# equate: An R Package for Statistical Equating

**Anthony D. Albano**

University of Nebraska-Lincoln

### Abstract

The R package **equate** (Albano 2013) contains functions for statistical equating under equivalent and nonequivalent groups designs. This paper introduces these designs and provides an overview of equating with details about each of the supported methods. Examples demonstrate the basic functionality of the package.

*Keywords*: equating, linking, R.

## 1. Introduction

Equating is a statistical procedure commonly used in testing programs where administrations across more than one occasion and more than one examinee group can lead to overexposure of items, threatening the security of the test. Another somewhat less common use is in progress monitoring and growth modeling, where administrations occur across multiple time points for the same individuals, and using the same test form may lead to practice effects. In each of these contexts item exposure can be limited by using alternate test forms; however, multiple forms require multiple score scales for a single test. Despite being designed based on the same specifications, to cover the same content at the same level of difficulty, these scales cannot be considered identical. Instead, they are assumed to differ in difficulty.

Difficulty differences across forms complicate the comparison of ability estimates, whether across individuals taking different forms, or across time points for an individual. In each case, ability differences are confounded by differences in form difficulty. The goal of equating is to adjust for these differences in difficulty across alternate forms of a test, so as to produce comparable score scales and thereby improve the estimation of ability.

The **equate** package focuses on statistical or observed-score equating, as opposed to item response theory (IRT) equating (for IRT equating in R, see Weeks 2010). Most of the **equate** procedures fit under what is called *traditional equating*, or non-IRT equating. IRT equating is based on a measurement model, the IRT model, whereas most traditional methods are not. For this reason, the equating described here is referred to as statistical equating. Although there are many benefits of equating forms using IRT, statistical equating can often be a simpler and more practical alternative, one involving fewer and less stringent assumptions (for further discussion, see Kolen and Brennan 2004; Livingston 2004).

Equating defines a functional relationship between multiple test score distributions and thereby between multiple score scales. When the test forms have been created according to the same specifications and are similar in statistical characteristics, this functional relationship is referred to as an *equating function* which serves to translate scores from one scale directly to

their equivalent values on another. Whether score distributions are based on samples from a single examinee population or different examinee populations, if the appropriate assumptions are met the equating function can then be generalized to other examinees (for more details, see Holland and Dorans 2006). The generalizability of an equating function depends, in part, on the equating design.

# 2. Equating Designs

An *equating design* refers to the basic structure of an equating study, just as a research design refers to the structure of a research study. The equating study organizes all the stages which are essential to and which lead up to the equating process. These stages include creation of test forms, sampling of examinees, and administration of the test. The equating design specifies the test administration procedures, i.e., data collection, and just as the control of variables in a research study depends on the design, control of examinee ability in an equating study depends on the equating design.

An equating study can take place in a variety of situations, depending on the needs and resources of a testing program. As a result, numerous equating designs have been documented in the literature. For simplicity, in this vignette and in the **equate** package, equating designs are categorized as either involving *equivalent groups* or *nonequivalent groups*.

## 2.1. Equivalent Groups

The equivalent groups design consists of either a single group of examinees taking both forms of a test, or two groups sampled randomly from a single population and considered to be randomly equivalent. In either case it is assumed that the two groups are equivalent in ability, and that any differences in scores across forms can be attributed entirely to form difficulty. When forms are administered to a single group, administration procedures can be complicated by order and fatigue effects. Thus, the single group design is often not a practical option. However, because it involves only the examinee population of interest, referred to as the target population, the equivalent groups design is the most efficient, as examinee ability is controlled directly.

## 2.2. Nonequivalent Groups

Without equivalent examinee groups two related problems arise: the target population must be defined indirectly using samples from two different examinee populations, and the ability of these groups must then be controlled. In the nonequivalent groups design[1] these issues are both addressed through the use of what is referred to as an *anchor test*, a subset of items appearing on both test forms. All non-equivalence in ability is assumed to be removed via these anchor items or common items. Though this design is often more practical, as nonequivalent groups are more easily obtained than equivalent ones, it also involves additional assumptions, as discussed in the next section (for details see Holland and Dorans 2006).

As noted above, the equivalent groups is the simpler equating design. The traditional equating types applied with this design are the mean, linear, and equipercentile. More complex

---

[1]The nonequivalent groups design is also referred to as the nonequivalent groups with anchor test design, the common-item nonequivalent groups design, or simply the anchor- or common-item design.

extensions of these have been developed for use with the nonequivalent groups design, each of which handles the issues inherent with nonequivalent groups in a slightly different way. These methods are described briefly below, followed by examples of their implementation in the **equate** package.

# 3. Equating Types and Methods

## 3.1. Types of Equating

Equating procedures used with the equivalent groups design are referred to here and in the **equate** package as equating *types*. These types are categorized as either linear, including mean and linear equating, or nonlinear, including equipercentile and circle-arc equating. The functional definition of each equating type is presented below.

*Identity equating*

The identity equating function simply reproduces the original score value unchanged, and thus un-equated:

$$id_Y(x_i) = x_i. \tag{1}$$

Here, $id$ abbreviates the identity function, where the $Y$ subscript indicates that the function produces equated scores on form $Y$ and $(x_i)$ refers to a raw score on form $X$. The identity conversion of $x_i$ to form $Y$ is simply $x_i$.

With small samples, and especially when test forms are believed to be parallel, identity equating, or no equating, has been recommended over other types (Kolen and Brennan 2004). The identity function can also be combined with any of the functions described below to obtain the synthetic equating function (Kim, von Davier, and Haberman 2008):

$$s_Y(x_i) = (1 - w_{id})g_Y(x_i) + w_{id}id_Y(x_i), \tag{2}$$

where $s_Y(x_i)$ is a weighted combination of any generic equating function $g_Y(x_i)$ with the identity, and $w_{id}$ is a value between zero and one.

*Linear equating*

Linear equating defines a linear relationship between scores from forms $X$ and $Y$, based on the mean and standard deviation of each. In other words, the standardized deviation scores, or $z$-scores, are set equal for all score points $i$:

$$\frac{x_i - \hat{\mu}(X)}{\hat{\sigma}(X)} = \frac{y_i - \hat{\mu}(Y)}{\hat{\sigma}(Y)}. \tag{3}$$

When solved for $y_i$, the linear function $l_Y(x_i)$ can be rewritten in slope-intercept form as

$$l_Y(x_i) = \frac{\hat{\sigma}(Y)}{\hat{\sigma}(X)}x_i - \frac{\hat{\sigma}(Y)}{\hat{\sigma}(X)}\hat{\mu}(X) + \hat{\mu}(Y). \tag{4}$$

*Mean equating*

Mean equating is a simplification of linear where the slope, or ratio of standard deviations, is not estimated but is instead assumed to be 1. Deviation scores across forms are set equal:

$$x_i - \hat{\mu}(X) = y_i - \hat{\mu}(Y), \tag{5}$$

and the resulting mean function $m_Y(x_i)$ for equating $X$ to $Y$ is

$$m_Y(x_i) = x_i - \hat{\mu}(X) + \hat{\mu}(Y). \tag{6}$$

*Equipercentile equating*

Equipercentile equating defines a nonlinear relationship between score scales by setting equal the percentile ranks for each score point. Specifically, the equipercentile equivalent of a form-$X$ score on the $Y$ scale is calculated by finding the percentile rank in $X$ of score $i$, and then the form-$Y$ score associated with that form-$Y$ percentile rank:

$$e_Y(x_i) = Q^{-1}[P(x_i)]. \tag{7}$$

Here, $P(x)$ is the percentile rank function in $X$ and $Q^{-1}(x)$ is the inverse percentile rank function in $Y$. The process is complicated by the fact that scores are typically discrete, and must be made continuous (for a detailed description see Kolen and Brennan 2004, ch. 2).

Because it involves estimation at each score point, equipercentile equating is especially susceptible to sampling error. Smoothing methods are typically used to reduce irregularities due to sampling error in either the score distributions or the equating function itself. Two commonly used smoothing methods include polynomial loglinear presmoothing (Holland and Thayer 2000) and cubic-spline postsmoothing (Kolen 1984). The **equate** package supports loglinear presmoothing via the `glm` function (see Appendix A.2 for details).

