Paper PO-11

POSTEQUATE: A SAS® Macro for Conducting Non-IRT Test Post-Equating

Isaac Li, University of South Florida, Tampa, FL
Jeffrey D. Kromrey, University of South Florida, Tampa, FL

ABSTRACT

Testing organizations publish multiple editions of an exam (test forms) for practical purposes. Equating studies are conducted to ensure score interchangeability between different test forms. This paper presents a SAS macro that conducts post-equating using six different methods. The macro reads in response data from the two test forms to be equated and outputs two conversion tables converting every possible total score on one form to a set of six equivalent scores on the other form calculated using these methods. In addition, this macro produces score distributions of the two forms. The paper provides a demonstration of the SAS code, an example of its application with sample data, and related output.

Keywords: EQUATING, NEAT DESIGN

INTRODUCTION

Procedures in equating or scale linking are required to achieve comparability across examination forms as stipulated in the Standards for Educational and Psychological Testing (AERA, APA, & NCME, 1999). Those studies which utilize item statistics derived from previous administrations of the exam are labeled as pre-equating. The other class of studies that analyzes collected response data directly after the exam administration is called post-equating. Despite the popularity of Item Response Theory, in many situations post-equating studies rely on non-IRT statistical methods, which usually require simpler processes and produce results easier to explain (Livingston, 2004). Post-equating procedures include implementing a data collection plan before the exam and selecting statistical methods to estimate score correspondence between the old and the new form after the exam.

Nonequivalent Anchor Test (NEAT) design. Various data collection plans are in use but the post-equating methods discussed in this paper are all based on the NEAT design. In this equating plan, alternative forms share a group of items in common (anchor items) that are representative of the whole form in content and statistical properties and these forms are given to different groups of examinees which may differ in their abilities. Although the score on the anchor items can either be included or excluded from the total score, the anchor items allow the assessment of differences in capabilities between the two groups.

The synthetic group in the NEAT design. Braun and Holland (1982) proposed the notion of the synthetic group formed by weighted aggregation of the two groups of examinees which took each of the two exam forms to be equated. In practice there are different ways of weighting dependent on practical considerations, however, it has been found that how to weight the groups has a minimal effect on the resulting score conversion (Kolen & Brennan, 1987). Suppose population 1 took Form *X* and population 2 took Form *Y*, there are four parameters with regards to the synthetic population as described in Kolen and Brennan (2004):

$$\mu_s(X) = w_1\mu_1(X) + w_2\mu_2(X)$$

$$\mu_s(Y) = w_1\mu_1(Y) + w_2\mu_2(Y)$$

$$\sigma_s^2(X) = w_1\sigma_1^2(X) + w_2\sigma_2^2(X) + w_1w_2[\mu_1(X) - \mu_2(X)]^2$$

and

$$\sigma_s^2(Y) = w_1\sigma_1^2(Y) + w_2\sigma_2^2(Y) + w_1w_2[\mu_1(Y) - \mu_2(Y)]^2$$

where $\mu_s(X)$, $\sigma_s^2(X)$, and $\mu_s(Y)$, $\sigma_s^2(Y)$ are means and variances of the synthetic group on the two forms respectively. Since population 1 never took Form *Y* and population 2 never took Form *X*, parameters of $\mu_2(X)$, $\sigma_2^2(X)$, $\mu_1(Y)$, and $\sigma_1^2(Y)$ need to be estimated. For the weighting variables, $w_1 + w_2 = 1$.

