frame(t(pth), stringsAsFactors = FALSE)
chainec154 <- chainec(direclist = direclist2pl, pths = pth)
summary(chainec154)


# Rasch model
# direct equating coefficients using the "mean-mean" method
data(estrasch)
test<-paste("test", 1:5, sep = "")
modrasch <- modIRT(coef = estrasch$coef, var = estrasch$var, names = test,
display = FALSE)
direclistrasch <- alldirec(mods = modrasch, method = "mean-mean", all = TRUE)
# compute chain equating coefficients of two given paths
pth1 <- paste("test", 1:3, sep = "")
pth1 <- data.frame(t(pth1), stringsAsFactors = FALSE)
pth2 <- c(paste("test", c(1,5,4), sep = ""))
pth2 <- data.frame(t(pth2), stringsAsFactors = FALSE)
pths <- rbind(pth1, pth2)
chainec1 <- chainec(direclist = direclistrasch, pths = pths)
summary(chainec1)
```

convert *Item Parameters Conversion*

## Description

Converts item and person parameters using equating coefficients.

## Usage

```
convert(A, B, coef = NULL, person.par = NULL)
```

## Arguments

| | |
|---|---|
| A | equating coefficient A. |
| B | equating coefficient B. |
| coef | vector of item parameters return by function modIRT. |
| person.par | vector of person parameters estimates. |

## Details

Difficulty parameters $b$ are converted using transformation $b \cdot A + B$; discrimination parameters $a$ are converted using transformation $a/A$; guessing parameters c are not transformed.

Person parameters $\theta$ are converted using transformation $\theta \cdot A + B$.

## Value

A list with components

| | |
|---|---|
| coef | vector of item parameters transformed. |
| person.par | vector of person parameters transformed |

## Author(s)

Michela Battauz

## References

Kolen, M.J. and Brennan, R.L. (2004). *Test equating, scaling, and linking: methods and practices*, 2nd ed., New York: Springer

## See Also

itm, modIRT

## Examples

```
# conversion using direct coefficients
# three-parameter logistic model
# direct equating coefficients between forms 1 and 2 using the Stocking-Lord method
data(est3pl)
test <- paste("test", 1:5, sep = "")
mod3pl <- modIRT(coef = est3pl$coef, var = est3pl$var, names = test, display = FALSE)
l12 <- direc(mod1 = mod3pl[1], mod2 = mod3pl[2], method = "Stocking-Lord")
convert(A = l12$A, B = l12$B, coef = coef(mod3pl$test1))
# the conversion of item parameters is obtained also using
itm(l12)

# conversion using bisector coefficients
# two-parameter logistic model
# direct equating coefficients using the "Haebara" method
data(est2pl)
test <- paste("test", 1:5, sep = "")
mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test, display = FALSE)
direclist2pl <- alldirec(mods = mod2pl, method = "Haebara")
# compute chain equating coefficients for path 1,2,3,4
pth1 <- paste("test", 1:4, sep = "")
pth1 <- data.frame(t(pth1), stringsAsFactors = FALSE)
chainec1 <- chainec(direclist = direclist2pl, pths = pth1)
# compute chain equating coefficients for path 1,5,4
pth2 <- c(paste("test", c(1,5,4), sep = ""))
pth2 <- data.frame(t(pth2), stringsAsFactors = FALSE)
chainec2 <- chainec(direclist = direclist2pl, pths = pth2)
# create a list of objects of class ceqc
ecall <- c(chainec1, chainec2)
# compute bisector and weighted bisector coefficients
allec <- bisectorec(ecall = ecall, mods = mod2pl, weighted = TRUE, unweighted = TRUE)
summary(allec)
eqc14<-eqc(allec, link = "test1.test4", path = "bisector")
convert(A = eqc14$A, B = eqc14$B, coef = coef(mod2pl$test1), person.par = seq(-3, 3, 0.5))
```

---

# data2pl *Simulated Data Sets*

---

## Description

Five simulated data sets from a two-parameter logistic model.

## Usage

```
data(data2pl)
```

## Format

A list of length 5, containing 5 data frames with 5000 dichotomous responses to 20 items.

## Author(s)

Michela Battauz

## See Also

est2pl, import.ltm

## Examples

```
data(data2pl)
data2pl[[1]][1:3,]
```

---

direc *Direct Equating Coefficients*

---

## Description

Calculates direct equating coefficients and standard errors using IRT methods.

## Usage

