The R package kequate enables observed-score equating using the kernel method of test equating. We present the recent developments of kequate, which provide additional support for item-response theory observed score equating using 2-PL and 3-PL models in the equivalent groups design and non-equivalent groups with anchor test design using chain equating. The implementation also allows for local equating using IRT observed-score equating. Support is provided for the R package ltm.

Keywords: kernel equating, observed-score test equating, item-response theory, R.

1. Introduction

The kernel method of test equating (von Davier, Holland, and Thayer 2004) is a flexible observed-score equating framework which enables the equating of two tests using all common equating designs. Kernel equating has usually been described using pre-smoothing through log-linear models but the framework provides support for input data of various types, such as observed data and data derived from IRT models. Here, we focus on IRT observed-score equating in the kernel method of test equating. We introduce IRT observed-score kernel equating in the equivalent groups (EG) design and non-equivalent groups with anchor test (NEAT) design using chain equating (CE) and illustrate how to conduct these equating methods using the R (R Development Core Team 2013) package kequate (Andersson, Bränberg, and Wiberg 2013). It is also shown how local equating using IRT observed-score equating van der Linden (2011) can be conducted in kequate.

This document has the following structure. In Section 2, IRT observed-score equating in the kernel equating framework is described and in Section 3 the implementation of IRT observed-score equating in kequate is introduced. In Section 4 examples of the available methods of IRT observed-score equating in kequate are given and in Section 5 future additions to the package are presented.

2. IRT observed-score kernel equating

The kernel equating framework enables the usage of score probabilities which are either observed or estimated using a statistical model. Typically the kernel equating framework has utilized score probabilities derived from log-linear models (Holland, King, and Thayer 1989;
von Davier et al. 2004; Lee and von Davier 2011). The usage of score probabilities derived from IRT models, which would enable IRT observed-score equating, has been suggested (von Davier 2010) but has not been described in the literature. IRT observed-score equating has however been described in traditional equipercentile equating using linear interpolation (Lord and Wingersky 1984; Kolen and Brennan 2004). The asymptotic standard errors of equating for IRT observed-score equating in various NEAT designs were given in Ogasawara (2003). For kernel equating, the necessary components are the covariance matrices of the score probabilities which are needed to calculate the asymptotic standard errors of equating. In this section we show how the results of Ogasawara (2003) can be applied in the kernel equating framework for the NEAT CE design in the case of an external anchor test under the three parameter logistic model (3-PL). The results for the EG design and when using the two parameter logistic model (2-PL) are similar, but simpler, and are therefore omitted.

2.1. IRT observed-score kernel equating in the NEAT CE design

Let $X$ and $Y$ denote two tests, each with $k$ number of items. For the sake of simplicity we assume an equal number of items on the tests in this section but the results apply to the case where the number of items are not equal and the implementation in kequate allows for a non-equal number of items. The tests consist of $k$ unique items and $k_A$ common items. Denote the subtests of unique items $X^*$ and $Y^*$ and the subtest of common items $A$. Each test is administered to a separate group of test takers each from a separate population. Denote the populations $P$ and $Q$, respectively, with samples sizes $n$ and $m$ for the respective test groups.

Let $\Theta_P$ and $\Theta_Q$ be the random variables corresponding to the ability level of a member of the population from which each test taker for tests $X$ and $Y$ is taken. Now, let $P_{Xl}(\theta_P)$ and $P_{Yl}(\theta_Q)$ be the probabilities to answer item $l$ of tests $X$ and $Y$ correctly, viewed as a functions of the ability levels $\theta_P$ and $\theta_Q$. With the 3-PL model we have that

$$P_{Xl}(\theta_P) = c_{xl} + \frac{1 - c_{xl}}{1 + \exp[-a_{xl}(\theta_P - b_{xl})]},$$

(1)

where $a_{xl}$ is the discrimination parameter for item $l$, $b_{xl}$ is the difficulty parameter for item $l$ and $c_{xl}$ is the guessing parameter for item $l$ (Ogasawara 2003). $P_{Yl}(\theta_Q)$ is defined analogously. The 2-PL model is also defined by Equation 1, if $c_{xl} = 0$. Hence with the 3-PL model we have a total of $3k$ number of parameters across all items for tests $X$ and $Y$ respectively. Let $\alpha_X$ and $\alpha_Y$ denote the $1 \times 3k$ vectors of all item parameters for tests $X$ and $Y$.