EQUATING METHODS

The Tucker Method. The Tucker linear method assumes the linear regression function of total scores on anchor test scores for both population 1 and 2 being the same. Given a regression slope *α*, a regression intercept *β*, and the total score on all common items (anchors) *V*, this regression assumption can be described as

$$\alpha_1(X|V) = \alpha_2(X|V), \alpha_1(Y|V) = \alpha_2(Y|V)$$

$$\beta_1(X|V) = \beta_2(X|V), \beta_1(Y|V) = \beta_2(Y|V).$$

The other assumption is that the conditional variances of scores *X* and *Y* given *V* are the same for population 1 and 2. With these assumptions, it becomes possible to estimate γ_1, the slope resulting from the regression of *X* on *V* for population 1, and γ_2, the slope from a regression of *Y* on *V* for population 2:

$$\gamma_1 = \frac{\sigma_1(X,V)}{\sigma_1^2(V)}, \gamma_2 = \frac{\sigma_2(X,V)}{\sigma_2^2(V)}.$$

The covariance between the total score and common item score on the new form, $\sigma_1(X,V)$, can be directly estimated for population 1, and the same is true with that of the old form for population 2, $\sigma_2(Y,V)$, which then provide estimates of the two slopes. These estimates of regression slopes in turn help obtain the values of $\mu_s(X)$, $\sigma_s(X)$, $\mu_s(Y)$, and $\sigma_s(Y)$ as follows.

$$\mu_s(X) = \mu_1(X) - w_2\gamma_1[\mu_1(V) - \mu_2(V)]$$

$$\mu_s(Y) = \mu_2(Y) - w_1\gamma_2[\mu_1(V) - \mu_2(V)]$$

$$\sigma_s^2(X) = \sigma_1^2(X) - w_2\gamma_1^2[\sigma_1^2(V) - \sigma_2^2(V)] + w_1w_2\gamma_1^2[\mu_1(V) - \mu_2(V)]^2$$

$$\sigma_s^2(Y) = \sigma_2^2(Y) - w_1\gamma_2^2[\sigma_1^2(V) - \sigma_2^2(V)] + w_1w_2\gamma_2^2[\mu_1(V) - \mu_2(V)]^2$$

By utilizing standard deviation scores, the Tucker method then converts observed scores on Form *X* to the scale of observed scores on Form *Y* using this linear function

$$l_{Y_s}(x) = \frac{\sigma_s(Y)}{\sigma_s(X)}[x - \mu_s(X)] + \sigma_s(Y).$$

The Levine Observed Score Method. The first Levine method utilizes the same linear conversion as above but is based on different assumptions. It assumes that the correlation between true scores of the two populations are always one, either for *X* and *V* or for *Y* and *V*. Secondly, it assumes that the two regression coefficients of true scores, T_V on T_X and T_V on T_Y, are equal. The third assumption holds that the measurement error variance for *X*, *Y*, and *V* is the same across the two populations. Therefore, in the case of the anchor test being part of the whole exam (internal anchor), the relationship between *X* and *V* and relationship between *Y* and *V* can be expressed as

$$\gamma_1 = \frac{\sigma_1^2(X)}{\sigma_1(X,V)}, \gamma_2 = \frac{\sigma_2^2(Y)}{\sigma_2(X,V)}.$$

Since the pieces on the right of the equations can be estimated directly from observed data, the two regression coefficients can be calculated. Subsequently, using the same five linear equations above, observed scores on Form *X* are converted to the scale of observed scores on Form *Y*

The Levine True Score Method. This method is based on the same assumptions by the observed score method. However, Levine true score method utilizes a different linear function to convert true scores on *X* to the scale of true scores on *Y*:

$$l_Y(x) = \frac{\gamma_2}{\gamma_1}[x - \mu_1(X)] + \mu_2(Y) + \gamma_2[\mu_1(V) - \mu_2(V)].$$

In this equation, $\boldsymbol{\gamma_1}$ and $\boldsymbol{\gamma_2}$ are calculated the same way as with the Levine observed score method while the other pieces can all be computed directly from observed response data. Note that the weights have no role in this method.