```
direc(mod1, mod2, method = "mean-mean", suff1 = ".1", suff2 = ".2",
   D = 1, quadrature = TRUE, nq = 30)
```

## Arguments

| | |
|---|---|
| mod1 | an object of the class modIRT containing item parameter coefficients and their covariance matrix of the first form. |
| mod2 | an object of the class modIRT containing item parameter coefficients and their covariance matrix of the second form. |
| method | the equating method to be used. This should be one of "mean-mean", "mean-gmean", "mean-sigma", "Haebara" or "Stocking-Lord". |
| suff1 | suffix to identify the first form to be equated. |
| suff2 | suffix to identify the second form to be equated. |
| D | constant D of the IRT model used to estimate item parameters. See below for more details. |
| quadrature | logical; if TRUE the Gauss-Hermite quadrature is used to approximate the integral in the function that is minimized in the Haebara and Stocking-Lord methods. If FALSE the integral is replaced with a sum over 40 equally spaced values ranging from -4 to 4 with an increment of 0.05 and weights equal to one for all values. |
| nq | number of quadrature points used for the Gauss-Hermite quadrature if quadrature is TRUE. |

## Details

Equating coefficients perform the conversion from the scale of the first form to the scale of the second form.

In the three-parameter logistic model the probability of a positive response on item $i$ is

$$\pi_i = c_i + (1 - c_i)\frac{\exp[Da_i(\theta - b_i)]}{1 + \exp[Da_i(\theta - b_i)]}.$$

where $a_i$ is the item discrimination parameter, $b_i$ is the item difficulty parameter, $c_i$ is the item guessing parameter and $\theta$ is the latent ability. The constant $D$ can be specified using argument `D` of the `direc` function. The two-parameter logistic model can be obtained by setting $c_i$ equal to zero, the one-parameter logistic model can be obtained by setting $c_i$ equal to zero and $a_i$ costant across items, while the Rasch model can be obtained by setting $c_i$ equal to zero and $a_i$ equal to 1.

The type of IRT model does not need to be specified as it is obtained from arguments `mod1` and `mod2`.

## Value

An object of class `eqc` with components

| | |
|---|---|
| `tab1` | item parameters of the first form. |
| `tab2` | item parameters of the second form. |
| `tab` | Data frame containing item names (`Item`), item parameters of the first form (e.g. `test1`), item parameters of the second form (e.g. `test2`), and item parameters of the first form converted in the scale of the second form (e.g. `test1.as.test2`). |
| `var12` | covariance matrix of item parameters of the first and the second form. |
| `partial` | partial derivatives of equating coefficients A and B with respect to item parameters. |
| `A` | equating coefficient A. |
| `B` | equating coefficient B. |
| `varAB` | covariance matrix of the equating coefficients. |
| `commonitem` | list of length 1 containing the names of common item parameters. |
| `ni` | number of common items. |
| `forms` | names of equated forms. |
| `method` | the equating method used. |
| `itmp` | number of item parameters of the IRT model. |

## Author(s)

Michela Battauz

## References

Kolen, M.J. and Brennan, R.L. (2004). *Test equating, scaling, and linking: methods and practices*, 2nd ed., New York: Springer

Ogasawara, H. (2000). Asymptotic standard errors of IRT equating coefficients using moments. *Economic Review (Otaru University of Commerce)*, **51**, 1–23.

Ogasawara, H. (2001). Standard Errors of Item Response Theory Equating/Linking by Response Function Methods. *Applied Psychological Measurement*, **25**, 53–67.

## See Also

eqc, itm, modIRT, summary.eqc

## Examples

```
# three-parameter logistic model
# direct equating coefficients between forms 1 and 2 using the Stocking-Lord method
data(est3pl)
test <- paste("test", 1:5, sep = "")
mod3pl <- modIRT(coef = est3pl$coef, var = est3pl$var, names = test, display = FALSE)
l12 <- direc(mod1 = mod3pl[1], mod2 = mod3pl[2], method = "Stocking-Lord")
summary(l12)

# two-parameter logistic model
# direct equating coefficients between forms 1 and 5 using the Haebara method
data(est2pl)
test <- paste("test", 1:5, sep = "")
mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test, display = FALSE)
l15 <- direc(mod1 = mod2pl[1], mod2 = mod2pl[5], method = "Haebara")
summary(l15)

# Rasch model
# direct equating coefficients between forms 5 and 4 using the mean-mean method
data(estrasch)
test <- paste("test", 1:5, sep = "")
modrasch <- modIRT(coef = estrasch$coef, var = estrasch$var, names = test,
display = FALSE)
l54 <- direc(mod1 = modrasch[5], mod2 = modrasch[4], method = "mean-mean")
summary(l54)
```

---

eqc *Extract Equating Coefficients*

---

## Description

eqc is a generic function which extracts the equating coefficients.

## Usage

```
eqc(x, ...)

