equateIRT: An R Package for IRT Test Equating

Michela Battauz
Università degli Studi di Udine

Abstract

This introduction to the R package **equateIRT** is a (slightly) modified version of Battauz (2015a), published in the *Journal of Statistical Software*.

The R package **equateIRT** implements item response theory (IRT) methods for equating different forms composed of dichotomous items. In particular, the IRT models included are the three-parameter logistic model, the two-parameter logistic model, the one-parameter logistic model and the Rasch model. Forms can be equated when they present common items (direct equating) or when they can be linked through a chain of forms that present common items in pairs (indirect or chain equating). When two forms can be equated through different paths, a single conversion can be obtained by averaging the equating coefficients. The package calculates direct and chain equating coefficients. The averaging of direct and chain coefficients that link the same two forms is performed through the bisector method. Furthermore, the package provides analytic standard errors of direct, chain and average equating coefficients.

Keywords: bisector, chain, equating, equating coefficients, IRT, Rasch model, standard errors.

1. Introduction

In many testing programs, security reasons require that test forms are composed of different items, making test scores not comparable across different administrations. The equating process permits the comparison of scores obtained from different forms taken. The term equating traditionally refers to the adjustment of scores from parallel forms, that are as similar as possible in content and statistical characteristics (Kolen and Brennan 2014, Chapter 1.1.2 and Chapter 1.2.3). The methods presented in this paper, implemented in the R (R Core Team 2014) package **equateIRT** (Battauz 2015b), allow more general forms of equating, such as horizontal and vertical scaling. For example, vertical scaling is intended to make scores comparable across different educational grades, where the content of the tests differs accordingly to the educational level. The package is available from the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/package=equateIRT`.

Various statistical methods have been proposed to perform equating (for a broad review, see Kolen and Brennan 2014). The R package **equateIRT** focuses on item response theory (IRT) methods for dichotomous items. IRT models are statistical models that have as response variable the responses given to the items of a questionnaire or test. These responses are modeled as a function of a latent trait, for example the ability level, and the item parameters that are related to some characteristics of the items, such as the difficulty. The purpose of IRT models is to provide a measure of the latent trait under investigation. For a broad

review of IRT models see van der Linden and Hambleton (1997). The IRT models included in the **equateIRT** package are the three-parameter logistic model, the two-parameter logistic model, the one-parameter logistic model and the Rasch model. When the parameters of the IRT model are estimated separately for each group of people taking a different test form, they are not comparable because the origin of the measurement scale is not identifiable. However, when two forms present a subset of common items, the parameters can be put on the same scale. There are two methods available to pursue this end. Concurrent calibration consists of estimating the item parameters together and yields measurements directly on a common metric. Alternatively, item parameters are estimated separately for each test form (separate calibration) and the estimates of the item parameters for the common items can be used to estimate the scale transformation. Both approaches can be extended to the case of multiple test forms, provided that the forms can be linked through common items. Concurrent calibration presents the advantage of not requiring any conversion after the estimation of the parameters as they are already expressed on the same scale. However, this approach requires the combination of the data from each form in a single dataset that could become quite large when there are many forms or when each form is administered to a very large number of people. In some cases, the large dimension of the dataset then makes the estimation very slow or even unfeasible. Instead, by equating the coefficients of separate calibrations, it is not necessary to estimate again the parameters of previous administrations, thus avoiding the construction of a single dataset.

The conversion of parameter estimates is attained by applying a linear transformation and the coefficients of this transformation are called equating coefficients. For each pair of forms containing common items, direct equating coefficients can be calculated. Suppose that Forms 1 and 2 have common items and that Forms 2 and 3 have common items. Forms 1 and 3 can then be equated employing the chain going through Form 2. In this case, indirect equating coefficients linking Forms 1 and 3 can be calculated as a function of direct equating coefficients linking pairs of forms with common items. In general, when two forms can be linked through a chain of forms that present common items in pairs, indirect equating coefficients can be calculated that permit the conversion of parameter estimates of one form into the scale of the other form using a linear transformation. These coefficients are a function of direct equating coefficients. Furthermore, some linkage designs are quite complex and two forms can be linked through different chains and possibly a direct link. For every path that links the same two forms the equating coefficients can be computed, thus yielding different scale conversions. In this case, the equating coefficients can be averaged in order to obtain a single transformation.

For the computation of direct equating coefficients, the **equateIRT** package implements methods based on moments of item parameters, that are the mean-sigma (Marco 1977), the mean-mean (Loyd and Hoover 1980), and the mean-geometric mean (Mislevy and Bock 1990) methods, and response function methods, that are the Haebara (Haebara 1980) and the Stocking-Lord (Stocking and Lord 1983) methods. The package computes also indirect equating coefficients through a chain of forms. The bisector method is used to average equating coefficients derived from different paths (Holland and Strawderman 2011; Battauz 2013).

Since the estimated equating coefficients are subject to sampling variation, it is important to assess the magnitude of this variability in order to evaluate the accuracy of the equating process (Ogasawara 2011). This objective can be accomplished by observing the asymptotic standard errors of the equating coefficients. Despite the importance of verifying the precision

of the equating performed, equating software generally does not provide standard errors of equating coefficients. To the best of my knowledge, just a few computer programs calculate standard errors based on bootstrap techniques. The computation of analytic standard errors of IRT equating coefficients is not implemented in any software, with the only exception being a set of subroutines that implement the methods developed by Ogasawara (2000) and Ogasawara (2001) for direct equating coefficients, available from the author. Instead, the **equateIRT** package provides analytical standard errors for direct, chain and average equating coefficients. It is important to note that analytical standard errors of equating coefficients can only be computed if the covariance matrix of item parameter estimates is available. This covariance matrix should be obtained from the software used to estimate the item parameters. The package provides functions to import data from **flexMIRT** (Cai 2013), **IRTPRO** (Cai, Thissen, and du Toit 2011) and the R packages **ltm** (Rizopoulos 2006) and **mirt** (Chalmers 2012). All of these IRT programs supply the covariance matrix of item parameter estimates.

Other R packages provide implementation of IRT equating methods. The package **irtoys** (Partchev 2014) performs IRT equating for dichotomous items and the package **plink** (Weeks 2010) implements unidimensional and multidimensional IRT equating for both dichotomous and polytomous items. The **plink** package performs chain equating by providing direct equating coefficients between forms that present common items, but does not provide indirect equating coefficients that permit the conversion from the first form into the last form of the path. Chain equating is not included in the **irtoys** package. Furthermore, the packages **irtoys** and **plink** do not provide average equating coefficients nor standard errors of direct, indirect and average coefficients.