Mean equating with the Tucker method and the Levine methods. For exam forms with fewer than 100 examinees, applying linear equating could result in very large equating errors. Mean equating is appropriate in this situation (Kolen & Brennan, 2004, *p*. 125). For Tucker and Levine observed score methods, which share the same conversion equation, mean equating is achieved by specifying

$$\frac{\sigma_s(Y)}{\sigma_s(X)} = 1$$

to reach the new shared equation as

$$m_{Y_s}(x) = [x - \mu_s(X)] + \sigma_s(Y).$$

Similarly, the Levine true score method can be switched to mean equating by setting

$$\frac{\gamma_2}{\gamma_1} = 1$$

to arrive at the conversion equation

$$m_Y(x) = [x - \mu_1(X)] + \mu_2(Y) + \gamma_2[\mu_1(V) - \mu_2(V)].$$

The elements of these mean equating equations are calculated the same way as described above.

MACRO POSTEQUATE

A SAS macro was written to perform the non-IRT numeric equating methods described above. The macro was developed to provide researchers with an easily accessible tool for conducting equating studies over collected exam response data. At this moment, it is compatible with only dichotomous items. Nonetheless, it will help testing organizations deploying SAS software to verify their pre-equating results.

First, the macro compares the score distributions of the two exam forms with two histogram graphs, one for total scores and the other for common item scores. Next in the equating analysis, it generates various statistics that will be used to compute equivalent scores such as means, standard deviations, intercepts, and slopes. At the end, the macro outputs one conversion table that equates every possible score on the new form to six equivalent scores on the old form computed with the six equating methods. Reversely, it also produces a conversion table that converts the scores on the old form to those on the new form using the same regression slopes and intercepts.

```
options mprint;
%macro postequate (oldfile=, newfile=, w1=, nitems=, ncommon=);

* +---------------------------------------------------------------------------+
    Read in data files for old and new form
* +---------------------------------------------------------------------------+;
 data old;
  infile "&oldfile" missover;
  input @1 (item1 - item&nitems) (1.) ;
  oldtot = sum (of item1 - item&nitems);
  oldcom = sum (of item1 - item&ncommon);
  if oldtot = 0 then delete;
  group = "Old";

 data new;
  infile "&newfile" missover;
  input @1 (item1 - item&nitems) (1.) ;
  newtot = sum (of item1 - item&nitems);
  newcom = sum (of item1 - item&ncommon);
  if newtot = 0 then delete;
  group = "New";
 run;

 data both;
  set old new;
  Total_Score = sum (of item1 - item&nitems);
  Common_Score = sum (of item1 - item&ncommon);
 run;
* +---------------------------------------------------------------------------+
    Outputs histogram of distributions
* +---------------------------------------------------------------------------+;
 proc univariate data = both noprint;
  class group;
  histogram Total_Score / nohlabel cfill = ltgray ctext = blue;
  inset n = 'Examinees' median (8.2) mean (8.2) std = 'Standard Deviation' (8.3)
  / position = nw;
 label group = 'population';
 run;

 proc univariate data = both noprint;
  class group;
```

```
  histogram Common_Score / nohlabel cfill = ltgray ctext = blue
            midpoints = 0 1 2 3 4 5 6 7 8 9 10;