## S3 method for class 'eqc'
eqc(x, ...)

## S3 method for class 'eqclist'
eqc(x, link = NULL, ...)

## S3 method for class 'ceqc'
eqc(x, ...)

## S3 method for class 'ceqclist'
eqc(x, link = NULL, path = NULL, ...)

## S3 method for class 'meqc'
eqc(x, link = NULL, path = NULL, ...)
```

## Arguments

| | |
|---|---|
| x | object of the class eqc returned by function direc or of the class eqclist returned by function alldirec or of the class ceqc and ceqclist returned by function chainec or of the class meqc returned by function bisectorec. |
| link | a character string with the names of the two forms being linked separated by a dot (e.g. "test1.test3"). |
| path | a character string with the names of the forms that constitute the path separated by a dot (e.g. "test1.test2.test3"). |
| ... | further arguments passed to or from other methods. |

## Value

A data frame containing the equating coefficients for every link and path.

## Author(s)

Michela Battauz

## See Also

alldirec, bisectorec, chainec, direc

## Examples

```
# two-parameter logistic model
data(est2pl)
test<-paste("test", 1:5, sep = "")
mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test, display = FALSE)
# direct equating coefficients between forms 1 and 2 using the Haebara method
l12 <- direc(mod1 = mod2pl[1], mod2 = mod2pl[2], method = "Haebara")
```

```
# all direct equating coefficients using the Haebara method
direclist2pl <- alldirec(mods = mod2pl, method = "Haebara")
# compute all chain equating coefficients of length 3
chainec3 <- chainec(r = 3, direclist = direclist2pl)
# compute chain equating coefficients for path 1,2,3,4
pth1 <- paste("test", 1:4, sep = "")
pth1 <- data.frame(t(pth1), stringsAsFactors = FALSE)
chainec1 <- chainec(direclist = direclist2pl, pths = pth1)
# compute chain equating coefficients for path 1,5,4
pth2 <- c(paste("test", c(1,5,4), sep = ""))
pth2 <- data.frame(t(pth2), stringsAsFactors = FALSE)
chainec2 <- chainec(direclist = direclist2pl, pths = pth2)
# create a list of objects of class ceqc
ecall <- c(chainec1, chainec2)
# compute bisector and weighted bisector coefficients
allec <- bisectorec(ecall = ecall, mods = mod2pl, weighted = TRUE, unweighted = TRUE)

# extract equating coefficients
eqc(l12)
eqc(direclist2pl)
eqc(direclist2pl,link="test1.test2")
eqc(chainec3)
eqc(chainec3,link="test1.test3")
eqc(allec)
eqc(allec,path="bisector")
```

---

`est2pl` *Item Parameter Estimates and Covariance Matrices of a Two-Parameter Logistic Model*

---

## Description

This dataset includes item parameter estimates and covariance matrices of a two-parameter logistic model applied to 5 simulated datasets with common items. The dichotomous item responses can be found in the dataset `data2pl`. See details for more information on the linkage plan.

## Usage

```
data(est2pl)
```

## Format

A list of length 2 with components:

`coef` a list of length 5 containing the matrices of item parameter estimates. Each matrix presents 2 columns; the first column contains difficulty parameters and the second column contains discrimination parameters. See details for information on the parameterization used. Names of rows correspond to the names of the items.

`var` a list of length 5 containing the covariance matrices of item parameter estimates.

## Details

Every form is composed by 20 items and presents 10 items in common with adjacent forms. Furthermore, forms 1 and 5 present 10 common items. Use `linkp` to obtain a matrix with elements equal to the number of common items between different forms.

Item parameters are given under the parameterization used in the `ltm` package. Under this parameterization, the two-parameter logistic model is as follows

$$\pi_i = \frac{\exp(\beta_{1i} + \beta_{2i} z)}{1 + \exp(\beta_{1i} + \beta_{2i} z)},$$

where $\pi_i$ denotes the conditional probability of responding correctly to the $i$th item given $z$, $\beta_{1i}$ is the easiness parameter, $\beta_{2i}$ is the discrimination parameter, and $z$ denotes the latent ability.

## Author(s)

Michela Battauz

## See Also

`data2pl`, `linkp`, `modIRT`

## Examples