The paper is structured as follows. The theory on IRT test equating is summarized in Section 2. Section 3 illustrates the **equateIRT** package and Section 4 concludes the paper.

2. IRT test equating

Consider a single test form that is denoted by g. In the three-parameter logistic model, the probability of a positive response on item j in form g for a person with ability θ is given by

$$p_{gj}(\theta_{(g)}; a_{gj}, b_{gj}, c_{gj}) = c_{gj} + (1 - c_{gj})\frac{\exp\left\{Da_{gj}(\theta_{(g)} - b_{gj})\right\}}{1 + \exp\left\{Da_{gj}(\theta_{(g)} - b_{gj})\right\}}, \tag{1}$$

where a_{gj} is the item discrimination parameter, b_{gj} is the item difficulty parameter, c_{gj} is the item guessing parameter and D is a constant typically set to 1.7. We define the parameter vector of form g as $\boldsymbol{\alpha}_g = (\boldsymbol{\alpha}_{g1}^\top, \ldots, \boldsymbol{\alpha}_{gn_g}^\top)^\top$, where $\boldsymbol{\alpha}_{gj} = (a_{gj}, b_{gj}, c_{gj})^\top, j = 1, \ldots, n_g$, and n_g is the number of items of form g. Item parameters are estimated separately for each form by using the marginal maximum likelihood method (Bock and Aitkin 1981), regarding the person parameter θ as a random variable with standard normal distribution.

2.1. Direct equating

Let $g-1$ be another form that presents $n_{g-1,g}$ items in common with Form g. The parameters estimated for Form $g-1$ can be transformed to the scale of Form g by using the following equations

$$\theta_g = A_{g-1,g}\,\theta_{g-1} + B_{g-1,g}\,,\tag{2}$$

$$a_g = \frac{a_{g-1}}{A_{g-1,g}},\tag{3}$$

$$b_g = A_{g-1,g}\,b_{g-1} + B_{g-1,g}\,,\tag{4}$$

where $A_{g-1,g}$ and $B_{g-1,g}$ are the equating coefficients. These coefficients can be estimated by using moments of item parameters (Kolen and Brennan 2014, Chapter 6.3.2), or response function methods (Kolen and Brennan 2014, Chapter 6.3.3). In any case, direct equating coefficients are estimated using only the item parameter estimates for the items in common between two forms, irrespective of the items in common with other forms.

The mean-sigma, mean-mean and mean-geometric mean methods are based on moments of item parameters. In particular, the equating coefficient $A_{g-1,g}$ is given by

$$A_{g-1,g} = \sqrt{\frac{\sum_{j=1}^{n_{g-1,g}} b_{gj}^2 - \frac{1}{n_{g-1,g}}\left(\sum_{j=1}^{n_{g-1,g}} b_{gj}\right)^2}{\sum_{j=1}^{n_{g-1,g}} b_{g-1j}^2 - \frac{1}{n_{g-1,g}}\left(\sum_{j=1}^{n_{g-1,g}} b_{g-1j}\right)^2}}\tag{5}$$

using the mean-sigma method,

$$A_{g-1,g} = \frac{\sum_{j=1}^{n_{g-1,g}} a_{g-1j}}{\sum_{j=1}^{n_{g-1,g}} a_{gj}}\tag{6}$$

using the mean-mean method, and

$$A_{g-1,g} = \left(\prod_{j=1}^{n_{g-1,g}} \frac{a_{g-1j}}{a_{gj}}\right)^{\frac{1}{n_{g-1,g}}}\tag{7}$$

using the mean-geometric mean method, while the equating coefficient $B_{g-1,g}$ is given by

$$B_{g-1,g} = \frac{1}{n_{g-1,g}} \sum_{j=1}^{n_{g-1,g}} b_{gj} - A_{g-1,g}\,\frac{1}{n_{g-1,g}} \sum_{j=1}^{n_{g-1,g}} b_{g-1j}\tag{8}$$

for all methods.

The Haebara and Stocking-Lord methods are based on the response function. The equating coefficients using the Haebara method are obtained by minimizing the following function

$$f_H(A_{g-1,g}, B_{g-1,g}) = \frac{1}{2}\int_{-\infty}^{+\infty} \sum_{j=1}^{n_{g-1,g}} \left[p_{gj}(\theta; a_{gj}, b_{gj}, c_{gj}) - p_{g-1j}(\theta; a_{g-1j}^*, b_{g-1j}^*, c_{g-1j})\right]^2 h(\theta)d\theta,\tag{9}$$

where $h(\theta)$ is the density of a standard normal variable, $a^*_{g-1j} = a_{g-1j}/A_{g-1,g}$ and $b^*_{g-1j} = A_{g-1,g} b_{g-1j} + B_{g-1,g}$. The Stocking-Lord method requires, instead, the minimization of the function

$$f_{SL}(A_{g-1,g}, B_{g-1,g}) = \frac{1}{2} \int_{-\infty}^{+\infty} \left\{ \sum_{j=1}^{n_{g-1,g}} \left[p_{gj}(\theta; a_{gj}, b_{gj}, c_{gj}) - p_{g-1j}(\theta; a^*_{g-1j}, b^*_{g-1j}, c_{g-1j}) \right] \right\}^2 h(\theta)d\theta. \tag{10}$$

The integrals in Equations 9 and 10 do not have an analytical solution and they are generally approximated using Gaussian quadrature.

Using the delta method, Ogasawara (2000) and Ogasawara (2001) derived the asymptotic covariance matrix for the vector of estimates $(\hat{A}_{g-1,g}, \hat{B}_{g-1,g})^\top$, that is given by

$$\mathsf{ACOV}(\hat{A}_{g-1,g}, \hat{B}_{g-1,g})^\top = \frac{\partial(A_{g-1,g}, B_{g-1,g})^\top}{\partial \boldsymbol{\alpha}_{g-1,g}^\top} \mathsf{ACOV}(\hat{\boldsymbol{\alpha}}_{g-1,g}) \frac{\partial(A_{g-1,g}, B_{g-1,g})}{\partial \boldsymbol{\alpha}_{g-1,g}}, \tag{11}$$

where $\boldsymbol{\alpha}_{g-1,g} = (\boldsymbol{\alpha}_g^\top, \boldsymbol{\alpha}_{g-1}^\top)^\top$ is a vector containing all the item parameters related to Forms $g-1$ and g, $\partial(A_{g-1,g}, B_{g-1,g})^\top/\partial \boldsymbol{\alpha}_{g-1,g}^\top$ is the matrix containing the partial derivatives of $A_{g-1,g}$ and $B_{g-1,g}$ with respect to the item parameters evaluated at their true values, and $\mathsf{ACOV}(\hat{\boldsymbol{\alpha}}_{g-1,g})$ is the asymptotic covariance matrix of $\hat{\boldsymbol{\alpha}}_{g-1,g}$. The derivatives depend on the method used to determine the equating coefficients and are given in Ogasawara (2000, 2011) for methods based on moments, and in Ogasawara (2001) for response function methods. The estimate of the asymptotic covariance matrix is obtained by inserting item parameter estimates into Equation 11.