  inset n = 'Examinees' median (8.2) mean (8.2) std = 'Standard Deviation' (8.3)
  / position = nw;
 label group = 'population';
 run;
* +-----------------------------------------------------------------------------+
    Produce statistical moments for two forms separately
* +-----------------------------------------------------------------------------+;
 proc corr outp = old_corr data = old cov noprint;
  var oldtot oldcom;
 run;

 data _null_;
  set old_corr;
   if lowcase(_type_) = 'mean' then call symputx ('oldtotmn',oldtot );
   if lowcase(_type_) = 'mean' then call symputx ('oldcommn',oldcom );
   if lowcase(_type_) = 'std' then call symputx ('oldtotstd',oldtot );
   if lowcase(_type_) = 'std' then call symputx ('oldcomstd',oldcom );
   if lowcase(_type_) = 'n' then call symputx ('oldtotn',oldtot );
   if lowcase(_type_) = 'n' then call symputx ('oldcomn',oldcom );
   if lowcase(_type_) = 'cov' and lowcase(_name_) = 'oldcom' then call symputx
('oldcov',oldtot );
   if lowcase(_type_) = 'corr' and lowcase(_name_)= 'oldcom' then call symputx
('oldcor',oldtot );
 run;
%put &oldtotmn &oldcommn &oldtotstd &oldcomstd &oldtotn &oldcomn &oldcov &oldcor;

 proc corr outp = new_corr data = new cov noprint;
  var newtot newcom;
 run;

 data _null_;
  set new_corr;
   if lowcase(_type_) = 'mean' then call symputx ('newtotmn',newtot );
   if lowcase(_type_) = 'mean' then call symputx ('newcommn',newcom );
   if lowcase(_type_) = 'std' then call symputx ('newtotstd',newtot );
   if lowcase(_type_) = 'std' then call symputx ('newcomstd',newcom );
   if lowcase(_type_) = 'n' then call symputx ('newtotn',newtot );
   if lowcase(_type_) = 'n' then call symputx ('newcomn',newcom );
   if lowcase(_type_) = 'cov' and lowcase(_name_) = 'newcom' then call symputx
('newcov',newtot );
   if lowcase(_type_) = 'corr' and lowcase(_name_)= 'newcom' then call symputx
('newcor',newtot );
 run;
%put &newtotmn &newcommn &newtotstd &newcomstd &newtotn &newcomn &newcov &newcor;

* +-----------------------------------------------------------------------------+
    Populates a table of weights, means, regression coefficients, means, etc.
* +-----------------------------------------------------------------------------+;
data elements;

 method = "TLN"; *Tucker linear;
 w1 = &w1;
w2 = 1-w1;
 gamma1 =&newcov/(&newcomstd**2) ; *4.21;
 gamma2 =&oldcov/(&oldcomstd**2) ; *4.22;
 musx = &newtotmn ;
 musy = &oldtotmn + w1*gamma2*(&newcommn - &oldcommn);
 sigmasx = &newtotstd;
 sigmasy = sqrt(&oldtotstd**2+w1*gamma2**2*(&newcomstd**2 -
&oldcomstd**2)+w1*w2*gamma2**2*(&newcommn-&oldcommn)**2);
 slp = sigmasy/sigmasx;
 int = musy - slp*musx ;
 output;
```

```
 method = "LLN"; *Levine linear;
 w1 = &w1;
 w2 = 1-w1;
 gamma1 =&newcov/(&newcomstd**2) ;
 gamma2 = (&oldtotstd**2)/&oldcov ;
 musx = &newtotmn ;
 musy = &oldtotmn + w1*gamma2*(&newcommn - &oldcommn);
 sigmasx = &newtotstd;
 sigmasy = sqrt(&oldtotstd**2+w1*gamma2**2*(&newcomstd**2 -
&oldcomstd**2)+w1*w2*gamma2**2*(&newcommn-&oldcommn)**2);
 slp = sigmasy/sigmasx;
 int = musy - slp*musx ;
 output;

 method = "LTR"; *Levine true score;
 w1 = .;
 w2 = .;
 gamma1 = &newtotstd**2/&newcov ;
 gamma2 = (&oldtotstd**2)/&oldcov ;
 musx = &newtotmn ;
 musy = &oldtotmn + gamma2*(&newcommn - &oldcommn);
 sigmasx = .;
 sigmasy = .;
 slp = gamma2/gamma1;
 int = -slp*&newtotmn + &oldtotmn + gamma2*(&newcommn - &oldcommn);
 output;


 method = "TMN"; *Tucker mean;
 w1 = &w1;
 w2 = 1-w1;