```
data(est2pl)
est2pl$coef
est2pl$var
linkp(coef = est2pl$coef)
```

---

`est3pl` *Item Parameter Estimates and Covariance Matrices of a Three-Parameter Logistic Model*

---

## Description

This dataset includes item parameter estimates and covariance matrices of a three-parameter logistic model applied to 5 simulated datasets with common items. See details for more information on the linkage plan.

## Usage

```
data(est3pl)
```

## Format

A list of length 2 with components:

`coef` a list of length 5 containing the matrices of item parameter estimates. Each matrix presents 3 columns; the first column contains guessing parameters, the second column contains difficulty parameters and the third column contains discrimination parameters. See details for information on the parameterization used. Names of rows correspond to the names of the items.

`var` a list of length 5 containing the covariance matrices of item parameter estimates.

## Details

Every form is composed by 20 items and presents 10 items in common with adjacent forms. Furthermore, forms 1 and 5 present 10 common items. Use `linkp` to obtain a matrix with elements equal to the number of common items between different forms.

Item parameters are given under the parameterization used in the `ltm` package. Under this parameterization, the three-parameter logistic model is as follows

$$\pi_i = c_i + (1 - c_i)\frac{\exp(\beta_{1i} + \beta_{2i}z)}{1 + \exp(\beta_{1i} + \beta_{2i}z)},$$

where $\pi_i$ denotes the conditional probability of responding correctly to the $i$th item given $z$, $c_i$ denotes the guessing parameter, $\beta_{1i}$ is the easiness parameter, $\beta_{2i}$ is the discrimination parameter, and $z$ denotes the latent ability. Furthermore, the guessing parameters are given under this parameterization

$$c_i = \frac{\exp(c_i^*)}{1 + \exp(c_i^*)}.$$

## Author(s)

Michela Battauz

## See Also

`linkp`, `modIRT`

## Examples

```
data(est3pl)
est3pl$coef
est3pl$var
linkp(coef = est3pl$coef)
```

estrasch *Item Parameter Estimates and Covariance Matrices of a Rasch Model*

## Description

This dataset includes item parameter estimates and covariance matrices of a Rasch model applied to 5 simulated datasets with common items. See details for more information on the linkage plan.

## Usage

```
data(estrasch)
```

## Format

A list of length 2 with components:

`coef` a list of length 5 containing the matrices of item parameter estimates. Each matrix presents 2 columns; the first column contains difficulty parameters and the second column is equal to 1. See details for information on the parameterization used. Names of rows correspond to the names of the items.

`var` a list of length 5 containing the covariance matrices of item parameter estimates.

## Details

Every form is composed by 20 items and presents 10 items in common with adjacent forms. Furthermore, forms 1 and 5 present 10 common items. Use `linkp` to obtain a matrix with elements equal to the number of common items between different forms.

Item parameters are given under the parameterization used in the `ltm` package. Under this parameterization, the Rasch model is as follows

$$\pi_i = \frac{\exp(\beta_{1i} + z)}{1 + \exp(\beta_{1i} + z)},$$

where $\pi_i$ denotes the conditional probability of responding correctly to the $i$th item given $z$, $\beta_{1i}$ is the easiness parameter, and $z$ denotes the latent ability.

## Author(s)

Michela Battauz

## See Also

`linkp`, `modIRT`

## Examples

```
data(estrasch)
estrasch$coef
estrasch$var
linkp(coef = estrasch$coef)
```

`import.ltm` *Import Item Parameters Estimates and Covariance Matrices from IRT Software*

## Description

Import estimated item parameters and covariance matrix from the R packages ltm and mirt, and from external software IRTPRO and flexMIRT.

## Usage