We define $\beta_{X,x}(\theta_P)$ and $\beta_{Y,y}(\theta_Q)$ as the probabilities to obtain score values $x, y \in \{0, 1, \ldots, k\}$ on tests $X$ and $Y$, respectively, as a function of the ability levels $\theta_P$ and $\theta_Q$. Similarly, we define $\beta_{X^*,x^*}(\theta_P)$ and $\beta_{Y^*,y^*}(\theta_Q)$ as the probabilities to obtain the score values $x^*, y^* \in \{0, 1, \ldots, k^*\}$ and $\beta_{A,0}(\theta_P)$ and $\beta_{AQ,0}(\theta_Q)$ as the probabilities to obtain the score values $a \in \{0, 1, \ldots, k_A\}$. These probabilities can be obtained by using the procedure outlined in Lord and Wingersky (1984).

Now, let $\beta_{X^*,x^*}$, $\beta_{Y^*,y^*}$, $\beta_{A,0}$ and $\beta_{AQ,0}$ be the probabilities to obtain score values $x^*, y^*$ and $a$ across all ability levels and let $\beta_X$ and $\beta_Y$ be the $1 \times (k + 1)$ vectors of probabilities $\beta_{X^*,x^*}$ and $\beta_{Y^*,y^*}$ to obtain each of the score values $x^*, y^* \in \{0, 1, \ldots, k^*\}$ on the tests $X^*$ and $Y^*$ and let $\beta_{A,0}$ and $\beta_{AQ,0}$ be the $1 \times (k_A + 1)$ vectors of probabilities $\beta_{A,0}$ and $\beta_{AQ,0}$ to
obtain each of the score values \( a \in \{0, 1, \ldots, k_A\} \) on test A. We have that

\[
\beta_{X^*, x^*} \approx \sum_{r=1}^{R} \beta_{X, x^*}(t_r) W(t_r),
\]

where \( t_r \) denotes the ability level for the \( r \)-th quadrature point, \( r \in \{1, 2, \ldots, R\} \), and where \( W(\cdot) \) is a weight function such that each quadrature point is weighted in accordance with the assumptions made about the distribution of the ability level. Corresponding expressions apply for \( \beta_{Y^*, y^*} \), \( \beta_{AP,a} \) and \( \beta_{AQ,a} \). We are interested in finding \( \Sigma(\beta_{X^*,AP}) \) and \( \Sigma(\beta_{Y^*,AQ}) \). The results are of the same form for both \((\beta_{X^*,AP})\) and \((\beta_{Y^*,AQ})\) so we consider only \((\beta_{X^*,AP})\) hereafter. The vector \((\beta_{X^*,AP})\) is a function of parameters \( \alpha_X \) which are estimated using marginal maximum likelihood. We thus have that \( \sqrt{n}(\alpha_X - \alpha_X) \rightarrow N(0, \Sigma_{\alpha_X}) \) as \( n \rightarrow \infty \). Since \((\beta_{X^*,AP})\) is a differentiable function of the item parameters, the variance of \((\beta_{X^*,AP})\) can be derived using Cramer’s theorem, retrieving

\[
\sqrt{n} \left[ (\beta_{X^*,AP}) - (\beta_{X^*,AP}) \right] \rightarrow N \left\{ 0, \frac{\partial (\beta_{X^*,AP})}{\partial \alpha_X} \Sigma_{\alpha_X} \left[ \frac{\partial (\beta_{X^*,AP})}{\partial \alpha_X} \right] \right\},
\]