 gamma1 =&newcov/(&newcomstd**2) ; *4.21;
 gamma2 =&oldcov/(&oldcomstd**2) ; *4.22;
 musx = &newtotmn ;
 musy = &oldtotmn + w1*gamma2*(&newcommn - &oldcommn);
 sigmasx = &newtotstd;
 sigmasy = sqrt(&oldtotstd**2+w1*gamma2**2*(&newcomstd**2 -
&oldcomstd**2)+w1*w2*gamma2**2*(&newcommn-&oldcommn)**2);
 slp = 1;
 int = musy - slp*musx ;
 output;

 method = "LMN"; *Levine mean;
 w1 = &w1;
 w2 = 1-w1;
 gamma1 =&newcov/(&newcomstd**2) ;
 gamma2 = (&oldtotstd**2)/&oldcov ;
 musx = &newtotmn ;
 musy = &oldtotmn + gamma2*(&newcommn - &oldcommn);
 sigmasx = &newtotstd;
 sigmasy = sqrt(&oldtotstd**2+w1*gamma2**2*(&newcomstd**2 - oldcomstd**2)+
               w1*w2*gamma2**2*(&newcommn-&oldcommn)**2);
 slp = 1;
 int = musy - slp*musx ;
 output;

 method = "LTM"; *Levine true score mean;
 w1 = .;
 w2 = .;
 gamma1 = &newtotstd**2/&newcov ;
 gamma2 = (&oldtotstd**2)/&oldcov ;
 musx = &newtotmn ;
 musy = &oldtotmn + gamma2*(&newcommn - &oldcommn);
 sigmasx = .;
```

```
sigmasy = .;
slp = 1;
int = -slp*&newtotmn + &oldtotmn + gamma2*(&newcommn - &oldcommn);
output;

* +--------------------------------------------------------------------------+
   Call intercepts and slopes for conversion
* +--------------------------------------------------------------------------+;
data _null_;
 set elements;

 if method = 'TLN' then call symputx ('tlnslp',slp);
 if method = 'TLN' then call symputx ('tlnint',int);
 if method = 'LLN' then call symputx ('llnslp',slp);
 if method = 'LLN' then call symputx ('llnint',int);
 if method = 'LTR' then call symputx ('ltrslp',slp);
 if method = 'LTR' then call symputx ('ltrint',int);
 if method = 'TMN' then call symputx ('tmnslp',slp);
 if method = 'TMN' then call symputx ('tmnint',int);
 if method = 'LMN' then call symputx ('lmnslp',slp);
 if method = 'LMN' then call symputx ('lmnint',int);
 if method = 'LTM' then call symputx ('ltmslp',slp);
 if method = 'LTM' then call symputx ('ltmint',int);

run;

* +--------------------------------------------------------------------------+
   Creates conversion tables
* +--------------------------------------------------------------------------+;
data scores;
 name = "scores";
 x0 = 0;
 array x(&nitems) x1 - x&nitems;
  xscore=0;
  do i =1 to &nitems;
  xscore =xscore + 1;
  x(i) =xscore;
 end;
 drop xscore i;
run;

proc transpose data = scores out = convtable prefix = x;
 var x0 - x&nitems;
run;

data convtable1 (drop=_name_);
 set convtable;
 rename x1 = x;
 tln = x1*&tlnslp + &tlnint;
 lln = x1*&llnslp + &llnint;
 ltr = x1*&ltrslp + &ltrint;
 tmn = x1*&tmnslp + &tmnint;
 lmn = x1*&lmnslp + &lmnint;
 ltm = x1*&ltmslp + &ltmint;
 label x1="X Scores" tln="Tucker Linear" lln="Levine Linear" ltr="Levine True Score"
                     tmn="Tucker Mean" lmn="Levine Mean" ltm="Levine True Score Mean";
run;

title "Conversion Table - New to Old";
proc print data=convtable1 noobs round label;
run;

data convtable2 (drop=_name_);
 set convtable;
 rename x1 = y;
```

```
 tln = (x1 - &tlnint)/&tlnslp;
 lln = (x1 - &llnint)/&llnslp;
 ltr = (x1 - &ltrint)/&ltrslp;
 tmn = (x1 - &tmnint)/&tmnslp;
 lmn = (x1 - &lmnint)/&lmnslp;
 ltm = (x1 - &ltmint)/&ltmslp;
 label x1="Y Scores" tln="Tucker Linear" lln="Levine Linear" ltr="Levine True Score"
                   tmn="Tucker Mean" lmn="Levine Mean" ltm="Levine True Score Mean";
run;

title "Conversion Table - Old to New";
proc print data=convtable2 noobs round label;
run;

%mend;
```