```
import.ltm(mod, display = TRUE, digits = 4)

import.mirt(mod, display = TRUE, digits = 3)

import.irtpro(fnamep, fnamev = NULL, fnameirt = NULL, display = TRUE, digits = 2)

import.flexmirt(fnamep, fnamev = NULL, fnameirt = NULL, display = TRUE, digits = 2)
```

## Arguments

| | |
|---|---|
| `mod` | output object from functions `rasch`, `ltm`, or `tpm` of the `ltm` package or from function `mirt` of the `mirt` package. |
| `display` | logical; if TRUE coefficients and standard errors are printed. |
| `digits` | integer indicating the number of decimal places to be used if `display` is `TRUE`. |
| `fnamep` | name of the file containing the estimated item parameters. Typically, -prm.txt. |
| `fnamev` | name of the file containing the covariance matrix of the estimated item parameters. Typically, -cov.txt. |
| `fnameirt` | name of the file containing additional information to link item parameters with the covariance matrix. Typically, -irt.txt. |

## Details

Item parameters are imported with the parameterization used by the software to estimate the IRT model. The usual IRT parameterization can be obtained later by using function `modIRT`.

## Value

A list with components

| | |
|---|---|
| `coef` | item parameter estimates. |
| `var` | covariance matrix of item parameter estimates. |

## Author(s)

Michela Battauz

## References

Cai L. (2013). *FlexMIRT version 2: Flexible Multilevel Multidimensional Item Analysis and Test Scoring [Computer Software]*. Chapel Hill, NC: Vector Psychometric Group.

Cai, L., du Toit, S. H. C., Thissen, D. (2011). *IRTPRO: Flexible, multidimensional, multiple categorical IRT modeling [Computer software]*. Chicago: Scientific Software International.

Chalmers, R. P. (2012). mirt: A Multidimensional Item Response Theory Package for the R Environment. *Journal of Statistical Software*, **48**, 1–29.

Rizopoulos, D. (2006). ltm: an R package for latent variable modelling and item response theory analyses. *Journal of Statistical Software*, **17**, 1–25.

## See Also

modIRT

## Examples

```
## Not run:

#====================
# from package ltm
library(ltm)

# one-parameter logistic model
mod1pl <- rasch(LSAT)
est.mod1pl <- import.ltm(mod1pl)
est.mod1pl

# two-parameter logistic model
data(data2pl)
m1 <- ltm(data2pl[[1]] ~ z1)
estm1 <- import.ltm(m1, display = FALSE)
estm1

#====================
# from package mirt
library(mirt)

# one-parameter logistic model
data(LSAT, package = "ltm")
val <- mirt(LSAT, 1, SE = TRUE, pars = "values")
cnstr <- val[val$name == "a1",]$parnum
mod1pl.m <- mirt(LSAT, 1, SE = TRUE, constrain = list(cnstr))
est.mod1pl.m <- import.mirt(mod1pl.m, digits = 4)
est.mod1pl.m

# two-parameter logistic model
data(data2pl)
m1.m <- mirt(data2pl[[1]], 1, SE = TRUE)
estm1.m <- import.mirt(m1.m, display = FALSE)
estm1.m
```

```
## End(Not run)
```

---

`itm` *Extract Item Parameters*

---

## Description

`itm` is a generic function which extracts a data frame containing the item parameters of two forms being equated in the original scale and item parameters of the first form converted in the scale of the second form.

## Usage

```
itm(x, ...)

## S3 method for class 'eqc'
itm(x, ...)

## S3 method for class 'eqclist'
itm(x, link, ...)

## S3 method for class 'ceqc'
itm(x, ...)

## S3 method for class 'ceqclist'
itm(x, path, ...)
```

## Arguments

| | |
|---|---|
| `x` | object of the class `eqc` returned by function `direc` or of the class `eqclist` returned by function `alldirec` or of the class `ceqc` and `ceqclist` returned by function `chainec`. |
| `link` | a character string with the names of the two forms being linked separated by a dot (e.g. "`test1.test3`"). |
| `path` | a character string with the names of the forms that constitute the path separated by a dot (e.g. "`test1.test2.test3`"). |
| `...` | further arguments passed to or from other methods. |

## Value

A data frame containing item names (`Item`), item parameters of the first form (e.g. `test1`), item parameters of the last form (e.g. `test3`), and item parameters of the first form converted in the scale of the last form (e.g. `test1.as.test3`).

## Author(s)

Michela Battauz

## See Also

convert, alldirec, chainec, direc

## Examples