*Circle-arc equating*

Circle-arc equating also defines a nonlinear relationship between score scales, but it utilizes only three score points in $X$ and $Y$: the lowest meaningful score $(x_1, y_1)$, which for a multiple-choice test could be the lowest score expected by chance; a midpoint based on the center (e.g., means) of each form $(x_2, y_2)$; and the maximum possible score on each form $(x_3, y_3)$. Only the midpoint requires estimation. The low and high points define the linear component of the function:

$$lin_Y(x_i) = y_1 + \frac{y_3 - y_1}{x_3 - x_1}(x_i - x_1). \tag{8}$$

This linear function is combined with a curvilinear one, a circle-arc that is based on $y_{2*}$, the distance in $Y$ units of the point $(x_2, y_2)$ from the line $lin_Y(x)$. The center $(x_c, y_c)$ and radius $r$ of the circle define the curvilinear component:

$$arc_Y(x_i) = y_c \pm \sqrt{r^2 - (x_i - x_c)^2}, \tag{9}$$

where the second quantity, under the square root, is added to $y_c$ if $y_{2*}$ is positive (i.e., above the linear function) and subtracted if it is negative (i.e., below the linear function). The circle-arc function $c_Y(x_i)$ combines the linear and curvilinear components:

$$c_Y(x_i) = lin_y(x_i) + arc_y(x_i). \tag{10}$$

Equations for the center points and radius of the circle are included in Appendix A.3. Livingston and Kim (2009) provide a complete description of the circle-arc equating process.

## 3.2. Equating Methods

The equating functions presented above are defined in terms of a single examinee population, and are assumed to generalize to this examinee population. In the nonequivalent groups design, scores come from two distinct populations, referred to here as populations 1 and 2. As a result, the equating functions are redefined in terms of a hypothetical population of examinees. This population is referred to as the *synthetic population* (Braun and Holland 1982). As described by Kolen and Brennan (2004), the linear equating function from equation (4) can be rewritten in terms of the synthetic population as follows:

$$l_{Y_S}(x_i) = \frac{\hat{\sigma}_S(Y)}{\hat{\sigma}_S(X)} x_i - \frac{\hat{\sigma}_S(Y)}{\hat{\sigma}_S(X)} \hat{\mu}_S(X) + \hat{\mu}_S(Y). \tag{11}$$

Since population $S$ did not take forms $X$ or $Y$, all of the terms $\hat{\mu}_S$ and $\hat{\sigma}_S$ in this equation must be estimated indirectly using: for the means,

$$\hat{\mu}_S(X) = \hat{\mu}_1(X) - w_2 \gamma_1 [\hat{\mu}_1(V) - \hat{\mu}_2(V)], \tag{12}$$
$$\hat{\mu}_S(Y) = \hat{\mu}_2(Y) + w_1 \gamma_2 [\hat{\mu}_1(V) - \hat{\mu}_2(V)]; \tag{13}$$

and for the variances,

$$\hat{\sigma}_S^2(X) = \hat{\sigma}_1^2(X) - w_2 \gamma_1^2 [\hat{\sigma}_1^2(V) - \hat{\sigma}_2^2(V)] + w_1 w_2 \gamma_1^2 [\hat{\mu}_1(V) - \hat{\mu}_2(V)]^2, \tag{14}$$
$$\hat{\sigma}_S^2(Y) = \hat{\sigma}_2^2(Y) + w_1 \gamma_2^2 [\hat{\sigma}_1^2(V) - \hat{\sigma}_2^2(V)] + w_1 w_2 \gamma_2^2 [\hat{\mu}_1(V) - \hat{\mu}_2(V)]^2. \tag{15}$$

In these equations the weights $w_1$ and $w_2$ sum to 1, and are used to specify the desired influence of populations 1 and 2 in the estimation. They are typically chosen to be proportional to sample size. The $\gamma$ terms represent the relationship between total scores on $X$ and $Y$ and the respective anchor test scores on $V$ (described further below). As is clear, $\gamma_1$ and $\gamma_2$ are used along with the weights to adjust the $\hat{\mu}$ and $\hat{\sigma}^2$ terms for $X$ and $Y$ in order to obtain corresponding estimates for the synthetic population. For example, setting $w_1 = 0$ and $w_2 = 1$ will force $\hat{\mu}_S(Y)$ to equal $\hat{\mu}_2(Y)$, and conversely $\hat{\mu}_2(X)$ will be adjusted the maximum amount to obtain $\hat{\mu}_S(X)$. The same would occur with the estimation of synthetic variances. Furthermore, the adjustments would be completely removed if $\hat{\mu}_1(V) = \hat{\mu}_2(V)$ and $\hat{\sigma}_1^2(V) = \hat{\sigma}_2^2(V)$.

A variety of techniques have been developed for estimating the $\gamma$ terms required by equations (12)-(15). These techniques are referred to here as equating *methods*. The **equate** package currently supports the Tucker, nominal weights, Levine observed score, Levine true score, Braun/Holland, frequency estimation, and chained equating methods. Table 1 shows the supported methods that apply to each equating type.

### *Tucker equating*

In Tucker equating the relationship between total and anchor test scores is defined in terms of regression slopes, where $\gamma_1$ is the slope resulting from the regression of $X$ on $V$ for population 1, and $\gamma_2$ the slope from a regression of $Y$ on $V$ for population 2:

$$\gamma_1 = \frac{\hat{\sigma}_1(X, V)}{\hat{\sigma}_1^2(V)} \qquad \text{and} \qquad \gamma_2 = \frac{\hat{\sigma}_2(Y, V)}{\hat{\sigma}_2^2(V)}. \tag{16}$$

The Tucker method assumes that across populations 1 and 2: 1) the coefficients resulting form a regression of $X$ on $V$ are the same, and 2) the conditional variance of $X$ given $V$ is the same. These assumptions apply to the regression of $Y$ on $V$ and the covariance of $Y$ given $V$ as well.

Table 1: Applicable Equating Types and Methods

|               | nominal    | tucker     | levine     | braun      | frequency  | chained    |
|---------------|------------|------------|------------|------------|------------|------------|
| mean          | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ |            | $\checkmark$ |
| linear        |            | $\checkmark$ | $\checkmark$ | $\checkmark$ |            | $\checkmark$ |
| equipercentile |            |            |            |            | $\checkmark$ | $\checkmark$ |
| circle-arc    | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ |            | $\checkmark$ |

*Nominal weights equating*

Nominal weights equating is a simplified version of the Tucker method where the total and anchor tests are assumed to have similar statistical properties and to correlate perfectly within populations 1 and 2. In this case the $\gamma$ terms can be approximated by the ratios

$$\gamma_1 = \frac{K(X)}{K(V)} \qquad \text{and} \qquad \gamma_2 = \frac{K(Y)}{K(V)}, \tag{17}$$

where $K$ is the number of items on the test. See Babcock, Albano, and Raymond (2012) for a description and examples.

*Levine equating*

Assumptions for the Levine observed score method are stated in terms of true scores (though only observed scores are used), where, across both populations: 1) the correlation between true scores on $X$ and $V$ is 1, as is the correlation between true scores on $Y$ and $V$; 2) the coefficients resulting form a regression of true scores for $X$ on $V$ are the same, as with true scores for $Y$ on $V$; and 3) measurement error variance is the same (across populations) for $X$, $Y$, and $V$. These assumptions make possible the estimation of $\gamma$ as

$$\gamma_1 = \frac{\hat{\sigma}_1^2(X)}{\hat{\sigma}_1(X,V)} \qquad \text{and} \qquad \gamma_2 = \frac{\hat{\sigma}_2^2(Y)}{\hat{\sigma}_2(Y,V)}, \tag{18}$$

which are the inverses of the respective regression slopes for $V$ on $X$ and $V$ on $Y$. The Levine true score method is based on the same assumptions as the observed score method; however, it uses a slightly different linear equating function:

$$l_Y(x_i) = \frac{\gamma_2}{\gamma_1}(X)[x_i - \hat{\mu}_1(X)] + \hat{\mu}_2(Y) + \gamma_2[\hat{\mu}_1(V) - \hat{\mu}_2(V)]. \tag{19}$$

Hanson (1991) and Kolen and Brennan (2004) provide justifications for using this approach.