INVOKING THE MACRO

Inputs to the macro include the paths and names of the two data files in plain text for the new and old form respectively (oldfile= and newfile=), the weight to be used for the new form (*w*1=, 0 ≤ w1 ≤ 1), the total number of items on either exam (nitems=), and the number of common items (ncommon=). The preparation of the two input data sets is straightforward. For either form, scores of all items (1's and 0's) need to be compiled into a matrix, with the scores of each item occupying a column. Scores of common items must be placed in front of the columns for non-common items. For example, if 10 out of the 36 items on the test are common items, the first 10 columns of the two input data files should be scores for these 10 items, which will then be aggregated as the common item scores. The records below can be an example of the input file, where the first examinee had a total score of 18 and a common item score of 7 out of 10 items.

```
111010101100111110111010000011000000
111111111111111111111111111111111111
100011011101101010111100110000100000
…
```

The beginning part of the macro that reads in the input can be modified to accommodate the format of the score files to be analyzed. However, users will want to keep two item lists to record the sequence of items appearing in the two input files. This macro can be invoked by such a call.

```
%postequate (oldfile=C:\old.txt, newfile=C:\new.txt, w1=1, nitems=36, ncommon=10);
```

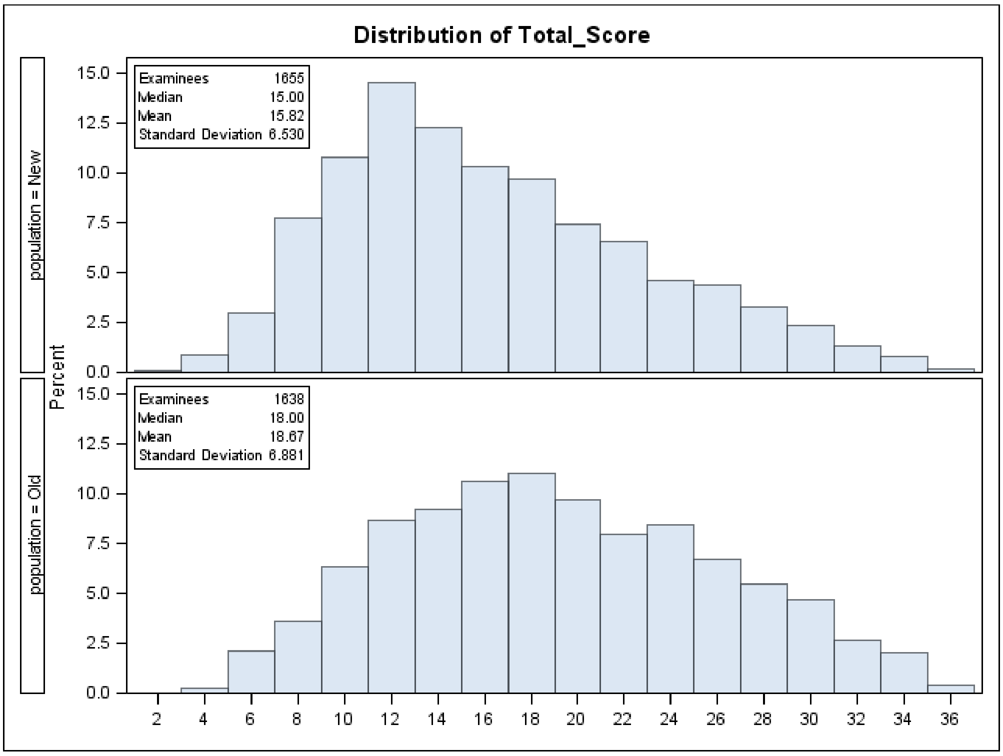
OUTPUT FROM THE MACRO

To demonstrate the outputs of this macro, data sets used by Kolen and Brennan for illustration (2004, *p*. 