where \( \frac{\partial (\beta_{X^*,AP})}{\partial \alpha_X} \) is a \((k+1) \times 3k\) matrix of partial derivatives with \( 1 \times 3 \) vector entries \( \frac{\partial \beta_{X^*,x^*}}{\partial \alpha_1} \) and \( \frac{\partial \beta_{X^*,x^*}}{\partial \alpha_k} \), \( x^* \in \{0, 1, \ldots, k^*\}, l \in \{1, 2, \ldots, k^*\}, a \in \{0, 1, \ldots, k_A\}, l_A \in \{1, 2, \ldots, k_A\} \) of the same form as those in Ogasawara (2003).

Since Equation 3 defines the asymptotic distribution of the score probabilities the results can be directly applied in the kernel equating framework by the derivations provided in von Davier et al. (2004).

### 3. Implementation of IRT observed-score equating in kequate

The package kequate for R supports IRT observed-score equating for the EG and NEAT CE designs with the 2-PL or 3-PL IRT models. Asymptotic or bootstrap standard errors are calculated for each of the methods. The input used can either be matrices of observed item responses for each individual or objects containing IRT models which have been estimated using the R package ltm (Rizopoulos 2006).

To conduct an IRT observed-score equating in kequate, the function irtose() is used. The function irtose() has the following formal function call:

```r
irtose(design="CE", P, Q, x, y, a=0, qpoints, model="2pl", see="analytical", replications=50, kernel="gaussian", h=list(hx=0, hy=0, hxP=0, haP=0, hyQ=0, haQ=0), hlin=list(hxlin=0, hylin=0, hxPlin=0, haPlin=0, hyQlin=0, haQlin=0), KPEN=0, wpen=0.5, linear=FALSE, slog=1, bunif=1, altopt=FALSE)
```

Explanations of each of the arguments supplied to irtose() are given in Table 1.

If matrices of responses are provided as input to irtose(), the IRT models will be estimated using the R package ltm. The settings used in ltm will then be the default ones, except for the case of the 3-PL model where the nlminn optimizer is used instead of the default. Note that the 3-PL model has issues with convergence, hence it will not always be possible to get stable estimates of item parameters using this model. It is recommended to estimate the 3-PL
IRT Equating with `kequate`