*Frequency estimation equating*

The frequency estimation method is used in equipercentile equating under the nonequivalent groups design. It is similar to the methods described above in that it involves a synthetic

population. However, in this case score distributions (i.e., percentile ranks) for the synthetic population taking forms $X$ and $Y$ are required:

$$e_{Y_S}(x_i) = Q_S^{-1}[P_S(x_i)]. \qquad (20)$$

When the assumption is made that the conditional distribution of total scores on $X$ for a given score point in $V$ is the same across populations 1 and 2 (as with $Y$ and $V$) the synthetic distributions can be obtained:

$$f_S(x_i) = w_1 f_1(x_i) + w_2 \sum f_1(x|v) h_2(v), \qquad (21)$$

$$g_S(y_i) = w_2 g_2(y_i) + w_1 \sum g_2(y|v) h_1(v) \qquad (22)$$

Here, $f$, $g$, and $h$ denote the distribution functions for forms $X$, $Y$, and $V$ respectively. As before, $w_1$ and $w_2$ specify the amount of adjustment to be made to each observed distribution in the estimation of the corresponding synthetic distribution.

### Braun/Holland equating

As a kind of extension of the frequency estimation method, the Braun/Holland method defines a linear function relating $X$ and $Y$ that is based on the estimates $\hat{\mu}_S(X)$, $\hat{\mu}_S(Y)$, $\hat{\sigma}_S(X)$, and $\hat{\sigma}_S(Y)$ for the synthetic distributions $f_S(x)$ and $g_S(y)$ obtained via frequency estimation. Thus the full synthetic distributions are estimated, as with frequency estimation, but only in order to obtain the means and standard deviations of each. Though not often used in practice, the method provides an interesting combination of the linear and nonlinear procedures (Braun and Holland 1982).

### Chained equating

Finally, chained equating (Livingston, Dorans, and Wright 1990) can be applied to both linear and equipercentile equating under the nonequivalent groups with anchor test design. It differs from all other methods discussed here in that it does not reference a synthetic population. Instead, it introduces an additional equating function in the process of estimating score equivalents (see Appendix A.1 for details). For both linear and equipercentile equating the steps are as follows:

1. Define the function relating $X$ to $V$ for population 1, $l_{V1}(x)$ or $e_{V1}(x)$

2. Define the function relating $V$ to $Y$ for population 2, $l_{Y2}(v)$ or $e_{Y2}(v)$

3. Equate $X$ (population 1) to the scale of $Y$ using both equating functions, where

$$lchain_Y(x) = l_{Y2}[l_{V1}(x)] \qquad \text{and} \qquad echain_Y(x) = e_{Y2}[e_{V1}(x)]$$

### Methods for circle-arc equating

As discussed above, the circle-arc equating function combines a linear with a curvilinear component based on three points in the $X$ and $Y$ score distributions. The first and third of these points are determined by the score scale, whereas the midpoint must be estimated. Thus, equating methods used with circle-arc equating apply only to estimation of this midpoint.

Livingston and Kim (2009) demonstrate chained linear equating of means, under a nonequivalent groups design. The midpoint could also be estimated using other linear methods, such as Tucker or Levine.

Note that circle-arc equating is defined here as an equating *type*, and equating *methods* are used to estimate the midpoint, which implies a nonequivalent groups design. When groups are considered equivalent (i.e., an anchor test is not used) equating at the midpoint is simply mean equating, as mentioned above (replace $x_i$ with $\hat{\mu}(X)$ in equation 4 to see why this is the case). With scores on an anchor test, both Tucker and Levine equating at the midpoint also reduce to mean equating. However, chained linear equating at the midpoint differs from chained mean (see Appendix A.1).

# 4. Application Using the equate Package

## 4.1. Sample Test Scores

The examples below rely on three data sets, all of which are provided in the **equate** package. The first, `ACTmath`, is used throughout Kolen and Brennan (2004), and comes from two administrations of the ACT mathematics test. The test scores are based on a random groups design and are contained in a three-column matrix where column one is the 40-point score scale and columns two and three the number of examinees for forms `x` and `y` obtaining each score point.

```
> library(equate)

[1] "equate"    "stats"     "graphics"  "grDevices" "utils"     "datasets"
[7] "methods"   "base"

> head(ACTmath)

     scale xcount ycount
[1,]     0      0      0
[2,]     1      1      1
[3,]     2      1      3
[4,]     3      3     13
[5,]     4      9     42
[6,]     5     18     59
```

The second data set, `KBneat`, is also referenced in Kolen and Brennan (2004). It contains scores for two forms of a 36-item test administered under a nonequivalent groups with anchor test design. The 12-item anchor test is internal, that is, the total-test score for an examinee includes the score on the anchor items. Thus, the number of non-anchor items, items unique to each form, is 24, and the highest possible score is 36. Unlike the first data set, `KBneat` contains a separate total-test and anchor-test score for each examinee, as is required by the nonequivalent groups equating methods described above. It is a list of length two where the list elements `x` and `y` each consist of a two-column matrix of scores on the total test, and scores on the anchor test `v`.

```
> head(KBneat$x)


      x xv
[1,]  8  3
[2,] 21  6
[3,] 31 10
[4,]  7  2
[5,] 18  5
[6,] 36 12
```

The third data set, `PISA`, contains scored cognitive item response data from the 2009 administration of the Programme for International Assessment (PISA). Four `data.frame`s are included in `PISA`: `PISA$students` contains scores on the cognitive assessment items in math, reading, and science for all 5233 students in the USA cohort; `PISA$booklets` contains information about the structure of the test design, where multiple item sets, or clusters, were administered across 13 test booklets; and `PISA$items` contains the cluster, subject, and maximum possible score for each item; and `PISA$totals` contains a list of cluster total scores for each booklet, calculated using `PISA$students` and `PISA$booklets`. For additional details, see `?PISA`.

```
> attach(PISA)
> head(booklets)


  bookid clusterid   itemid order
1      1        m1  m033q01     1
2      1        m1 m034q01t     1
3      1        m1  m155q01     1
4      1        m1 m155q02d     1
5      1        m1 m155q03d     1
6      1        m1 m155q04t     1


> head(items)


    itemid clusterid max subject
1  m033q01        m1   1       m
2 m034q01t        m1   1       m
3  m155q01        m1   1       m
4 m155q02d        m1   2       m
5 m155q03d        m1   2       m
6 m155q04t        m1   1       m


> head(totals$b1)


   m1 r1 r3a m3
4  11 10  10  2
10  7  5   6  7
```

```
37  7  4   2 5
38  6  4   4 3
65  9 10   9 5
68  3  8   6 2
```

## 4.2. Preparing the Score Distributions

The **equate** package handles score distributions primarily as frequency tables, as described by the `freqtab` function, which is used to create them. The `ACTmath` data set is an example of a frequency table; scores for over 8,000 examinees ($N_X = 4,329$, $N_Y = 4,152$) are stored compactly in three columns and 41 rows. The trade-off is that there is no record of scores at the individual level, but this information is not required under the random groups design, as is evident in equations (3)-(10). Frequency tables of class `"freqtab"` are created for the 2 `ACTmath` forms as follows:

```
> act.x <- as.freqtab(cbind(ACTmath[, 1], ACTmath[, 2]))
> act.y <- as.freqtab(cbind(ACTmath[, 1], ACTmath[, 3]))
> act.x[1:4,]


     x count
[1,] 0     0
[2,] 1     1
[3,] 2     1
[4,] 3     3
```

Here, the command `as.freqtab` is used because the vectors for the score scale and counts are already tabulated, thus they are simply combined and the class changed. The tables can be summarized with the `descript` function:

```
> rbind(descript(act.x), descript(act.y))


         mean       sd      skew     kurt    n
[1,] 19.85239 8.212585 0.3752283 2.301911 4329
[2,] 18.97977 8.940397 0.3526516 2.145847 4152
```

The function `freqtab` creates a frequency table from scratch, using a vector of scores and the corresponding score scale. With an anchor test this becomes a bivariate frequency table for forms x and y, and the arguments sent to `freqtab` are the vectors of total scores and anchor scores, and the total and anchor score scales:

```
> neat.x <- freqtab(KBneat$x[, 1], KBneat$x[, 2], xscale = 0:36, vscale = 0:12)
> neat.y <- freqtab(KBneat$y[, 1], KBneat$y[, 2], xscale = 0:36, vscale = 0:12)
> neat.x[50:55,]


     x  v count
[1,] 3 10     0
```

```
[2,] 3 11    0
[3,] 3 12    0
[4,] 4  0    0
[5,] 4  1    4
[6,] 4  2    3
```

These bivariate tables contain all possible score combinations in columns 1 and 2, along with the number of examinees obtaining each combination in column 3. For example, rows 50 through 55 are displayed above for form $X$, where counts for 6 $X$ and $V$ score combinations are shown. Based on the scale lengths, tables for `neat.x` and `neat.y` contain $37 \times 13 = 481$ rows of scores, many of which have counts of zero.