211) were downloaded and analyzed. The old form had 1,638 examinees and the new form got 1,655 takers. Output 1 contrasts the total score distributions on the two forms, showing the group of examinees taking the old form scoring higher than that taking the new form. The first 10 items were arbitrarily designated as the common items. In Output 2, it is shown that the two groups had roughly equal scores on common items. Output 3 lists the excerpt of the most commonly used conversion table in equating studies. For example, a score of 10 on the new form would be equivalent to a score of 10.48 based on the Tucker linear method but 9.10 using the Levine observed score method. Giving only the top half of the table, Output 4 simply provides the same equating results in reverse direction. For example, a score of 10 on the old form would be equivalent to a score of 9.57 based on the Tucker linear method and 10.75 with the Levine observed score method.

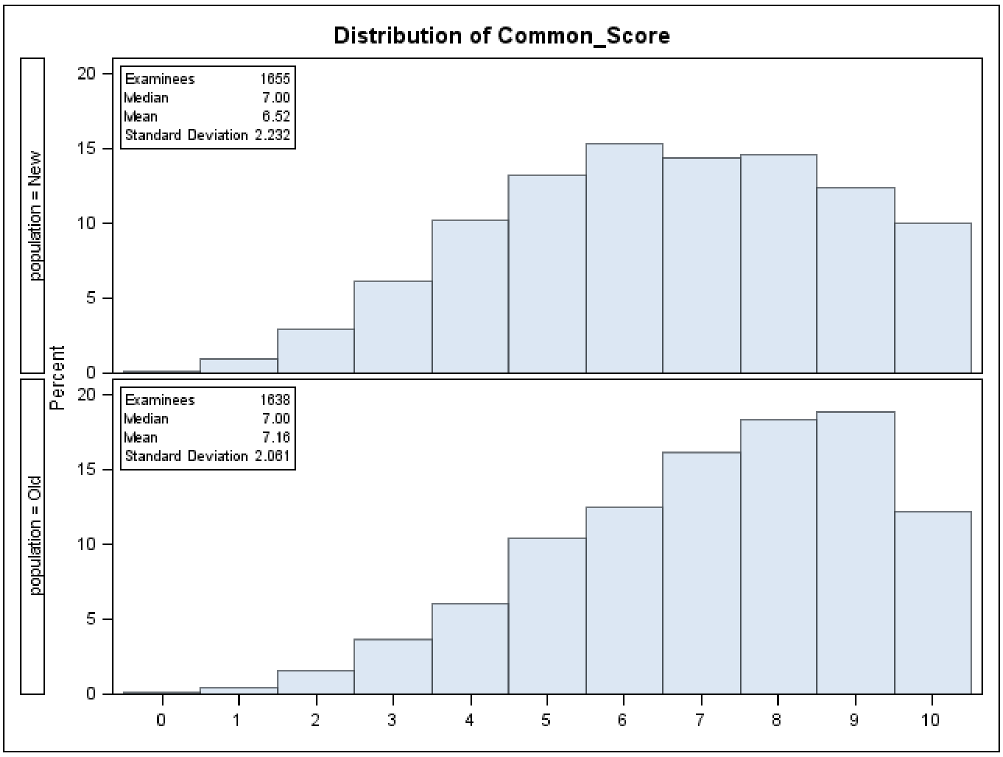
CONCLUSIONS

Even as Item Response Theory and its applications have become the primary psychometric research methods, traditional methods to conduct exam form equating remain indispensible in many situations. For example, various programs in professional licensure and certification examinations attract such small numbers of candidates to sit for each exam administration that IRT equating is impossible for them. The few existing specialized computer programs for test equating (e.g., *CIPE* and *ST*) were created long ago and run on archaic interfaces. In fact, traditional equating methods such as those discussed in this paper do not require complex mathematics and should be easy to implement with general-purpose statistical packages. This SAS macro was written to perform several non-IRT numeric equating methods. The macro was developed to provide researchers with an easily accessible tool for conducting equating studies over collected exam response data. At this moment, it is compatible with only dichotomous items and the design of internal anchors. Nonetheless, it will help testing organizations that have acquired SAS software to verify their pre-equating results.