```
# two-parameter logistic model
data(est2pl)
test<-paste("test", 1:5, sep = "")
mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test, display = FALSE)
# direct equating coefficients between forms 1 and 2 using the Haebara method
l12 <- direc(mod1 = mod2pl[1], mod2 = mod2pl[2], method = "Haebara")
# all direct equating coefficients using the Haebara method
direclist2pl <- alldirec(mods = mod2pl, method = "Haebara")
# compute all chain equating coefficients of length 3
chainec3 <- chainec(r = 3, direclist = direclist2pl)

# extract item parameters
itm(l12)
itm(direclist2pl,link="test1.test2")
itm(chainec3,path="test1.test2.test3")
```

---

# linkp *Linkage Plan*

---

## Description

Calculates the number of common items between a list of forms.

## Usage

```
linkp(coef)
```

## Arguments

| | |
|---|---|
| coef | list of matrices (one for each form) containing item parameter estimates. The names of the rows of each matrix should be the names of the items. |

## Value

A matrix whose elements indicate the number of common items between the forms. On the diagonal of the matrix there are the number of items of each form.

## Author(s)

Michela Battauz

## See Also

est2pl, est3pl, estrasch

## Examples

```
data(est3pl)
linkp(coef = est3pl$coef)
```

---

`modIRT` *Estimated Coefficients and Covariance Matrix of IRT Models*

---

## Description

Creates an object of the class `modIRT` containing estimated coefficients and covariance matrices of IRT models. Rasch, one-parameter logistic, two-parameter logistic and three-parameter logistic models are included.

## Usage

```
modIRT(coef, var = NULL, names = NULL, ltparam = TRUE, lparam = TRUE,
display = TRUE, digits = 2)
```

## Arguments

| | |
|---|---|
| `coef` | list of matrices (one for each form) containing item parameter estimates. Guessing, difficulty and discrimination parameters should strictly be given in this order and they are contained in different columns of the matrix. The name of the rows of each matrix should be the name of the item. |
| `var` | list of matrices (one for each form) containing the covariance matrix of item parameter estimates. They should be given in the same order of coefficients. |
| `names` | character vector containing the names of the forms. This should have the same length of `coef` and `var`. If NULL, the names of the forms are assigned by function `modIRT`. |
| `ltparam` | logical; if TRUE the latent trait parameterization is used for difficulty parameters. See below for more details. |
| `lparam` | logical; if TRUE the logistic parameterization is used for guessing parameters. See below for more details. |
| `display` | logical; if TRUE coefficients and standard errors are printed. |
| `digits` | integer indicating the number of decimal places to be used if `display` is `TRUE`. |

## Details

`ltparam` and `lparam` refers the the parameterization used by the software used to estimate item parameters. The R package `ltm`, and the programs IRTPRO and flexMIRT use these parameterizations. If `ltparam` is `TRUE` the latent trait parameterization is used. Under this parameterization, the three-parameter logistic model is as follows

$$\pi_i = c_i + (1 - c_i)\frac{\exp(\beta_{1i} + \beta_{2i}z)}{1 + \exp(\beta_{1i} + \beta_{2i}z)},$$

where $\pi_i$ denotes the conditional probability of responding correctly to the $i$th item given $z$, $c_i$ denotes the guessing parameter, $\beta_{1i}$ is the easiness parameter, $\beta_{2i}$ is the discrimination parameter, and $z$ denotes the latent ability. The two-parameter logistic model, the one-parameter logistic model and the Rasch model present the same formulation. The two-parameter logistic model can be obtained by setting $c_i$ equal to zero, the one-parameter logistic model can be obtained by setting $c_i$ equal to zero and $\beta_{2i}$ costant across items, while the Rasch model can be obtained by setting $c_i$ equal to zero and $\beta_{2i}$ equal to 1.

If `lparam` is TRUE the guessing parameters are given under this parameterization

$$c_i = \frac{\exp(c_i^*)}{1 + \exp(c_i^*)}.$$

The `modIRT` function returns parameter estimates under the usual IRT parameterization, that is,

$$\pi_i = c_i + (1 - c_i)\frac{\exp[Da_i(\theta - b_i)]}{1 + \exp[Da_i(\theta - b_i)]},$$

where $Da_i = \beta_{2i}$, $b_i = -\beta_{1i}/\beta_{2i}$ and $\theta = z$.

If `ltparam` or `lparam` are TRUE, the covariance matrix is calculated using the delta method.

If item parameters are already given under the usual IRT parameterization, arguments `ltparam` and `lparam` should be set to `FALSE`.

## Value

An object of class `modIRT` consisting in a list with length equal to the number of forms containing lists with components

| | |
|---|---|
| `coefficients` | item parameter estimates. |
| `var` | covariance matrix of item parameter estimates. |
| `itmp` | number of item parameters of the IRT model. This is 1 for the Rasch model, 2 for the one-parameter logistic model, 2 for the two-parameter logistic model and 3 for the three-parameter logistic model. |

## Author(s)

Michela Battauz

## References

Bartholomew, D., Knott, M. and Moustaki, I. (2011) *Latent Variable Models and Factor Analysis: a Unified Approach*, 3rd ed. Wiley.

Rizopoulos, D. (2006). ltm: an R package for latent variable modelling and item response theory analyses. *Journal of Statistical Software*, **17**, 1–25.

## See Also

`direc`, `import.ltm`

## Examples