The `freqtab` function can also be used to tabulate scored item responses, where the arguments `xitems` and `vitems` contain the columns over which total scores will be calculated. For example, the following code creates a frequency table for the reading clusters in PISA booklet 1, with clusters R1 containing unique items and R3A the anchor items:

```
> r1items <- paste(items$itemid[items$clusterid == "r1"])
> r3aitems <- paste(items$itemid[items$clusterid == "r3a"])
> r1r3a <- freqtab(students[students$book == 1, ],
+        xitems = c(r1items, r3aitems), vitems = r3aitems)
> descript(r1r3a)


        mean       sd      skew     kurt   n
[1,] 17.05172 6.429468 -0.3052625 2.080696 406
```

Total scores could also be used to obtain the same result:

```
> descript(freqtab(totals$b1$r1 + totals$b1$r3a, totals$b1$r3a))


        mean       sd      skew     kurt   n
[1,] 17.05172 6.429468 -0.3052625 2.080696 406
```

The **equate** package provides a basic plot method for tables of class `"freqtab"`. Univariate frequency tables (up to two) are plotted together as lines with `type = "h"`. For a single bivariate frequency table a scatter plot with marginal barplots is produced (see Figures 1 and 2).

```
> plot(x = act.x, y = act.y, lwd = 2, xlab = "Score", ylab = "Count")
```

```
> plot(neat.x)
```

Finally, presmoothing options are available for equipercentile equating. Three methods are currently supported, all of which can be requested from within the `equate` function. Two of the methods are designed to adjust (i.e., increase) frequencies falling below a specified threshold. Frequency averaging (described by Moses and Holland 2008), using `freqavg`, replaces scores falling below `jmin` with averages based on adjacent scores:

```
> cbind(act.x, avg = freqavg(act.x, jmin = 2))[1:5,]


     x count  avg
[1,] 0     0 1.25
[2,] 1     1 1.25
[3,] 2     1 1.25
[4,] 3     3 1.25
[5,] 4     9 9.00
```

In columns 1 and 2 are the scale and original counts for `act.x`. Column three contains the adjusted counts which are averaged based on any score points with counts below 2 (scores of 0, 1, and 2), along with the next adjacent value (score of 3, with count of 3). The function `freqbump` simply adds a small relative frequency (`jmin`) to each score point while adjusting the probabilities to sum to one (as described by Kolen and Brennan 2004, p. 48).

As described above and in Appendix A.2, polynomial loglinear smoothing is a flexible option for reducing irregularities throughout the score distribution. In the **equate** package a loglinear model is fit using the function `loglinear`, which calls on `glm`. Model terms are specified with either a set of score functions (see `example(loglinear)`), or simply by including the degree of the highest desired polynomial term. Here, the bivariate distribution of $X$ and $V$ is smoothed with `degree = 3`, and a frequency table is created from the fitted values. The smoothed distributions in Figure 3 can be compared to the unsmoothed ones in Figure 2. Descriptive statistics show that the smoothed distributions match the unsmoothed in the first three moments.

```
> neat.x.smoothout <- loglinear(neat.x, degree = 3)
> neat.xs <- as.freqtab(cbind(neat.x[, 1:2], neat.x.smoothout))
> rbind(descript(neat.x), descript(neat.xs))


          mean       sd      skew      kurt    n
[1,] 15.82054 6.529799 0.5797331 2.720015 1655
[2,] 15.82054 6.529799 0.5797331 3.270636 1655


> rbind(descript(neat.x[, -1]), descript(neat.xs[, -1]))


         mean       sd      skew      kurt    n
[1,] 5.106344 2.376742 0.4115535 2.766619 1655
[2,] 5.106344 2.376742 0.4115535 2.976297 1655


> plot(neat.xs)
```

The `loglinear` function can also be used to compare model fit for a sequence of nested models. With the argument `compare = TRUE`, an ANOVA table of deviance statistics is returned with likelihood ratio $\chi^2$ tests.

```
> loglinear(neat.x, degree = 3, compare = TRUE)
```

```
Analysis of Deviance Table

Model 1: f ~ V1
Model 2: f ~ V1 + V2
Model 3: f ~ V1 + V2 + V3
Model 4: f ~ V1 + V2 + V3 + V4
Model 5: f ~ V1 + V2 + V3 + V4 + V5
Model 6: f ~ V1 + V2 + V3 + V4 + V5 + V6
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       479     4669.0
2       478     3666.3  1  1002.73 < 2.2e-16 ***
3       477     3559.9  1   106.41 < 2.2e-16 ***
4       476     3464.9  1    94.96 < 2.2e-16 ***
5       475     2593.3  1   871.64 < 2.2e-16 ***
6       474     2551.9  1    41.38 1.256e-10 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

### 4.3. The `equate` Function

Most of the functionality of the **equate** package can be accessed via `equate`, which integrates the equating types and methods described above into a single function. The equivalent groups design provides a simple example, where, besides the frequency tables, only the equating `type` need be specified:

```
> equate(act.x, act.y, type = "mean")


Mean Equating: Equivalent Groups

Summary Statistics:
      mean     sd   skew   kurt    n
x  19.8524 8.2126 0.3752 2.3019 4329
y  18.9798 8.9404 0.3527 2.1458 4152
yx 18.9798 8.2126 0.3752 2.3019 4329

Coefficients:
intercept    slope
  -0.8726   1.0000
```

Summary statistics and the intercept and slope are printed (for a full description of available output see `?equate`). The nonequivalent groups design is requested by specifying an equating `method`:

```
> neat <- equate(neat.x, neat.y, type = "equip",
+         method = "chained")
```
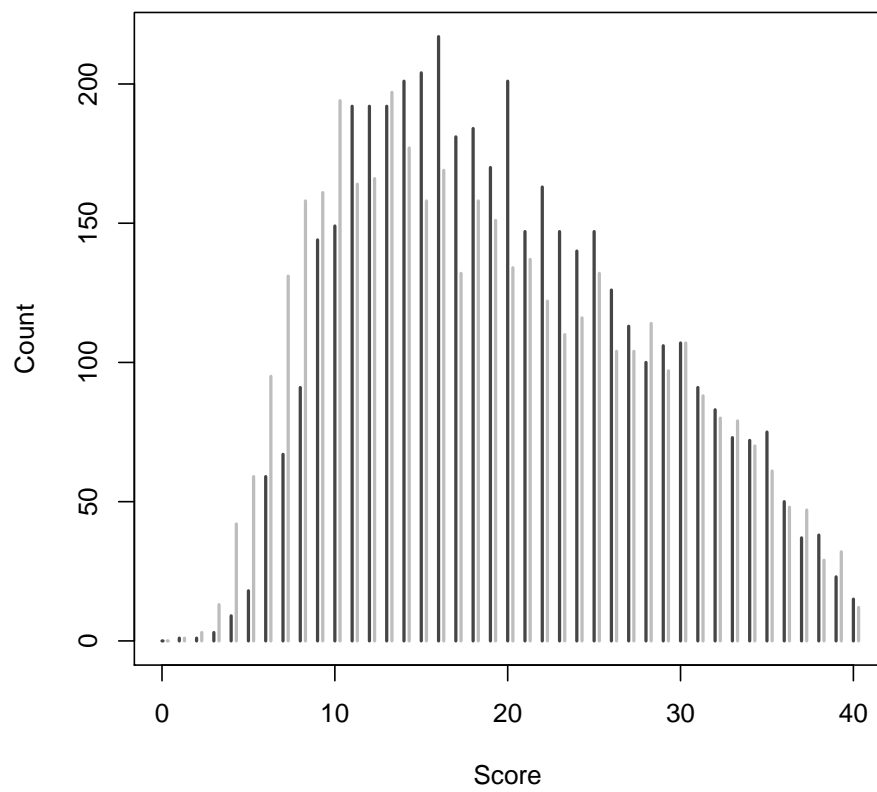
Figure 1: Univariate Plot of ACTmath forms X (dark) and Y (light)
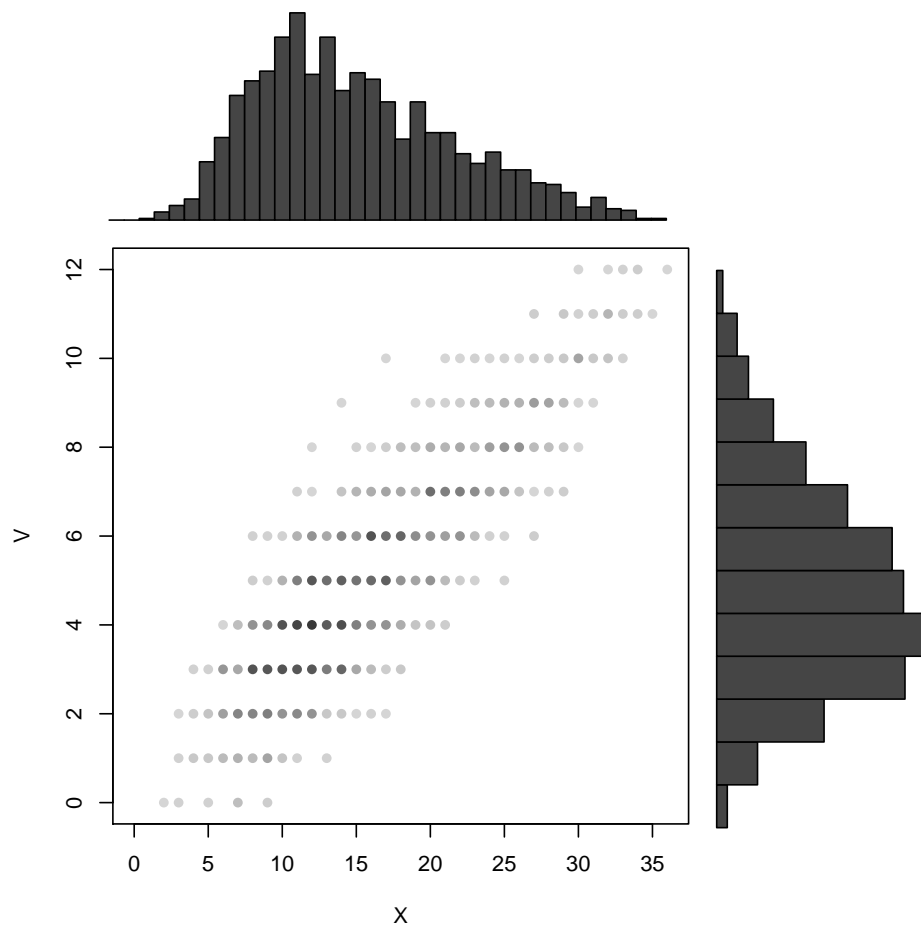
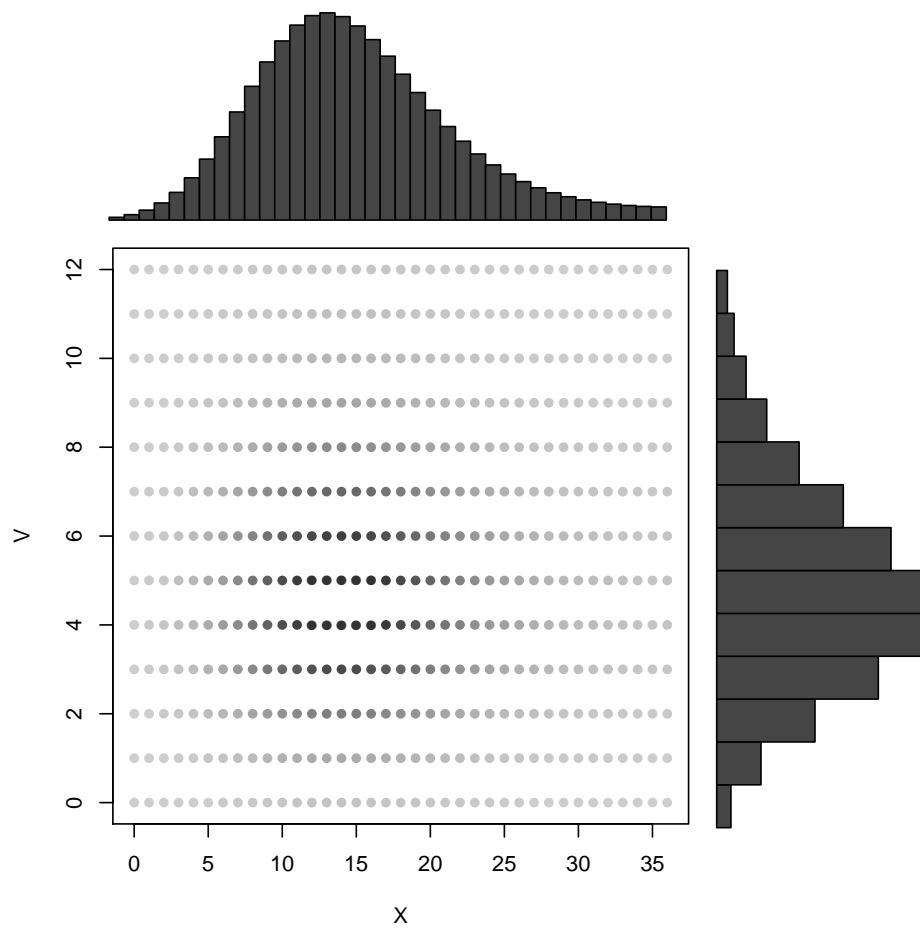Figure 2: Bivariate Plot of KBneat Total (X) and Anchor (V) Distributions

Figure 3: Bivariate Plot of Smoothed KBneat Total (X) and Anchor (V) Distributions

```
Chained Equipercentile Equating: Nonequivalent Groups

Summary Statistics:
      mean     sd   skew   kurt     n
x  15.8205 6.5298 0.5797 2.7200 1655
y  18.6728 6.8805 0.2051 2.3014 1638
yx 16.5556 6.5909 0.5439 2.6925 1655
xv  5.1063 2.3767 0.4116 2.7666 1655
yv  5.8626 2.4522 0.1072 2.5089 1638
```

Table 1 above summarizes the equating methods that apply to each equating type in the nonequivalent groups design. For convenience, these may all be specified in the `equate` function using only the first letter, as in `type="c"` for circle-arc equating. Levine true-score equating (`lts`) is requested by including the additional argument `lts=TRUE`.

The `equate` function can also be used to convert scores from one scale to another based on the function from a previous equating. For example, scores on $Y$ for a few more examinees taking `KBneat` form $X$ could be obtained:

```
> cbind(newx = c(3, 29, 8, 7, 13),
+         yx = equate(x = c(3, 29, 8, 7, 13), y = neat))


     newx       yx
[1,]    3  4.375000
[2,]   29 30.722689
[3,]    8  9.492958
[4,]    7  7.580986
[5,]   13 14.729046
```

Here, the argument `y` passed to `equate` is the chained equipercentile equating from above, which is an object of class `"equate"`. The `equate` function recognizes it as such and attempts to perform the conversion. Note that since the equating function from `neat` relates scores on $X$ to the scale of $Y$, anchor test scores are not needed for the examinees `newx`.

## 4.4. Comparing Equatings

There are many considerations involved in choosing a type and method for equating two test forms (see Kolen and Brennan 2004, ch. 8). Sample size is a major concern, as statistical equating involves the estimation of different numbers of parameters, and accurate estimation depends on adequate and representative samples. As shown above, each equating type and method creates an equating function using different estimates of the score distributions. The equated equivalent at a given score point can vary substantially across equating methods, and within a single equating method across examinee samples.

When samples are small[2] or inadequate for a specific method, random sampling error can become problematic. Random error can be indexed by the standard error of equating ($SEE$),

---

[2]Kolen and Brennan (2004) refer to "small" as less than 100. Other literature discusses small-sample equating with 20-30 examinees per form, for example Livingston (1993) and Skaggs (2005)

which is defined as the standard deviation of equated scores for a given $x_i$ over multiple repeated equatings (systematic error is an important consideration, but is not as easily estimated; see Appendix A.4). The **equate** package provides estimates of linear and equipercentile $SEE$ under the random groups design, based on equations derived by Braun and Holland (1982) and (Lord 1982, p. 168). Additionally, bootstrap standard errors are obtained through the **equate** function using the argument `bootse`:

```
> boots <- equate(act.x, act.y, type = "lin", bootse = TRUE)$bootse
> round(boots, 4)

 [1] 0.2916 0.2819 0.2725 0.2635 0.2549 0.2467 0.2391 0.2320 0.2256 0.2198
[11] 0.2147 0.2104 0.2069 0.2042 0.2025 0.2016 0.2017 0.2027 0.2047 0.2075
[21] 0.2112 0.2156 0.2209 0.2268 0.2334 0.2406 0.2483 0.2565 0.2652 0.2743
[31] 0.2838 0.2936 0.3036 0.3140 0.3246 0.3355 0.3465 0.3577 0.3691 0.3806
[41] 0.3923
```

The sample size taken with each bootstrap replication is specified via `xn` and `yn`, the number of replications via `reps`, and the matrix of equated scores (one column per replication) is requested by setting `returnboots=TRUE` (see `?se.boot` for details).

The example below compares mean and linear Tucker and Levine equating, frequency estimation and chained equipercentile equating, and circle-arc chained (linear) and Tucker (mean) equating of the forms `neat.x` and `neat.y`. Thus there are eight separate nonequivalent groups equatings (see Appendix B.1 for R code). Table 2 contains $Y$ equivalents of scores on $X$ for each (R code in Appendix B.2). The conversion table reveals that equated scores vary somewhat by method. Equipercentile equating with frequency estimation (e.f) produced the highest scores of any method between $X = 5$ and $X = 32$. The largest difference between equated scores was between e.f and mean Levine (m.l) at $X = 21$, a difference of 3.25 points on $Y$. Across methods the smallest equated scores came from circle-arc Tucker equating (c.t) at score points $X < 3, X > 31$, linear Levine (l.l) at points $2 < X < 16$, and mean Levine (m.l) at scores of $15 > X < 32$.

In Figure 4 are plotted the bootstrap standard errors (code for this plot is found in Appendix B.3). The four equating types exhibit a clear trend in $SEE$ across the score scale. As expected, $SEE$ for both mean equatings do not vary by score point, since the scores are equated by a constant amount. Also as expected, random error for linear equating is lowest in the center (slightly lower than estimates for mean) and increases in the tails of the distribution. Overall $SEE$ for equipercentile equating appear to be the largest, despite the fact that the raw score distributions were smoothed. Finally, random error for circle-arc equating is lowest overall, though values increase toward the center of the distribution.

Since equating methods do not extend across all types, they are most easily compared within equating type. Tucker mean outperforms Levine mean; however, the opposite is true for linear equating where Levine $SEE$ are smaller than Tucker across the scale. Until a score on $X$ of 10, values for the two equipercentile methods are comparable. Beyond $X = 10$ random error for chained equating is much lower. Finally, circle-arc equating using the Tucker method to obtain the midpoint results in the lowest $SEE$ of all, values about half as large as those of the chained circle-arc.

Again, it is important to note that random sampling error paints only half the picture when describing equating accuracy. Though a method such as Tucker circle-arc results in some
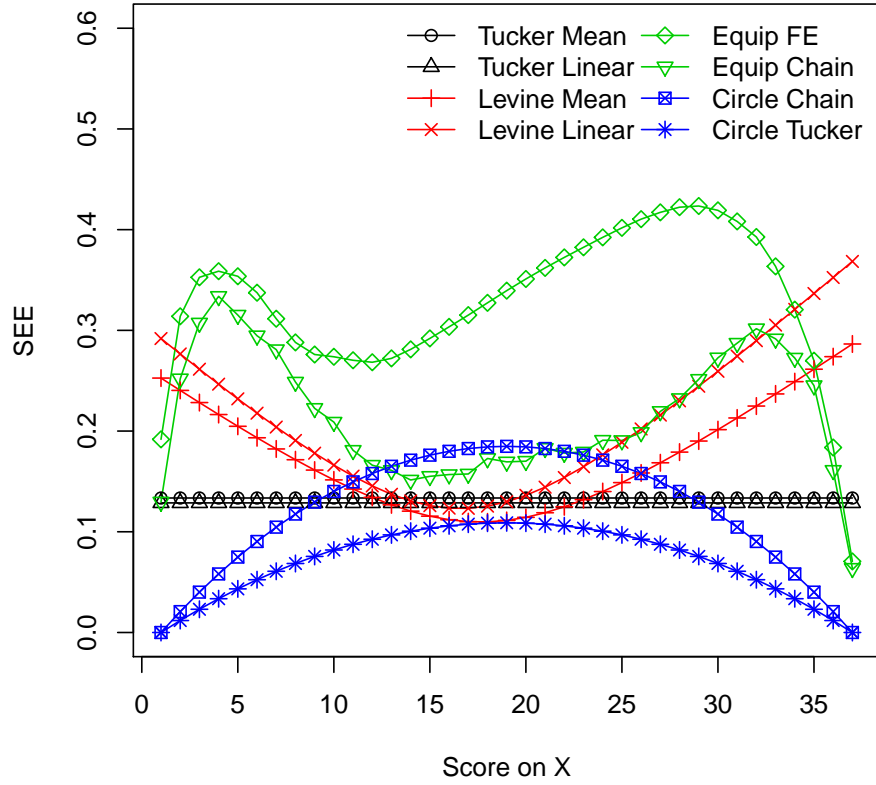
Figure 4: Bootstrap Standard Errors for Eight Nonequivalent Groups Equatings

*SEE* of nearly zero, it may very well be that the estimates are stable (i.e., not varying) around a point that is far from the true equated score. Nevertheless, this example serves to demonstrate the ease with which multiple equatings can be conducted and compared using the `equate` function.

Table 2: Form Y Equivalents for Eight Nonequivalent Groups Equatings

| scale | m.t | m.l | l.t | l.l | e.f | e.c | c.c | c.t |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.995 | 0.428 | 0.537 | 0.251 | 0.191 | 0.038 | 0.000 | 0.000 |
| 1 | 1.995 | 1.428 | 1.566 | 1.263 | 1.486 | 1.142 | 1.233 | 1.109 |
| 2 | 2.995 | 2.428 | 2.595 | 2.274 | 2.677 | 2.295 | 2.451 | 2.212 |
| 3 | 3.995 | 3.428 | 3.624 | 3.285 | 3.879 | 3.472 | 3.656 | 3.309 |
| 4 | 4.995 | 4.428 | 4.653 | 4.296 | 5.097 | 4.535 | 4.847 | 4.400 |
| 5 | 5.995 | 5.428 | 5.682 | 5.307 | 6.326 | 5.554 | 6.025 | 5.484 |
| 6 | 6.995 | 6.428 | 6.710 | 6.318 | 7.546 | 6.593 | 7.189 | 6.562 |
| 7 | 7.995 | 7.428 | 7.739 | 7.330 | 8.731 | 7.610 | 8.339 | 7.633 |
| 8 | 8.995 | 8.428 | 8.768 | 8.341 | 9.925 | 8.620 | 9.477 | 8.699 |
| 9 | 9.995 | 9.428 | 9.797 | 9.352 | 11.123 | 9.631 | 10.601 | 9.758 |
| 10 | 10.995 | 10.428 | 10.826 | 10.363 | 12.320 | 10.661 | 11.711 | 10.811 |
| 11 | 11.995 | 11.428 | 11.855 | 11.374 | 13.511 | 11.659 | 12.809 | 11.857 |
| 12 | 12.995 | 12.428 | 12.884 | 12.385 | 14.682 | 12.692 | 13.893 | 12.898 |
| 13 | 13.995 | 13.428 | 13.913 | 13.396 | 15.850 | 13.722 | 14.965 | 13.932 |
| 14 | 14.995 | 14.428 | 14.942 | 14.408 | 17.010 | 14.696 | 16.023 | 14.960 |
| 15 | 15.995 | 15.428 | 15.971 | 15.419 | 18.159 | 15.781 | 17.068 | 15.982 |
| 16 | 16.995 | 16.428 | 17.000 | 16.430 | 19.294 | 16.797 | 18.101 | 16.997 |
| 17 | 17.995 | 17.428 | 18.029 | 17.441 | 20.411 | 17.808 | 19.120 | 18.006 |
| 18 | 18.995 | 18.428 | 19.058 | 18.452 | 21.509 | 18.881 | 20.127 | 19.010 |
| 19 | 19.995 | 19.428 | 20.087 | 19.463 | 22.591 | 19.878 | 21.120 | 20.006 |
| 20 | 20.995 | 20.428 | 21.116 | 20.475 | 23.647 | 20.918 | 22.101 | 20.997 |
| 21 | 21.995 | 21.428 | 22.145 | 21.486 | 24.675 | 21.967 | 23.068 | 21.982 |
| 22 | 22.995 | 22.428 | 23.174 | 22.497 | 25.669 | 22.948 | 24.023 | 22.960 |
| 23 | 23.995 | 23.428 | 24.203 | 23.508 | 26.626 | 23.980 | 24.965 | 23.932 |
| 24 | 24.995 | 24.428 | 25.232 | 24.519 | 27.542 | 24.998 | 25.893 | 24.898 |
| 25 | 25.995 | 25.428 | 26.260 | 25.530 | 28.425 | 25.963 | 26.809 | 25.857 |
| 26 | 26.995 | 26.428 | 27.289 | 26.542 | 29.272 | 26.947 | 27.711 | 26.811 |
| 27 | 27.995 | 27.428 | 28.318 | 27.553 | 30.075 | 27.921 | 28.601 | 27.758 |
| 28 | 28.995 | 28.428 | 29.347 | 28.564 | 30.829 | 28.864 | 29.477 | 28.699 |
| 29 | 29.995 | 29.428 | 30.376 | 29.575 | 31.534 | 29.788 | 30.339 | 29.633 |
| 30 | 30.995 | 30.428 | 31.405 | 30.586 | 32.235 | 30.705 | 31.189 | 30.562 |
| 31 | 31.995 | 31.428 | 32.434 | 31.597 | 32.898 | 31.621 | 32.025 | 31.484 |
| 32 | 32.995 | 32.428 | 33.463 | 32.609 | 33.522 | 32.524 | 32.847 | 32.400 |