<table>
<thead>
<tr>
<th>Argument(s)</th>
<th>Designs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>design</td>
<td>ALL</td>
<td>A character vector indicating which design to use. Possible designs are &quot;CE&quot; and &quot;EG&quot;.</td>
</tr>
<tr>
<td>P, Q</td>
<td>ALL</td>
<td>Matrices or objects created by the R package <code>ltm</code> containing either the responses for each question in groups P and Q or the estimated IRT models in groups P and Q.</td>
</tr>
<tr>
<td>x, y</td>
<td>ALL</td>
<td>Score value vectors for test X and test Y.</td>
</tr>
<tr>
<td>a</td>
<td>CE</td>
<td>Score value vector for the anchor test A.</td>
</tr>
<tr>
<td>qpoints</td>
<td>ALL</td>
<td>A numeric vector containing the quadrature points used in the equating. If not specified, the quadrature points from the IRT models will be used.</td>
</tr>
<tr>
<td>model</td>
<td>ALL</td>
<td>A character vector indicating which IRT model to use. Available models are 2PL and 3PL. Default is &quot;2PL&quot;.</td>
</tr>
<tr>
<td>see</td>
<td>ALL</td>
<td>A character vector indicating which standard errors of equating to use. Options are &quot;analytical&quot; and &quot;bootstrap&quot;, with default &quot;analytical&quot;.</td>
</tr>
<tr>
<td>replications</td>
<td>ALL</td>
<td>The number of bootstrap replications if using the bootstrap standard error calculations. Default is 50.</td>
</tr>
<tr>
<td>kernel</td>
<td>ALL</td>
<td>A character vector denoting which kernel to use, with options &quot;gaussian&quot;, &quot;logistic&quot;, &quot;stdgaussian&quot; and &quot;uniform&quot;. Default is &quot;gaussian&quot;.</td>
</tr>
<tr>
<td>h</td>
<td>ALL</td>
<td>Optional argument to specify the continuization parameters manually as a list with suitable bandwidth parameters. In an EG design design: hx and hy, in a NEAT CE design: hxP, haP, hyQ and haQ. (If <code>linear</code>=TRUE, then these arguments have no effect.)</td>
</tr>
<tr>
<td>hlin</td>
<td>ALL</td>
<td>Optional argument to specify the linear continuization parameters manually as a list with suitable bandwidth parameters. In an EG design: hxlin and hylin, in a NEAT CE design: hxPlin, haPlin, hyQlin and haQlin.</td>
</tr>
<tr>
<td>slog</td>
<td>ALL</td>
<td>The parameter used in the logistic kernel. Default is 1.</td>
</tr>
<tr>
<td>bunif</td>
<td>ALL</td>
<td>The parameter used in the uniform kernel. Default is 0.5.</td>
</tr>
<tr>
<td>KPEN</td>
<td>ALL</td>
<td>Optional argument to specify the constant used in deciding the optimal continuization parameter. Default is 0.</td>
</tr>
<tr>
<td>wpen</td>
<td>ALL</td>
<td>An argument denoting at which point the derivatives in the second part of the penalty function should be evaluated. Default is 1/4.</td>
</tr>
<tr>
<td>linear</td>
<td>ALL</td>
<td>Logical denoting if a linear equating only is to be performed. Default is FALSE.</td>
</tr>
<tr>
<td>altopt</td>
<td>ALL</td>
<td>Logical which sets the bandwidth parameter equal to a variant of Silverman’s rule of thumb. Default is FALSE.</td>
</tr>
</tbody>
</table>

Table 1: Arguments supplied to `irtose()`.
models separately using the package \texttt{ltm}. Currently, \texttt{kequate} only provides support for IRT models without particular restrictions on the parameters.

4. Examples

For these examples, data was simulated using \texttt{R} in accordance with the 2-PL and 3-PL IRT models. The simulated data for both the 2-PL model and the 3-PL model have the same ability level for each individual and the same discrimination and difficulty parameters for each item. The simulation procedure is identical to that for the 2-PL and 3-PL IRT models described in Ogasawara (2003). The \texttt{R} code which generated the data is given below.

\begin{verbatim}
R> library(kequate)
R> set.seed(7)
R> akX <- runif(15, 0.5, 2)
R> bkX <- rnorm(15)
R> ckX <- runif(15, 0.1, 0.2)
R> akY <- runif(15, 0.5, 2)
R> bkY <- rnorm(15)
R> ckY <- runif(15, 0.1, 0.2)
R> akA <- runif(15, 0.5, 2)
R> bkA <- rnorm(15)
R> ckA <- runif(15, 0.1, 0.2)
R> dataP <- matrix(0, nrow=1000, ncol=30)
R> dataQ <- matrix(0, nrow=1000, ncol=30)
R> data3plP <- matrix(0, nrow=1000, ncol=30)
R> data3plQ <- matrix(0, nrow=1000, ncol=30)
R> for(i in 1:1000){
+ ability <- rnorm(1)
+ dataP[i,1:15] <- (1/(1+exp(-akX*(ability-bkX)))) > runif(15)
+ dataP[i,16:30] <- (1/(1+exp(-akA*(ability-bkA)))) > runif(15)
+ data3plP[i,1:15] <- (ckX+(1-ckX)/(1+exp(-akX*(ability-bkX)))) > runif(15)
+ data3plP[i,16:30] <- (ckA+(1-ckA)/(1+exp(-akA*(ability-bkA)))) > runif(15)
+ }
R> for(i in 1:1000){
+ ability <- rnorm(1, mean=0.5)
+ dataQ[i,1:15] <- (1/(1+exp(-akY*(ability-bkY)))) > runif(15)
+ dataQ[i,16:30] <- (1/(1+exp(-akA*(ability-bkA)))) > runif(15)
+ data3plQ[i,1:15] <- (ckY+(1-ckY)/(1+exp(-akY*(ability-bkY)))) > runif(15)
+ data3plQ[i,16:30] <- (ckA+(1-ckA)/(1+exp(-akA*(ability-bkA)))) > runif(15)
+ }
\end{verbatim}