2.2. Equating chains

Suppose that two forms are linked through a chain of forms that present common items in pairs. Define the path from Form 0 to Form l as $p = \{0, 1, \ldots, l\}$. According to Battauz (2013), it is possible to obtain the equating coefficients transforming the scale of θ_0 to that of θ_l as a function of the direct equating coefficients that link the forms with common items. These coefficients will be referred to as indirect or chain equating coefficients and they are given by

$$A_p = A_{0,1,\ldots,l} = \prod_{g=1}^{l} A_{g-1,g} \tag{12}$$

and

$$B_p = B_{0,1,\ldots,l} = \sum_{g=1}^{l} B_{g-1,g} A_{g,\ldots,l}, \tag{13}$$

where $A_{g,\ldots,l} = \prod_{h=g+1}^{l} A_{h-1,h}$ is the coefficient that links Form g to Form l.

Similarly to the case of a direct link, the delta method can be exploited to obtain the asymptotic covariance matrix of the vector of estimates $(\hat{A}_p, \hat{B}_p)^\top$, that is

$$\mathsf{ACOV}(\hat{A}_p, \hat{B}_p)^\top = \frac{\partial(A_p, B_p)^\top}{\partial \boldsymbol{\alpha}_p^\top} \mathsf{ACOV}(\hat{\boldsymbol{\alpha}}_p) \frac{\partial(A_p, B_p)}{\partial \boldsymbol{\alpha}_p}, \tag{14}$$

where $\boldsymbol{\alpha}_p = (\boldsymbol{\alpha}_0^\top, \boldsymbol{\alpha}_1^\top, \ldots, \boldsymbol{\alpha}_l^\top)^\top$ is the vector containing all the item parameters related to the forms that compose the path, $\partial(A_p, B_p)^\top / \partial \boldsymbol{\alpha}_p^\top$ is the matrix containing the partial derivatives of A_p and B_p with respect to the item parameters evaluated at their true values, and $\mathsf{ACOV}(\hat{\boldsymbol{\alpha}}_p)$ is the asymptotic covariance matrix of the estimate $\hat{\boldsymbol{\alpha}}_p$. The derivatives are given in Battauz (2013). The estimate of the asymptotic covariance matrix is obtained by inserting item parameter estimates into Equation 14.

The computation of chain equating coefficients is then performed in two steps: first, the calculation of direct equating coefficients between consecutive forms of a path and second, the determination of indirect equating coefficients as a function of direct equating coefficients. Thus, in this process, only the items in common between two consecutive forms of a given chain are used. If, for example, Forms 0 and 2 present common items, this information is not used to estimate the chain equating coefficients for path $p = \{0, 1, \ldots, l\}$. This link can be used to constitute a further path that connects Forms 0 and l and a different scale conversion can be derived for this path. In this case, there are two different equatings for the same two forms, and they can be averaged as explained in Section 2.3.

2.3. Average equating coefficients

Suppose that two forms are linked through different paths. Define the set of paths that link two Forms 0 and l as $\mathcal{P}_{0l}$ and the linking coefficients related to path p as A_p and B_p, $p \in \mathcal{P}_{0l}$. The equations that transform the scale of θ_0 to that of θ_l are then

$$\theta_l^p = A_p\,\theta_0 + B_p, \quad p \in \mathcal{P}_{0l}. \tag{15}$$

As observed by Kolen and Brennan (2014, p. 280) and Braun and Holland (1982, p. 44), the equating relationships provided by each path could be averaged to produce a single conversion that is expected to be more accurate. Battauz (2013) proposed using the bisector method, suggested by Holland and Strawderman (2011) to average the equating functions obtained by using different equating methods. The angle bisector, in case of two linear functions that intersect at a point, is the linear function that bisects the angle between them. The bisector method satisfies the symmetry property, which requires that the inverse function of the average equating function equals the average of the inverse functions. Instead, the (weighted) mean of the equating functions does not satisfy the symmetry property, making the bisector method preferable. The bisector method yields a weighted average of the linear transformations (15):

$$\theta_l^* = \sum_{p \in \mathcal{P}_{0l}} w_p\,\theta_l^p, \tag{16}$$

where

$$w_p = \frac{n_p(1 + A_p^2)^{-1/2}}{\sum_{b \in \mathcal{P}_{0l}} n_b(1 + A_b^2)^{-1/2}}, \tag{17}$$

and n_p are optional weights associated with each path. The average equating coefficients are then

$$A_{0l}^* = \sum_{p \in \mathcal{P}_{0l}} A_p w_p \tag{18}$$

and

$$B_{0l}^* = \sum_{p \in \mathcal{P}_{0l}} B_p w_p. \tag{19}$$

The asymptotic covariance matrix of the vector of estimates $(\hat{A}_{0l}^*, \hat{B}_{0l}^*)^\top$ can then be obtained by using the delta method as follows

$$\mathsf{ACOV}(\hat{A}_{0l}^*, \hat{B}_{0l}^*)^\top = \frac{\partial(A_{0l}^*, B_{0l}^*)^\top}{\partial\boldsymbol{\alpha}^\top}\mathsf{ACOV}(\hat{\boldsymbol{\alpha}})\frac{\partial(A_{0l}^*, B_{0l}^*)}{\partial\boldsymbol{\alpha}}, \tag{20}$$

where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_p)_{p \in \mathcal{P}_{0l}}$ is the vector containing all the item parameters used in the equating process in at least one of the paths in $\mathcal{P}_{0l}$, $\partial(A_{0l}^*, B_{0l}^*)^\top/\partial\boldsymbol{\alpha}^\top$ is the matrix containing the partial derivatives of A_{0l}^* and B_{0l}^* with respect to the item parameters evaluated at their true values, and $\mathsf{ACOV}(\hat{\boldsymbol{\alpha}})$ is a block diagonal matrix composed of all $\mathsf{ACOV}(\hat{\boldsymbol{\alpha}}_p)$, $p \in \mathcal{P}_{0l}$, and it is the asymptotic covariance matrix of the estimate $\hat{\boldsymbol{\alpha}}$. The derivatives are given in Battauz (2013). The estimate of the asymptotic covariance matrix is obtained by inserting item parameter estimates into Equation 20.