| 33 | 33.995 | 33.428 | 34.492 | 33.620 | 34.171 | 33.402 | 33.656 | 33.309 |
| 34 | 34.995 | 34.428 | 35.521 | 34.631 | 34.805 | 34.323 | 34.451 | 34.212 |
| 35 | 35.995 | 35.428 | 36.550 | 35.642 | 35.445 | 35.205 | 35.233 | 35.109 |
| 36 | 36.995 | 36.428 | 37.579 | 36.653 | 36.150 | 36.057 | 36.000 | 36.000 |

# A. Additional Equations

## A.1. Chained Linear Equating

Chained linear equating involves two separate linear functions. In the equations below the anchor test $V$ is distinguished by population (1 taking form $X$ and 2 taking form $Y$), though the items on $V$ do not change. The first linear function in slope-intercept form converts $X$ to the scale of $V_1$:

$$l_{V_1}(x_i) = \frac{\hat{\sigma}(V_1)}{\hat{\sigma}(X)} x_i - \frac{\hat{\sigma}(V_1)}{\hat{\sigma}(X)} \hat{\mu}(X) + \hat{\mu}(V_1). \tag{23}$$

The second function converts $V_2$ to the scale of $Y$:

$$l_Y(v_{2i}) = \frac{\hat{\sigma}(Y)}{\hat{\sigma}(V_2)} v_{2i} - \frac{\hat{\sigma}(Y)}{\hat{\sigma}(V_2)} \hat{\mu}(V_2) + \hat{\mu}(Y). \tag{24}$$

These functions are combined, where the first, $l_{V_1}(x_i)$, takes the place of $v_{2i}$ in the second to obtain:

$$lchain_Y(x_i) = \frac{\hat{\sigma}(Y)}{\hat{\sigma}(V_2)} \left[ \frac{\hat{\sigma}(V_1)}{\hat{\sigma}(X)} x_i - \frac{\hat{\sigma}(V_1)}{\hat{\sigma}(X)} \hat{\mu}(X) + \hat{\mu}(V_1) \right] - \frac{\hat{\sigma}(Y)}{\hat{\sigma}(V_2)} \hat{\mu}(V_2) + \hat{\mu}(Y), \tag{25}$$

or, in slope-intercept form, after some rearranging:

$$lchain_Y(x_i) = \frac{\hat{\sigma}(Y)}{\hat{\sigma}(V_2)} \frac{\hat{\sigma}(V_1)}{\hat{\sigma}(X)} x_i + \frac{\hat{\sigma}(Y)}{\hat{\sigma}(V_2)} \left[ \hat{\mu}(V_1) - \frac{\hat{\sigma}(V_1)}{\hat{\sigma}(X)} \hat{\mu}(X) - \hat{\mu}(V_2) \right] + \hat{\mu}(Y). \tag{26}$$

Finally, for chained mean equating this reduces to:

$$mchain_Y(x_i) = x_i + \hat{\mu}(V_1) - \hat{\mu}(X) - \hat{\mu}(V_2) + \hat{\mu}(Y). \tag{27}$$

When used to obtain the midpoint coordinates in circle-arc equating, the chained method reduces even further, since $x_i$ is $\hat{\mu}(X)$. Here, the linear and mean functions simplify to

$$lchain_Y(\hat{\mu}(X)) = \frac{\hat{\sigma}(Y)}{\hat{\sigma}(V_2)} \hat{\mu}(V_1) - \frac{\hat{\sigma}(Y)}{\hat{\sigma}(V_2)} \hat{\mu}(V_2) + \hat{\mu}(Y), \tag{28}$$

and

$$mchain_Y(\hat{\mu}(X)) = \hat{\mu}(V_1) - \hat{\mu}(V_2) + \hat{\mu}(Y). \tag{29}$$

## A.2. Loglinear Presmoothing

Polynomial loglinear modeling is a flexible procedure for smoothing distributions of various shapes to varying degrees. The structure of a distribution can either be maintained or ignored depending on the complexity of the model, where the degree of the polynomial term included determines the moment of the raw score distribution to be preserved. For example, a model with terms to the first, second, and third powers would create a smoothed distribution which matches the raw in mean, variance, and skewness. In the model below, the log of the expected relative frequency $p_i$ at score point $i$ is expressed in terms of a normalizing constant ($\beta_0$) and three weighted score functions ($x_1$, $x_2$, $x_3$) of the possible score values of test $X$:

$$log(p_i) = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \beta_3 x_i^3. \tag{30}$$

Indicator variables may also be included to preserve specific moments for subsets of score points. In the next model the mean and variance of a sub-distribution are preserved, in addition to the first three moments of the full distribution. When $S_i = 1$, score point $i$ is included in this sub-distribution, and when $S_i = 0$, it is ignored:

$$log(p_i) = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_{S0} S_i + \beta_{S1} x_i^1 S_i + \beta_{S2} x_i^2 S_i. \tag{31}$$

An acceptable degree of smoothing is typically achieved by comparing multiple models with different numbers of polynomial terms based on their fit to the data (Kolen and Brennan 2004). The `loglinear` function in **equate** is a wrapper for the `glm` function in the **stats** package. It can be used to fit and compare nested models up to a maximum polynomial term. For details, see `?loglinear`.

## A.3. Circle-Arc Equating

The circle-arc in circle-arc equating is a section of the circle that is defined by the vertical distance of the three points $(x_1, y_1)$, $(x_2, y_2)$, and $(x_3, y_3)$ from the line $lin_Y(x)$. Since the low and high points define the line $lin_Y(x)$, they reduce to $(x_1, 0)$ and $(x_3, 0)$. The new midpoint is identified as $(x_2, y_{2*})$. These three points are used to determine the coordinates $x_c$ and $y_c$ for the center of the circle:

$$x_c = \frac{(x_3^2 - x_1^2)}{2(x_3 - x_1)}, \tag{32}$$

$$y_c = \frac{(x_1^2)(x_3 - x_2) - (x_2^2 + y_{2*}^2)(x_3 - x_1) + (x_3^2)(x_2 - x_1)}{2[y_{2*}(x_1 - x_3)]}. \tag{33}$$

These center points are then used to obtain the radius

$$r^2 = (x_1 - x_c)^2 + (y_{1*} - y_c)^2. \tag{34}$$

Since $y_{1*} = 0$ ($x_3$ and $y_{3*}$ could also be used) this reduces to

$$r = \sqrt{(x_1 - x_c)^2 + (y_c)^2}. \tag{35}$$

## A.4. Error in Equating

In the literature, equatings are typically compared based on both random and systematic error, where the first is estimated by the standard error of equating ($SEE$ or simply $SE$) and the second by the $Bias$. As demonstrated above, estimates of $SEE$ can be obtained through bootstrap resampling from the sample score distributions. However, both the $SE$ and $Bias$ are defined in terms of the population equating function. Using a generic equating function $g_Y(x_i)$ to represent a score on $X$ equated to $Y$, the systematic error is calculated as

$$Bias = \hat{\bar{g}}_Y(x_i) - g_Y(x_i), \tag{36}$$

where $g_Y(x_i)$ is the population equating equivalent and

$$\hat{\bar{g}}_Y(x_i) = \frac{1}{R} \sum_{r=1}^{R} \hat{g}_{Yr}(x_i) \tag{37}$$

is the average estimated equivalent over $R$ samples. The random error is defined as

$$SE = \frac{1}{R}\sqrt{\sum_{r=1}^{R}[\hat{g}_{Yr}(x_i) - \hat{\bar{g}}_Y(x_i)]^2}. \tag{38}$$

And combining both systematic error and random error, the root mean squared error ($RMSE$) is defined as

$$RMSE = \sqrt{\{Bias\}^2 + \{SE\}^2}. \tag{39}$$

# B. Additional **R** Code

## B.1. Eight Equatings