4.1. IRT observed-score kernel equating with the 2-PL model

For the 2-PL model data was simulated in a non-equivalent groups with anchor test design for two populations of size 1000 with differing ability levels. The main tests had 15 items each and the anchor test had 15 items. The simulated data were stored in matrices \texttt{dataP} for
group P and dataQ for group Q. To equate the two main tests using chain equating, we then call the function irtose() as follows:

```R
R> eq2pl <- irtose("CE", dataP, dataQ, 0:15, 0:15, 0:15)
```

To display a summary of the equating we write:

```R
R> summary(eq2pl)
```

Design: IRT-OSE CE

Kernel: gaussian

Sample Sizes:
  Test X: 1000
  Test Y: 1000

Score Ranges:
  Test X:
    Min = 0 Max = 15
  Test Y:
    Min = 0 Max = 15
  Test A:
    Min = 0 Max = 15

Bandwidths Used:
  hxP  hyQ  haP  haQ  hxPlin  hyQlin  haPlin
1 0.5656154 0.5559258 0.5170625 0.536149 2999.144 3296.431 3449.946
  haQlin
1 3741.527

Equating Function and Standard Errors:

<table>
<thead>
<tr>
<th>Score</th>
<th>eqYx</th>
<th>SEYx</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.4095018</td>
<td>0.1124074</td>
</tr>
<tr>
<td>2</td>
<td>0.3507320</td>
<td>0.1494146</td>
</tr>
<tr>
<td>3</td>
<td>1.1144228</td>
<td>0.1660853</td>
</tr>
<tr>
<td>4</td>
<td>1.9176735</td>
<td>0.1821037</td>
</tr>
<tr>
<td>5</td>
<td>2.7810503</td>
<td>0.1884568</td>
</tr>
<tr>
<td>6</td>
<td>3.6860775</td>
<td>0.1834772</td>
</tr>
<tr>
<td>7</td>
<td>4.6403829</td>
<td>0.1727179</td>
</tr>
<tr>
<td>8</td>
<td>5.6334253</td>
<td>0.1576295</td>
</tr>
<tr>
<td>9</td>
<td>6.6716804</td>
<td>0.1409382</td>
</tr>
<tr>
<td>10</td>
<td>7.7590188</td>
<td>0.1270293</td>
</tr>
<tr>
<td>11</td>
<td>8.9034599</td>
<td>0.1203290</td>
</tr>
<tr>
<td>12</td>
<td>10.1080795</td>
<td>0.1233701</td>
</tr>
<tr>
<td>13</td>
<td>11.3655142</td>
<td>0.1351404</td>
</tr>
<tr>
<td>14</td>
<td>12.6416920</td>
<td>0.1445380</td>
</tr>
</tbody>
</table>
Comparing the Moments:

<table>
<thead>
<tr>
<th>PREAx</th>
<th>PREYa</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.04139275</td>
<td>0.02388016</td>
</tr>
<tr>
<td>-0.11944000</td>
<td>-0.060078687</td>
</tr>
<tr>
<td>-0.88808023</td>
<td>-0.009477534</td>
</tr>
<tr>
<td>-1.93665049</td>
<td>0.140270330</td>
</tr>
<tr>
<td>-3.18244188</td>
<td>0.369599428</td>
</tr>
<tr>
<td>-4.56972851</td>
<td>0.669296557</td>
</tr>
<tr>
<td>-6.06482013</td>
<td>1.035297932</td>
</tr>
<tr>
<td>-7.64492710</td>
<td>1.465999964</td>
</tr>
<tr>
<td>-9.29376506</td>
<td>1.960971994</td>
</tr>
<tr>
<td>-10.99914952</td>
<td>2.520359880</td>
</tr>
</tbody>
</table>

The equating shows that the tests are similar in difficulty but that test Y is slightly more difficult than test X.