Obtaining average equating coefficients is then a process that takes place over three phases: First, the estimation of direct equating coefficients, then the calculation of chain equating coefficients for more than one path connecting two forms, and finally the computation of average equating coefficients. The distribution of common items across forms determines which forms can be directly linked in the first phase. From this follows the composition of paths connecting each pair of forms and the computation of chain equating coefficients. There are no restrictions on these connections: Different paths can share some parts, or the same item can be used to link more than one form. The relationship between the average coefficients and the item parameters used to compute them is reflected in the derivatives $\frac{\partial(A_{0l}^*, B_{0l}^*)^\top}{\partial\boldsymbol{\alpha}^\top}$, that determine the covariance matrix of the average equating coefficients.

A further issue concerns which weights n_p to use in averaging. Battauz (2013) proposed determining weights by minimizing the average variance of θ_l^*, namely

$$\mathsf{E}_{\theta_0}\left[\mathsf{VAR}(\hat{A}_{0l}^*\,\theta_0 + \hat{B}_{0l}^*|\theta_0)\right] = \mathsf{VAR}(\hat{A}_{0l}^*) + \mathsf{VAR}(\hat{B}_{0l}^*), \tag{21}$$

assuming that θ_0 has zero mean and variance equal to one.

3. The equateIRT package

In order to perform equating with the **equateIRT** package, it is necessary to have previously estimated item parameters. Item parameters can be estimated using R packages, or using external software and then import them into R. To calculate standard errors of equating coefficients it is also necessary to have estimated the covariance matrix of the item parameter estimates. If the program used to estimate the IRT model does not provide the covariance matrix of item parameter estimates or the user is not interested in standard errors of equating coefficients, the covariance matrix can be set to NULL.

3.1. Data preparation

To the best of my knowledge, the computer programs for IRT model estimation that export the covariance matrix of parameter estimates are **flexMIRT**, **IRTPRO** and the R packages **ltm** and **mirt**. The **equateIRT** package provides functions to import data from these programs. Item parameters and the covariance matrices can be also imported from other programs. In

this case, the user should import item parameter estimates and the covariance matrices using the general R function `read.table`.

For item parameter estimation, the programs **flexMIRT**, **IRTPRO** and the R packages **ltm** and **mirt** formulate the three-parameter logistic model as a latent trait model (Bartholomew, Knott, and Moustaki 2011),

$$\pi_j = c_j + (1 - c_j)\frac{\exp(\beta_{1j} + \beta_{2j}z)}{1 + \exp(\beta_{1j} + \beta_{2j}z)}, \tag{22}$$

where π_j is the probability of a positive response for the jth item, $\beta_{j1} = -D \cdot a_j \cdot b_j$ are easiness parameters, $\beta_{2j} = D \cdot a_j$ are discrimination parameters, c_j are guessing parameters and $z = \theta$ the latent ability. In the following, this parameterization will be referred to as the *latent trait parameterization*. The two-parameter logistic model, the one-parameter logistic model and the Rasch model can be presented with the same formulation. The two-parameter logistic model can be obtained by setting $c_j = 0$, the one-parameter logistic model can be obtained by constraining also β_{2j} to be constant across items, and the Rasch model can be obtained by setting $c_j = 0$ and $\beta_{2j} = 1$. Furthermore, for a three-parameter logistic model, if the guessing parameters are given under the parameterization

$$c_j = \frac{\exp(c_j^*)}{1 + \exp(c_j^*)}, \tag{23}$$

this will be called the *logistic parameterization* in the following. The IRT programs estimate the parameters $(c_j^*, \beta_{1j}, \beta_{2j})$, for every item j, and then calculate the item parameters of the usual IRT parameterization given in Equation 1 using Equation 23 and the equations

$$b_j = -\frac{\beta_{1j}}{Da_j} \qquad \text{and} \qquad a_j = \frac{\beta_{2j}}{D}. \tag{24}$$

The covariance matrix of parameter estimates exported by the programs is related to the vector of parameters $(c_j^*, \beta_{1j}, \beta_{2j})$. The covariance matrix of (c_j, b_j, a_j) can be obtained on the basis of the covariance matrix provided by the IRT programs by applying the delta method. The **equateIRT** package provides this functionality.

The functions provided by the **equateIRT** package to import item parameter estimates and the covariance matrix are `import.ltm`, `import.mirt`, `import.flexmirt` and `import.irtpro`. Since **ltm** and **mirt** are R packages, functions `import.ltm` and `import.mirt` just extract item parameter estimates and the covariance matrix from an object previously created with these packages. The arguments of the functions are:

mod: Output object from functions `rasch`, `ltm`, or `tpm` of the **ltm** package, or from function `mirt` of the **mirt** package.

display: Logical value indicating whether the coefficients and the standard errors should be printed. The default is `TRUE`.

digits: Integer value indicating the number of decimal digits to be used if display is `TRUE`.

Instead, functions `import.flexmirt` and `import.irtpro` read external files previously created with the programs **flexMIRT** or **IRTPRO**. The arguments of the functions are:

fnamep: The name of the file containing the item parameter estimates. This is a file whose name ends with "-prm.txt".

fnamev: The name of the file containing the covariance matrix of item parameter estimates. This is a file ending with "-cov.txt".

fnameirt: The name of the file containing additional information to link item parameters with the covariance matrix and to identify parameters that have been constrained to a fixed value. This is a file ending with "-irt.txt".

display: Logical value indicating whether the coefficients and the standard errors should be printed. The default is `TRUE`.

digits: Integer value indicating the number of decimal digits to be used if display is `TRUE`.

The functions return a list with components:

coef: The matrix with item parameter estimates.

var: The covariance matrix of item parameter estimates.

Item parameters are imported under the parameterization given in Equations 22 and 23. The usual IRT parameterization can be obtained later by using function `modIRT`. An example of importing data from the **ltm** package is given in Appendix A.

The **equateIRT** package does not handle mixed item types, but this issue can be easily overcome by using the more general model and introducing constraints on some parameters. For example, in the case of some item responses modeled as a two-parameter logistic model and others as a three-parameter logistic model, the user can specify a three-parameter logistic model for all items and fix the guessing parameter to zero for some items.