```
# three-parameter logistic model
data(est3pl)
test <- paste("test", 1:5, sep = "")
mod3pl <- modIRT(coef = est3pl$coef, var = est3pl$var, names = test, display = FALSE)

# two-parameter logistic model
data(est2pl)
test <- paste("test", 1:5, sep = "")
mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test, display = FALSE)

# Rasch model
data(estrasch)
test <- paste("test", 1:5, sep = "")
modrasch <- modIRT(coef = estrasch$coef, var = estrasch$var, names = test,
display = FALSE)

# one-parameter logistic model imported from the R package ltm
## Not run:
library(ltm)
mod1pl <- rasch(LSAT)
summary(mod1pl)
est.mod1pl <- import.ltm(mod1pl)
mod1pl.ltm <- modIRT(coef = list(est.mod1pl$coef), var = list(est.mod1pl$var), digits = 4)

## End(Not run)
```

---

`summary.ceqc` *Summarizing Estimated Chain Equating Coefficients*

---

## Description

`summary` method for class `ceqc`.

## Usage

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

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

## Arguments

| | |
|---|---|
| `object` | an object of the class `ceqc` returned by function `chainec`. |
| `x` | an object of class `summary.ceqc`, a result of a call to `summary.ceqc`. |
| `...` | further arguments passed to or from other methods. |

## Value

The function `summary.ceqc` returns a list with components

| | |
|---|---|
| `forms` | names of equated forms. |
| `method` | the equating method used. |
| `coefficients` | a $2 \times 2$ matrix with columns for the estimated coefficients A and B and standard errors. |

## Author(s)

Michela Battauz

## See Also

`chainec`

## Examples

```
# two-parameter logistic model
# direct equating coefficients using the "Haebara" method
data(est2pl)
test<-paste("test", 1:5, sep = "")
mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test, display = FALSE)
direclist2pl <- alldirec(mods = mod2pl, method = "Haebara")
# compute all chain equating coefficients of length 3
chainec3 <- chainec(r = 3, direclist = direclist2pl)
summary(chainec3$test1.test5.test4)
```

---

# summary.ceqclist *Summarizing a List of Estimated Chain Equating Coefficients*

---

## Description

`summary` method for class `ceqclist`.

## Usage

```
## S3 method for class 'ceqclist'
summary(object, path, ...)

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

### Arguments

| | |
|---|---|
| object | an object of the class ceqclist returned by function chainec. |
| path | a vector of character strings with the names of the forms that constitute the path separated by a dot (e.g. "test1.test2.test3"). |
| x | an object of class summary.ceqclist, a result of a call to summary.ceqclist. |
| ... | further arguments passed to or from other methods. |

### Value

The function summary.ceqclist returns a list containing the output of function summary.ceqc for each path contained in object.

### Author(s)

Michela Battauz

### See Also

chainec, summary.eqc

### Examples

```
# two-parameter logistic model
# direct equating coefficients using the "Haebara" method
data(est2pl)
test<-paste("test", 1:5, sep = "")
mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test, display = FALSE)
direclist2pl <- alldirec(mods = mod2pl, method = "Haebara")
# compute all chain equating coefficients of length 3
chainec3 <- chainec(r = 3, direclist = direclist2pl)
summary(chainec3)
summary(chainec3, path = "test1.test2.test3")
summary(chainec3, path = c("test1.test2.test3","test1.test5.test4"))
```

---

## summary.eqc *Summarizing Estimated Equating Coefficients*

---

### Description

summary method for class eqc.