When supplying matrices of responses to each item as input to `irtose()`, the IRT models are estimated using the package `ltm`. An equating is then conducted using the estimated IRT models. The objects created by `ltm` are stored in the output from `irtose()`. To access the objects we write:

```r
R> irtobjects <- eq2pl@irt
```

This will create a list of the objects created by `ltm` and the adjusted asymptotic covariance matrices of the item parameters. We save the objects from `ltm` for future usage:

```r
R> sim2plP <- irtobjects$ltmP
R> sim2plQ <- irtobjects$ltmQ
```

### 4.2. IRT observed-score kernel equating with the 3-PL model

For the 3-PL model data was again simulated in a non-equivalent groups with anchor test design for two populations of size 1000 with differing ability levels. As before, the main tests had 15 items each and the anchor test had 15 items. In this example, the IRT models were estimated using the function `tpm()` in the package `ltm`, creating the objects `sim3plP` and `sim3plQ containing the IRT models. For details of IRT model estimation using `ltm`, see Rizopoulos (2006). The resulting objects are then given as input to the function `irtose()` to conduct an equating:

```r
R> eq3pl <- irtose("CE", sim3plP, sim3plQ, 0:15, 0:15, 0:15, model="3pl")
R> summary(eq3pl)
```

**Design:** IRT-OSE CE

**Kernel:** gaussian
Sample Sizes:
  Test X: 1000
  Test Y: 1000

Score Ranges:
  Test X:
    Min = 0 Max = 15
  Test Y:
    Min = 0 Max = 15
  Test A:
    Min = 0 Max = 15

Bandwidths Used:
  hxP  hyQ  haP  haQ  hxPlin  hyQlin  haPlin
  1  0.5554813  0.5406952  0.5472448  0.5543  2760.746  2863.484  3074.537
  haQlin
  1  3406.924

Equating Function and Standard Errors:

<table>
<thead>
<tr>
<th>Score</th>
<th>eqYx</th>
<th>SEEYx</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3329018</td>
<td>0.2755159</td>
</tr>
<tr>
<td>2</td>
<td>1.2860273</td>
<td>0.3022574</td>
</tr>
<tr>
<td>3</td>
<td>2.1723155</td>
<td>0.2972621</td>
</tr>
<tr>
<td>4</td>
<td>3.0109638</td>
<td>0.2765362</td>
</tr>
<tr>
<td>5</td>
<td>3.8245077</td>
<td>0.2533002</td>
</tr>
<tr>
<td>6</td>
<td>4.6244139</td>
<td>0.2306618</td>
</tr>
<tr>
<td>7</td>
<td>5.4286157</td>
<td>0.2057037</td>
</tr>
<tr>
<td>8</td>
<td>6.2509278</td>
<td>0.1800856</td>
</tr>
<tr>
<td>9</td>
<td>7.1052576</td>
<td>0.1569177</td>
</tr>
<tr>
<td>10</td>
<td>8.0046533</td>
<td>0.1395113</td>
</tr>
<tr>
<td>11</td>
<td>8.9649110</td>
<td>0.1288226</td>
</tr>
<tr>
<td>12</td>
<td>10.0060465</td>
<td>0.1241381</td>
</tr>
<tr>
<td>13</td>
<td>11.1412949</td>
<td>0.1247086</td>
</tr>
<tr>
<td>14</td>
<td>12.3531419</td>
<td>0.1334708</td>
</tr>
<tr>
<td>15</td>
<td>13.5821981</td>
<td>0.1449046</td>
</tr>
<tr>
<td>16</td>
<td>14.7470747</td>
<td>0.1318868</td>
</tr>
</tbody>
</table>