```
> # Save each of the eight equatings
> # Note: the two equipercentile runs may be slow
> neat.m.t <- equate(neat.x, neat.y, type = "m", method = "t", bootse=TRUE)
> neat.m.l <- equate(neat.x, neat.y, type="m", method="l", bootse=TRUE)
> neat.l.t <- equate(neat.x, neat.y, type="l", method="t", bootse=TRUE)
> neat.l.l <- equate(neat.x, neat.y, type="l", method="l", bootse=TRUE)
> neat.e.f <- equate(neat.x, neat.y, type="e", method="f", bootse=TRUE,
+         smooth="loglin", degree=3)
> neat.e.c <- equate(neat.x, neat.y, type="e", method="c", bootse=TRUE,
+         smooth="loglin", degree=3)
> neat.c.c <- equate(neat.x, neat.y, type="c", method="c", bootse=TRUE)
> neat.c.t <- equate(neat.x, neat.y, type="c", method="t", bootse=TRUE)
```

## B.2. Concordance Table

```
> concordance <- cbind(
+         neat.m.t$conc,
+         neat.m.l$conc[,2],
+         neat.l.t$conc[,2],
+         neat.l.l$conc[,2],
+         neat.e.f$conc[,2],
+         neat.e.c$conc[,2],
+         neat.c.c$conc[,2],
+         neat.c.t$conc[,2])
> colnames(concordance)[-1] <-
+         c("m.t","m.l","l.t","l.l","e.f","e.c","c.c","c.t")
```

## B.3. Plotting Bootstrap SEE

```
> # Plot comparing bootstrap SEE
> plot(c(1, 37), c(0, .6), type = "n", xlab = "Score on X", ylab = "SEE")
> points(neat.m.t$bootsee, col = 1, type = "l")
> points(neat.m.l$bootsee, col = 1, type = "l")
> points(neat.l.t$bootsee, col = 2, type = "l")
> points(neat.l.l$bootsee, col = 2, type = "l")
> points(neat.e.f$bootsee, col = 3, type = "l")
> points(neat.e.c$bootsee, col = 3, type = "l")
> points(neat.c.c$bootsee, col = 4, type = "l")
> points(neat.c.t$bootsee, col = 4, type = "l")
```

```
> points(neat.m.t$bootsee, col = 1, type = "p", pch = 1)
> points(neat.m.l$bootsee, col = 1, type = "p", pch = 2)
> points(neat.l.t$bootsee, col = 2, type = "p", pch = 3)
> points(neat.l.l$bootsee, col = 2, type = "p", pch = 4)
> points(neat.e.f$bootsee, col = 3, type = "p", pch = 5)
> points(neat.e.c$bootsee, col = 3, type = "p", pch = 6)
> points(neat.c.c$bootsee, col = 4, type = "p", pch = 7)
> points(neat.c.t$bootsee, col = 4, type = "p", pch = 8)
> legend("topright", legend = c("Tucker Mean", "Tucker Linear", "Levine Mean",
+          "Levine Linear", "Equip FE", "Equip Chain", "Circle Chain", "Circle Tucker"),
+          col = rep(1:4, each = 2), pch = 1:8, lty = 1, bty = "n", ncol = 2)
```

# References

Albano AD (2013). **equate: Statistical Methods for Test Equating.** R package version 1.2-0, URL http://CRAN.R-project.org/package=equate.

Babcock B, Albano AD, Raymond M (2012). "Nominal Weights Mean Equating: A Method for Very Small Samples." *Educational and Psychological Measurement*, **72**, 608–628.

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. New York, NY: Academic.

Hanson BA (1991). "A Note on Levine's Formula for Equating Unequally Reliable Tests Using Data from the Common Item Nonequivalent Groups Design." *Journal of Educational and Behavioral Statistics*, **16**, 93.

Holland PW, Dorans NJ (2006). "Linking and Equating." In RL Brennan (ed.), *Educational Measurement*, 4 edition, pp. 187–220. Westport, CT: Greenwood.

Holland PW, Thayer DT (2000). "Univariate and Bivariate Loglinear Models for Discrete Test Score Distributions." *Journal of Educational and Behavioral Statistics*, **25**(2), 133–183.

Kim S, von Davier AA, Haberman S (2008). "Small-Sample Equating Using a Synthetic Linking Function." *Journal of Educational Measurement*, **45**, 325–342.

Kolen MJ (1984). "Effectiveness of Analytic Smoothing in Equipercentile Equating." *Journal of Educational and Behavioral Statistics*, **9**, 25–44.

Kolen MJ, Brennan RL (2004). *Test Equating, Scaling, and Linking.* New York, NY: Springer.

Livingston SA (1993). "Small-Sample Equating with Log-linear Smoothing." *Journal of Educational Measurement*, pp. 23–39.

Livingston SA (2004). *Equating Test Scores (Without IRT).* Princeton, NJ: ETS.

Livingston SA, Dorans NJ, Wright NK (1990). "What Combination of Sampling and Equating Methods Works Best?" *Applied Measurement in Education*, **3**, 73–95.

Livingston SA, Kim S (2009). "The Circle-Arc Method for Equating in Small Samples." *Journal of Educational Measurement*, **46**, 330–343.

Lord FM (1982). "The Standard Error of Equipercentile Equating." *Journal of Educational Statistics*, **7**, 165–174.

Moses TP, Holland PW (2008). *Notes on a General Framework for Observed Score Equating (ETS Research Rep. No. RR-08-59)*. Princeton, NJ: ETS.

Skaggs G (2005). "Accuracy of Random Groups Equating with Very Small Samples." *Journal of Educational Measurement*, **42**, 309–330.

Weeks JP (2010). "**plink**: An R Package for Linking Mixed-Format Tests Using IRT-Based Methods." *Journal of Statistical Software*, **35**(12), 1–33. URL http://www.jstatsoft.org/v35/i12/.

**Affiliation:**

Anthony D. Albano
Department of Educational Psychology
College of Education and Human Sciences
University of Nebraska-Lincoln
Lincoln, NE 68516, USA
E-mail: albano@unl.edu