If item parameter estimates are obtained with IRT programs other than **flexMIRT**, **IRT-PRO**, **ltm** or **mirt**, the user has to create a matrix containing the item parameter estimates. These can be provided using typical IRT parameterization (1) or with parameterization (22) and/or (23). In this matrix guessing, difficulty and discrimination parameters should strictly be given in this order and they are contained in different columns of the matrix. It is important that the names of the rows of the matrix are the names of the items because this information is used to link different forms. The covariance matrix is only necessary if the user is interested in obtaining the standard errors of the equating coefficients. It is important that the order of the items in the covariance matrix is the same as in the matrix with the item parameter estimates. So, there are first guessing parameter, then difficulty parameters and finally discrimination parameters. An example of item parameters and covariance matrices organized in this way is given in the datasets contained in the **equateIRT** package. The package includes three simulated datasets for illustrative purposes containing item parameter estimates and covariance matrices of five forms. The item parameter estimates were obtained with package **ltm**. In particular, dataset `est3pl` contains parameters of a three-parameter logistic model, dataset `est2pl` contains parameters of a two-parameter logistic model and dataset `estrasch` contains parameters of a Rasch model. Each dataset is composed of a list of length 2 with components:

coef: A list of length 5 containing the matrices of item parameter estimates.

var: A list of length 5 containing the covariance matrices of item parameter estimates.

The following section explains how to obtain the equating coefficients.

3.2. Data analysis

In this section, we use the dataset `est2pl` to illustrate the use of the package. The code for loading the package and the data is

```
R> library("equateIRT")
R> data("est2pl", package = "equateIRT")
```

To perform the equating, it is necessary to reorganize the data using function `modIRT`. This function creates an object of class 'modIRT' consisting of a list with length equal to the number of forms where each element contains a lists with components:

coefficients: Item parameter estimates.

var: Covariance matrix of item parameter estimates.

itmp: Number of item parameters of the model.

The names of the forms can be specified in function `modIRT` using argument `names`. If `names` is not specified, function `modIRT` assigns the names. If item parameters are given under the latent trait parameterization, use option `ltparam = TRUE`. If guessing parameters of a three-parameter logistic model are given under the logistic parameterization, use option `lparam = TRUE`. The default values of both these arguments is `TRUE`, so the user does not need to specify them if these parameterizations were used in the estimation of the parameters. Using these options, the item parameters are returned under the usual IRT parameterization as in Equation 1. In this case, the covariance matrix of the item parameters under parameterization of model (1) is obtained, by using the delta method, on the basis of the covariance matrix of item parameters under the parameterizations (22) and/or (23), supplied by the user. If item parameters already conform to the traditional parameterization, the options `ltparam` and `lparam` should be set to `FALSE`, and the function does not perform any transformation. Argument `coef` is used to specify the item parameter estimates. They should be given as a list of matrices (one for each form) whose row names are the names of the items. Argument `var` can be used to specify the covariance matrix of item parameter estimates and it should be a list of matrices. If it is left equal to `NULL`, standard errors of equating coefficients will not be computed. Option `display = TRUE` can be used to print item parameter estimates and the corresponding standard errors in order to compare them with those provided by the software used to estimate the item parameters. The coefficients can be extracted using the `coef` method.

```
R> test <- paste("test", 1:5, sep = "")
R> mod2pl <- modIRT(coef = est2pl$coef, var = est2pl$var, names = test,
+ display = FALSE)
R> coef(mod2pl$test1)[1:5]
```

```
  Dffclt.I1   Dffclt.I2   Dffclt.I3   Dffclt.I4   Dffclt.I5
 0.05774085  0.02753964  0.07275128  0.41568210 -0.00716265
```

The linkage plan can be inspected using the function `linkp`. Given a list of item parameter estimates with item labels as row names, this function computes the number of common items between all pairs of forms and returns 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.

```
R> linkp(coef = est2pl$coef)
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]   20   10    0    0   10
[2,]   10   20   10    0    0
[3,]    0   10   20   10    0
[4,]    0    0   10   20   10
[5,]   10    0    0   10   20
```

The output of the function shows that every form is composed of 20 items and presents 10 items in common with adjacent forms. Furthermore, Forms 1 and 5 present 10 common items.

Function `direc` calculates direct equating coefficients between two forms with common items. Arguments `mod1` and `mod2` are objects of class 'modIRT' containing item parameter coefficients and their covariance matrix. Argument `method` indicates the equating method to use and should be one of `"mean-mean"`, `"mean-gmean"`, `"mean-sigma"`, `"Haebara"` or `"Stocking-Lord"`. For example, to calculate direct equating coefficients between Form 1 and Form 5 using the Haebara method for the `est2pl` dataset the code is

```
R> l15 <- direc(mod1 = mod2pl[1], mod2 = mod2pl[5], method = "Haebara")
R> l15
```

```
Direct equating coefficients
Method: Haebara
Link: test1.test5
```

The `direc` function approximates the integrals in Equations 9 and 10 using a Gauss-Hermite quadrature with 30 points by default. The number of quadrature points can be modified using argument `nq`. Alternatively, setting `quadrature = FALSE` the integrals are 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. Argument `D` can be used to specify the value of the constant D in model (1) used in the estimation of item parameters. A `summary` method for displaying the output of the function is available.

```
R> summary(l15)
```

```
Link: test1.test5
Method: Haebara
```

```
Equating coefficients:
  Estimate    StdErr
A  0.97152 0.021344
B -0.47511 0.023886
```

The package provides also function `alldirec` to calculate direct equating coefficients between all pairs of forms that present common items. Using the mean-mean method, the code for our example is

```
R> direclist2pl <- alldirec(mods = mod2pl, method = "mean-mean")
R> direclist2pl
```

```
Direct equating coefficients
Method: mean-mean
Links:
test1.test2
test1.test5
test2.test1
test2.test3
test3.test2
test3.test4
test4.test3
test4.test5
test5.test1
test5.test4
```

The `summary` function can be used to display all the equatings performed

```
R> summary(direclist2pl)
```

or just one of them

```
R> summary(direclist2pl, "test1.test5")
```

```
Link: test1.test5
Method: mean-mean
Equating coefficients:
  Estimate    StdErr
A  0.96497 0.021313
B -0.47418 0.024224
```