Comparing the Moments:

<table>
<thead>
<tr>
<th>PREAx</th>
<th>PREYy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.007419065</td>
</tr>
<tr>
<td>2</td>
<td>-0.135681072</td>
</tr>
<tr>
<td>3</td>
<td>-0.691806423</td>
</tr>
<tr>
<td>4</td>
<td>-1.603890067</td>
</tr>
<tr>
<td>5</td>
<td>-2.794730890</td>
</tr>
<tr>
<td>6</td>
<td>-4.199379276</td>
</tr>
<tr>
<td>7</td>
<td>-5.770833795</td>
</tr>
</tbody>
</table>
We plot the results with the method for the function `plot()` for the class `keout` created by `irttose()`.

```r
R> plot(eq3pl)
```

The plot is seen in Figure 1.

![Equated values and standard errors of equating for the IRT observed-score equating using the 3-PL model.](image)

Figure 1: The equated values and standard errors of equating for the IRT observed-score equating using the 3-PL model.

### 4.3. IRT observed-score local equating

IRT observed-score equating can be utilized when conducting what is called local equating, where different equating functions are calculated based on the ability level or a proxy of the ability level of the individuals taking the tests to be equated. Local equating using IRT
observed-score equating is conducted by fixing the ability level to a particular single value or a sequence of values and then only considering this value or sequence of values when calculating the score probabilities. These score probabilities are then used for the equating just as in a regular IRT observed-score equating.

In kequate, local equating using IRT observed-score equating can be conducted by adjusting the optional argument \texttt{qpoints} in the \texttt{irtose()} function call. For example, by specifying \texttt{qpoints=1} a local equating for the individuals with the ability level equal to 1 is conducted. The argument \texttt{qpoints} can be set to a numeric vector of any length.

As an example, we conduct a local equating for individuals with ability level equal to -1, 0 and 1, respectively, using the simulated 2-PL data previously described. We then call \texttt{irtose()} as follows:

\begin{verbatim}
R> eq2plLOW <- irtose("CE", sim2plP, sim2plQ, 0:15, 0:15, 0:15, qpoints=-1)
R> eq2plAVG <- irtose("CE", sim2plP, sim2plQ, 0:15, 0:15, 0:15, qpoints=0)
R> eq2plHIGH <- irtose("CE", sim2plP, sim2plQ, 0:15, 0:15, 0:15, qpoints=1)
\end{verbatim}

![Equated value vs. score value for different ability levels](image.png)

Figure 2: The equated values for each score value for three different ability levels in a local equating in the NEAT CE design.
The results of these equatings are displayed in Figure 2, showing that the equating function is somewhat different for the three different ability levels.

5. Future developments

In the present implementation, only the 2-PL and 3-PL IRT models without parameter restrictions are supported in kequate. Future work will include support for the additional IRT models available in ltm such as the Rasch model and the 1-PL model and the ability to use the features of parameter restrictions available in ltm when conducting IRT observed-score equating. Additionally, the NEAT design using post-stratification equating (PSE) with support for various ways of estimating the equating coefficients is planned to be included in the package.

References


**Affiliation:**

Björn Andersson  
Department of Statistics  
Uppsala University, Box 513  
SE-751 20 Uppsala, Sweden  
E-mail: bjorn.andersson@statistik.uu.se  
URL: [http://katalog.uu.se/empInfo?id=N11-1505](http://katalog.uu.se/empInfo?id=N11-1505)

Marie Wiberg  
Department of Statistics, USBE  
Umeå University  
SE-901 87 Umeå, Sweden  
E-mail: marie.wiberg@stat.umu.se  
URL: [http://www.usbe.umu.se/om-handelshogskolan/personal/maewig95](http://www.usbe.umu.se/om-handelshogskolan/personal/maewig95)