### Usage

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

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

### Arguments

| | |
|---|---|
| object | an object of the class eqc returned by function direc. |
| x | an object of class summary.eqc, a result of a call to summary.eqc. |
| ... | further arguments passed to or from other methods. |

### Value

The function summary.eqc returns a list with components

| | |
|---|---|
| forms | names of equated forms. |
| method | the equating method used. |
| coefficients | a $2 \times 2$ matrix with columns for the estimated coefficients A and B and standard errors. |

### Author(s)

Michela Battauz

### See Also

direc

### Examples

```
# three-parameter logistic model
# direct equating coefficients between forms 1 and 2 using the Stocking-Lord method
data(est3pl)
test <- paste("test", 1:5, sep = "")
mod3pl <- modIRT(coef = est3pl$coef, var = est3pl$var, names = test, display = FALSE)
l12 <- direc(mod1 = mod3pl[1], mod2 = mod3pl[2], method = "Stocking-Lord")
summary(l12)
```

---

## summary.eqclist *Summarizing a List of Estimated Direct Equating Coefficients*

---

### Description

summary method for class eqclist.

### Usage

```
## S3 method for class 'eqclist'
summary(object, link, ...)

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

## Arguments

| | |
|---|---|
| object | an object of the class eqclist returned by function alldirec. |
| link | a vector of character strings with the names of the two forms being linked separated by a dot (e.g. "test1.test2"). |
| x | an object of class summary.eqclist, a result of a call to summary.eqclist. |
| ... | further arguments passed to or from other methods. |

## Value

The function summary.eqclist returns a list containing the output of function summary.eqc for each link contained in object.

## Author(s)

Michela Battauz

## See Also

alldirec, summary.eqc

## Examples

```
# Rasch model
# direct equating coefficients using the "mean-mean" method
data(estrasch)
test<-paste("test", 1:5, sep = "")
modrasch <- modIRT(coef = estrasch$coef, var = estrasch$var, names = test,
display = FALSE)
direclistrasch <- alldirec(mods = modrasch, method = "mean-mean", all = TRUE)
summary(direclistrasch)
summary(direclistrasch, link = "test1.test2")
summary(direclistrasch, link = c("test1.test2", "test1.test5"))
```

---

summary.meqc *Summarizing Bisector Equating Coefficients*

---

## Description

summary method for class meqc.

## Usage

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

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

## Arguments

| | |
|---|---|
| object | an object of the class meqc returned by function bisectorec. |
| x | an object of class summary.meqc, a result of a call to summary.meqc. |
| ... | further arguments passed to or from other methods. |

## Value

The function summary.meqc returns a list with components

| | |
|---|---|
| link | character vector with names of equated forms. |
| method | the equating method used. |
| coefficients | list of data frames containing Path, Estimate and StdErr of direct, chain and bisector equating coefficients. |

## Author(s)

Michela Battauz

## See Also

bisectorec

## Examples

```
# three-parameter logistic model
# direct equating coefficients using the "Stocking-Lord" method
data(est3pl)
test <- paste("test", 1:5, sep = "")
mod3pl <- modIRT(coef = est3pl$coef, var = est3pl$var, names = test, display = FALSE)
direclist3pl <- alldirec(mods = mod3pl, method = "Stocking-Lord")
# compute chain equating coefficients for path 1,2,3,4
pth1 <- paste("test", 1:4, sep = "")
pth1 <- data.frame(t(pth1), stringsAsFactors = FALSE)
chainec1 <- chainec(direclist = direclist3pl, pths = pth1)
# compute chain equating coefficients for path 1,5,4
pth2 <- c(paste("test", c(1,5,4), sep = ""))
pth2 <- data.frame(t(pth2), stringsAsFactors = FALSE)
chainec2 <- chainec(direclist = direclist3pl, pths = pth2)
# compute chain equating coefficients for path 1,2,3,4,5
pth3 <- paste("test", 1:5, sep = "")
pth3 <- data.frame(t(pth3), stringsAsFactors = FALSE)
chainec3 <- chainec(direclist = direclist3pl, pths = pth3)
# create a list of objects of class eqc or ceqc
ecall <- c(chainec1, chainec2, chainec3, direclist3pl["test1.test5"])
# compute bisector and weighted bisector coefficients
allec <- bisectorec(ecall = ecall, mods = mod3pl, weighted = TRUE, unweighted = TRUE)
summary(allec)
```

# Index