In order to compute chain equating coefficients, it is necessary to have previously computed direct equating coefficients using function `alldirec`. Chain equating coefficients can be computed using function `chainec`. This function requires the specification of the length of the chain using argument `r`. For example, to compute all chain equating coefficients of length 4 for the `est2pl` dataset the code is

```
R> cec4 <- chainec(r = 4, direclist = direclist2pl)
R> cec4
```

```
Chain equating coefficients
Method: mean-mean
Paths:
test4.test5.test1.test2
test3.test2.test1.test5
test5.test1.test2.test3
test4.test3.test2.test1
test5.test4.test3.test2
test1.test2.test3.test4
test1.test5.test4.test3
test2.test3.test4.test5
test2.test1.test5.test4
test3.test4.test5.test1
```

The summary function displays all the equatings performed.

```
R> summary(cec4)
```

It is also possible to display only one equating using the following code

```
R> summary(cec4, "test1.test2.test3.test4")
```

```
Path: test1.test2.test3.test4
Method: mean-mean
Equating coefficients:
  Estimate    StdErr
A  1.25323 0.046518
B -0.49789 0.038505
```

Option `f1` can be used to restrict the equatings to those chains that start from the same form. To consider all chain equatings of length 4 that start from Form 1 the code is

```
R> cec4.1 <- chainec(r = 4, direclist = direclist2pl, f1 = "test1")
```

Option `f2` can be used to restrict the equatings to those chains that end with the same form. To consider all chain equatings of length 4 that start from Form 1 and end with Form 4 the code is

```
R> cec1234 <- chainec(r = 4, direclist = direclist2pl, f1 = "test1",
+ f2 = "test4")
```

Alternatively, it is possible to specify one or more particular paths using argument `pths`. For example, to compute chain equating coefficients between Forms 1 and 4 using path $\{1, 5, 4\}$ the code is

```
R> pth1 <- paste("test", c(1, 5, 4), sep = "")
R> pth1 <- data.frame(t(pth1), stringsAsFactors = FALSE)
R> cec154 <- chainec(direclist = direclist2pl, pths = pth1)
R> summary(cec154)
```

```
Path: test1.test5.test4
Method: mean-mean
Equating coefficients:
  Estimate   StdErr
A  1.15857 0.033390
B -0.39971 0.033076
```

Another example is the chain equating of Forms 1 and 5 using path $\{1, 2, 3, 4, 5\}$.

```
R> pth2 <- paste("test", 1:5, sep = "")
R> pth2 <- data.frame(t(pth2), stringsAsFactors = FALSE)
R> cec12345 <- chainec(direclist = direclist2pl, pths = pth2)
R> summary(cec12345)
```

```
Path: test1.test2.test3.test4.test5
Method: mean-mean
Equating coefficients:
  Estimate   StdErr
A  1.04381 0.043488
B -0.55595 0.037102
```

When two forms can be linked through different paths, the equating coefficients can be averaged using function `bisectorec`, that implements the bisector method. Options `weighted = TRUE` and `unweighted = TRUE` can be used to calculate the weighted bisector coefficients and the unweighted bisector coefficients. Weights are determined in order to minimize function (21). For example, to calculate bisector and weighted bisector coefficients to equate Forms 1 and 4 through paths $\{1, 2, 3, 4\}$ and $\{1, 5, 4\}$, and Forms 1 and 5 through paths $\{1, 2, 3, 4, 5\}$ and $\{1, 5\}$ the code is

```
R> ecall <- c(cec1234, cec154, cec12345, direclist2pl["test1.test5"])
R> fec <- bisectorec(ecall = ecall, weighted = TRUE, unweighted = TRUE)
R> fec
```

```
Bisector and weighted bisector equating coefficients
Method: mean-mean

Link: test1.test4
 Paths:
  test1.test2.test3.test4
  test1.test5.test4

Link: test1.test5
```

```
Paths:
 test1.test2.test3.test4.test5
 test1.test5
```

The `summary` function displays the results.

```
R> summary(fec)
```

```
Link: test1.test4
Method: mean-mean
Equating coefficients:
                       Path Estimate    StdErr
 A test1.test2.test3.test4  1.25323 0.046518
 A       test1.test5.test4  1.15857 0.033390
 A                bisector  1.20480 0.030609
 A       weighted bisector  1.18885 0.029302
 B test1.test2.test3.test4 -0.49789 0.038505
 B       test1.test5.test4 -0.39971 0.033076
 B                bisector -0.44766 0.029927
 B       weighted bisector -0.43112 0.029713

Link: test1.test5
Method: mean-mean
Equating coefficients:
                             Path Estimate    StdErr
 A test1.test2.test3.test4.test5  1.04381 0.043488
 A                   test1.test5  0.96497 0.021313
 A                      bisector  1.00361 0.025683
 A             weighted bisector  0.97533 0.020528
 B test1.test2.test3.test4.test5 -0.55595 0.037102
 B                   test1.test5 -0.47418 0.024224
 B                      bisector -0.51426 0.026383
 B             weighted bisector -0.48493 0.023780
```

Function `eqc` can be used to extract a data frame containing the equating coefficients. For direct equating coefficients the code is

```
R> eqc(l15)
```

```
        link         A          B
1 test1.test5 0.9715194 -0.4751087
```

while, for a list of direct equating coefficients, the code is

```
R> eqc(direclist2pl)
```

```
           link         A          B
1   test1.test2 1.2100375 -0.1419891
2   test1.test5 0.9649660 -0.4741816
3   test2.test1 0.8264207  0.1173427
4   test2.test3 1.0136460 -0.1365059
5   test3.test2 0.9865377  0.1346682
6   test3.test4 1.0217563 -0.2113561
7   test4.test3 0.9787069  0.2068557
8   test4.test5 0.8328952 -0.1412610
9   test5.test1 1.0363059  0.4913972
10  test5.test4 1.2006313  0.1696024
```

It is also possible to select a specific link.

```
R> eqc(direclist2pl, link = "test1.test5")
```

```
        link        A          B
1 test1.test5 0.964966 -0.4741816
```

For a list of chain equating coefficients it is possible to display the equating coefficients for all paths, or specify a particular link and/or path. The code is as follows

```
R> eqc(cec4)
```

```
           link                       path          A          B
1   test4.test2 test4.test5.test1.test2 1.0444247   0.2754830
2   test3.test5 test3.test2.test1.test5 0.7867321  -0.2535563
3   test5.test3 test5.test1.test2.test3 1.2710807   0.3222905
4   test4.test1 test4.test3.test2.test1 0.7979350   0.3972838
5   test5.test2 test5.test4.test3.test2 1.1592470   0.5024956
6   test1.test4 test1.test2.test3.test4 1.2532349  -0.4978899
7   test1.test3 test1.test5.test4.test3 1.1338989  -0.1843480
8   test2.test5 test2.test3.test4.test5 0.8626289  -0.4334672
9   test2.test4 test2.test1.test5.test4 0.9574649  -0.2637653
10  test3.test1 test3.test4.test5.test1 0.8819128   0.1625789
```

```
R> eqc(cec4, path = "test1.test2.test3.test4")
```

```
        link                       path         A          B
1 test1.test4 test1.test2.test3.test4 1.253235 -0.4978899
```

Also for the output of function `bisectorec`, it is possible to extract a data frame containing all the coefficients, or select the link and/or the path.

```
R> eqc(fec)
```

```
          link                             path          A          B
1 test1.test4          test1.test2.test3.test4 1.2532349 -0.4978899
2 test1.test4                  test1.test5.test4 1.1585684 -0.3997148
3 test1.test4                           bisector 1.2048012 -0.4476611
4 test1.test4                  weighted bisector 1.1888525 -0.4311214
5 test1.test5 test1.test2.test3.test4.test5 1.0438133 -0.5559511
6 test1.test5                        test1.test5 0.9649660 -0.4741816
7 test1.test5                           bisector 1.0036128 -0.5142608
8 test1.test5                  weighted bisector 0.9753335 -0.4849333
```

```
R> eqc(fec, link = "test1.test4", path = "bisector")
```

```
          link     path        A          B
1 test1.test4 bisector 1.204801 -0.4476611
```

Functions `direc`, `alldirec` and `chainec` provide also the scale conversion of item parameter estimates. This is included in the data frame `tab`, that is part of the outputs of the functions. Function `itm` extracts this data frame. Some examples for direct and chain equating coefficients are given below.

```
R> itm(l15)[1:3, ]
```

```
        Item       test1 test5 test1.as.test5 select
1  Dffclt.I1 0.05774085    NA     -0.4190124      0
2 Dffclt.I10 0.65469024    NA      0.1609356      0
3  Dffclt.I2 0.02753964    NA     -0.4483534      0
```

```
R> itm(direclist2pl, "test1.test5")[1:3, ]
```

```
        Item       test1 test5 test1.as.test5 select
1  Dffclt.I1 0.05774085    NA     -0.4184636      0
2 Dffclt.I10 0.65469024    NA      0.1575722      0
3  Dffclt.I2 0.02753964    NA     -0.4476068      0
```

```
R> itm(cec12345, "test1.test2.test3.test4.test5")[1:3, ]
```

```
        Item       test1 test5 test1.as.test5
1  Dffclt.I1 0.05774085    NA     -0.4956804
2 Dffclt.I10 0.65469024    NA      0.1274233
3  Dffclt.I2 0.02753964    NA     -0.5272049
```

The transformation of item parameter estimates can be found under column `test1.as.test5`.

Since the `bisectorec` function returns the equating coefficients of a plurality of scale conversions, it does not provide the scale conversion of the item parameters. For average equating coefficients the user can employ function `convert`, that is a specific function to convert item and person parameters, given the equating coefficients. The following code extracts the equating coefficients obtained with the bisector method to transform from the scale of Form 1 to the scale of Form 4 and performs the scale conversion of item parameters of Form 1 and of an hypothetical vector of person parameters.

```
R> eqc14 <- eqc(fec, link = "test1.test4", path = "bisector")
R> convert(A = eqc14$A, B = eqc14$B, coef = coef(mod2pl$test1),
+ person.par = seq(-3, 3, 0.5))

$coef
  Dffclt.I1    Dffclt.I2    Dffclt.I3    Dffclt.I4    Dffclt.I5
-0.37809488 -0.41448133 -0.36001029  0.05315316 -0.45629069
  Dffclt.I6    Dffclt.I7    Dffclt.I8    Dffclt.I9    Dffclt.I10
-1.37992251 -0.26139414 -0.04166382 -0.66801293  0.34111044
 Dffclt.I31   Dffclt.I32   Dffclt.I33   Dffclt.I34   Dffclt.I35
 0.18177450  0.55392509  0.24538460 -0.86040329  0.06882968
 Dffclt.I36   Dffclt.I37   Dffclt.I38   Dffclt.I39   Dffclt.I40
 0.57880081  0.10200975 -0.61407467  0.17753357  0.56645207
  Dscrmn.I1    Dscrmn.I2    Dscrmn.I3    Dscrmn.I4    Dscrmn.I5
 0.89322180  0.93158897  0.90584958  1.06301718  0.83608708
  Dscrmn.I6    Dscrmn.I7    Dscrmn.I8    Dscrmn.I9    Dscrmn.I10
 0.78740779  0.87759511  0.99655944  0.81101215  1.11236175
 Dscrmn.I31   Dscrmn.I32   Dscrmn.I33   Dscrmn.I34   Dscrmn.I35
 1.23076607  1.07889788  1.12225053  1.14717253  1.05509303
 Dscrmn.I36   Dscrmn.I37   Dscrmn.I38   Dscrmn.I39   Dscrmn.I40
 0.83680267  1.06871161  1.22268701  1.11314169  1.20883896

$person.par
 [1] -4.0620646 -3.4596641 -2.8572635 -2.2548629 -1.6524623 -1.0500617
 [7] -0.4476611  0.1547395  0.7571401  1.3595406  1.9619412  2.5643418
[13]  3.1667424
```

Function `convert` can also be used for direct or chain equating parameters.

4. Conclusion

The linkage plans can be rather complex involving many forms, several links, chains and the connection of forms through more than one path. This situation necessitates the performance of chain equating in order to link forms that do not present common items. Furthermore, when two forms can be connected through different paths it can be useful to synthesize the conversions obtained into a single one. The **equateIRT** package includes not only the computation of direct equating coefficients but also permits the computation of chain and average equating coefficients. Another important and exclusive feature of the **equateIRT** package is the provision of analytical standard errors of equating coefficients. Standard errors are an important tool for assessing the accuracy of the equating process and can be also used to perform further inferential analysis.

Acknowledgments

References

Bartholomew DJ, Knott M, Moustaki I (2011). *Latent Variable Models and Factor Analysis: A Unified Approach.* 3rd edition. John Wiley & Sons, New York. doi:10.1002/9781119970583.

Battauz M (2013). "IRT Test Equating in Complex Linkage Plans." *Psychometrika*, **78**(3), 464–480. doi:10.1007/s11336-012-9316-y.

Battauz M (2015a). "equateIRT: An R Package for IRT Test Equating." *Journal of Statistical Software*, **68**(7), 1–22. doi:10.18637/jss.v068.i07.

Battauz M (2015b). **equateIRT***: Direct, Chain and Average Equating Coefficients with Standard Errors Using IRT Methods.* R package version 1.2-2, URL http://CRAN.R-project.org/package=equateIRT.

Bock RD, Aitkin M (1981). "Marginal Maximum Likelihood Estimation of Item Parameters: Application of an EM Algorithm." *Psychometrika*, **46**(4), 443–459. doi:10.1007/bf02293801.

Braun HI, Holland PW (1982). "Observed-Score Test Equating: A Mathematical Analysis of Some ETS Equating Procedures." In PW Holland, DB Rubin (eds.), *Test Equating*, pp. 9–49. Academic, New York.

Cai L (2013). **flexMIRT** *Version 2: Flexible Multilevel Multidimensional Item Analysis and Test Scoring.* Chapel Hill, NC. Computer Software.

Cai L, Thissen D, du Toit SHC (2011). **IRTPRO***: Flexible, Multidimensional, Multiple Categorical IRT Modeling.* Chicago. Computer Software.

Chalmers RP (2012). "**mirt**: A Multidimensional Item Response Theory Package for the R Environment." *Journal of Statistical Software*, **48**(6), 1–29. doi:10.18637/jss.v048.i06.

Haebara T (1980). "Equating Logistic Ability Scales by a Weighted Least Squares Method." *Japanese Psychological Research*, **22**(3), 144–149.

Holland PW, Strawderman WE (2011). "How to Average Equating Functions If You Must." In AA von Davier (ed.), *Statistical Models for Test Equating, Scaling, and Linking*, pp. 89–107. Springer-Verlag, New York.

Kolen MJ, Brennan RL (2014). *Test Equating, Scaling, and Linking: Methods and Practices.* 3rd edition. Springer-Verlag, New York.

Loyd BH, Hoover HD (1980). "Vertical Equating Using the Rasch Model." *Journal of Educational Measurement*, **17**(3), 179–193. doi:10.1111/j.1745-3984.1980.tb00825.x.

Marco GL (1977). "Item Characteristic Curve Solutions to Three Intractable Testing Problems." *Journal of Educational Measurement*, **14**(2), 139–160. doi:10.1111/j.1745-3984.1977.tb00033.x.

Mislevy RJ, Bock RD (1990). **BILOG 3***. Item Analysis and Test Scoring with Binary Logistic Models.* Scientific Software International, Mooresville, IN.

Ogasawara H (2000). "Asymptotic Standard Errors of IRT Equating Coefficients Using Moments." *Economic Review (Otaru University of Commerce)*, **51**(1), 1–23.

Ogasawara H (2001). "Standard Errors of Item Response Theory Equating/Linking by Response Function Methods." *Applied Psychological Measurement*, **25**(1), 53–67. `doi:10.1177/01466216010251004`.

Ogasawara H (2011). "Applications of Asymptotic Expansion in Item Response Theory Linking." In AA von Davier (ed.), *Statistical Models for Test Equating, Scaling, and Linking.* Springer-Verlag, New York.

Partchev I (2014). **irtoys**: *Simple Interface to the Estimation and Plotting of IRT Models.* R package version 0.1.7, URL `http://CRAN.R-project.org/package=irtoys`.

R Core Team (2014). R: *A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org`.

Rizopoulos D (2006). "**ltm**: An R Package for Latent Variable Modelling and Item Response Theory Analyses." *Journal of Statistical Software*, **17**(5), 1–25. `doi:10.18637/jss.v017.i05`.

Stocking ML, Lord ML (1983). "Developing a Common Metric in Item Response Theory." *Applied Psychological Measurement*, **7**(2), 201–210. `doi:10.1177/014662168300700208`.

van der Linden WJ, Hambleton RK (1997). *Handbook of Modern Item Response Theory.* Springer-Verlag, New York. `doi:10.1007/978-1-4757-2691-6`.

Weeks JP (2010). "**plink**: An R Package for Linking Mixed-Format Tests Using IRT-Based Methods." *Journal of Statistical Software*, **35**(12), 1–33. `doi:10.18637/jss.v035.i12`.

A. Importing data from the ltm package

This appendix shows how to import the estimates obtained with the **ltm** package using the dataset `data2pl`. This dataset is a list of length five and includes five simulated data frames generated from a two-parameter logistic model. Each data frame contains 5000 dichotomous responses to 20 items. Item parameter estimates and covariance matrices included in the dataset `est2pl` are obtained from the dataset `data2pl`.

The code for loading packages **ltm** and **equateIRT** and the data, and for estimating a two-parameter logistic model is

```
R> library("ltm")
R> library("equateIRT")
R> data("data2pl", package = "equateIRT")
R> m1 <- ltm(data2pl[[1]] ~ z1)
R> m2 <- ltm(data2pl[[2]] ~ z1)
R> m3 <- ltm(data2pl[[3]] ~ z1)
R> m4 <- ltm(data2pl[[4]] ~ z1)
R> m5 <- ltm(data2pl[[5]] ~ z1)
```

Function `import.ltm` extracts the item parameter estimates and the covariance matrix.

```
R> estm1 <- import.ltm(m1, display = FALSE)
R> estm2 <- import.ltm(m2, display = FALSE)
R> estm3 <- import.ltm(m3, display = FALSE)
R> estm4 <- import.ltm(m4, display = FALSE)
R> estm5 <- import.ltm(m5, display = FALSE)
R> estm1$coef[1:3, ]

   (Intercept)       z1
I1 -0.06213808 1.076155
I2 -0.03090993 1.122379
I3 -0.07939847 1.091369

R> estm1$var[1:3,1:3]

             [,1]          [,2]          [,3]
[1,] 0.0012285184 0.0002460322 0.0002391000
[2,] 0.0002460322 0.0012628923 0.0002495126
[3,] 0.0002391000 0.0002495126 0.0012407430
```

The item parameters and the corresponding covariance matrix are given under parameterizations (22) and (23). The function `modIRT` transforms item parameters into the usual IRT parameterization given in Equation 1 and returns item parameter estimates and the covariance matrices in a format suitable for running the equating.

```
R> estc <- list(estm1$coef, estm2$coef, estm3$coef, estm4$coef, estm5$coef)
R> estv <- list(estm1$var, estm2$var, estm3$var, estm4$var, estm5$var)
R> mod2pl.ltm <- modIRT(coef = estc, var = estv, display = FALSE)
```

Affiliation:

Michela Battauz
Department of Economics and Statistics
Università degli Studi di Udine
via Tomadini 30/A
33100 Udine, Italy
E-mail: michela.battauz@uniud.it
URL: https://people.uniud.it/page/michela.battauz