Future developments to this macro will include other numerical equating methods, such as equipercentile equating. It will also provide estimates of standard error of equating.



Output 1. Example Output of Total Score Distributions for Both Groups



Output 2. Example Output of Score Distributions of Common Items for Both Groups

Conversion Table - New to Old

X Scores	Tucker Linear	Levine Linear	Levine True Score	Tucker Mean	Levine Mean	Levine True Score Mean
0	-0.63	-2.76	-2.67	1.13	0.19	0.19
1	0.48	-1.57	-1.49	2.13	1.19	1.19
2	1.59	-0.39	-0.31	3.13	2.19	2.19
3	2.70	0.80	0.87	4.13	3.19	3.19
4	3.81	1.99	2.05	5.13	4.19	4.19
5	4.93	3.17	3.23	6.13	5.19	5.19
6	6.04	4.36	4.42	7.13	6.19	6.19
7	7.15	5.55	5.60	8.13	7.19	7.19
8	8.26	6.73	6.78	9.13	8.19	8.19
9	9.37	7.92	7.96	10.13	9.19	9.19
10	10.48	9.10	9.14	11.13	10.19	10.19
11	11.59	10.29	10.32	12.13	11.19	11.19
12	12.70	11.48	11.50	13.13	12.19	12.19
13	13.82	12.66	12.68	14.13	13.19	13.19
14	14.93	13.85	13.86	15.13	14.19	14.19
15	16.04	15.04	15.04	16.13	15.19	15.19
16	17.15	16.22	16.22	17.13	16.19	16.19
17	18.26	17.41	17.40	18.13	17.19	17.19
18	19.37	18.60	18.58	19.13	18.19	18.19
19	20.48	19.78	19.76	20.13	19.19	19.19
20	21.59	20.97	20.94	21.13	20.19	20.19

Output 3. Excerpt of the Conversion Table to Equate the New Form to the Old Form

Conversion Table - Old to New

Y Scores	Tucker Linear	Levine Linear	Levine True Score	Tucker Mean	Levine Mean	Levine True Score Mean
0	0.57	2.33	2.26	-1.13	-0.19	-0.19
1	1.47	3.17	3.11	-0.13	0.81	0.81
2	2.37	4.01	3.95	0.87	1.81	1.81
3	3.27	4.85	4.80	1.87	2.81	2.81
4	4.17	5.70	5.65	2.87	3.81	3.81
5	5.07	6.54	6.49	3.87	4.81	4.81
6	5.97	7.38	7.34	4.87	5.81	5.81
7	6.87	8.23	8.19	5.87	6.81	6.81
8	7.77	9.07	9.04	6.87	7.81	7.81
9	8.67	9.91	9.88	7.87	8.81	8.81
10	9.57	10.75	10.73	8.87	9.81	9.81
11	10.47	11.60	11.58	9.87	10.81	10.81
12	11.37	12.44	12.42	10.87	11.81	11.81
13	12.27	13.28	13.27	11.87	12.81	12.81
14	13.17	14.13	14.12	12.87	13.81	13.81
15	14.07	14.97	14.96	13.87	14.81	14.81
16	14.97	15.81	15.81	14.87	15.81	15.81
17	15.87	16.65	16.66	15.87	16.81	16.81
18	16.77	17.50	17.51	16.87	17.81	17.81
19	17.67	18.34	18.35	17.87	18.81	18.81
20	18.57	19.18	19.20	18.87	19.81	19.81

Output 4 Excerpt of the Conversion Table to Equate the Old Form to the New Form

REFERENCES

AERA, APA, & NCME (1999). *Standards for educational and psychological testing*. Washington, D.C.: Author.

Braun, H. I., & Holland, P. W. (1982). Observed-score test equating: A mathematical analysis of some ETS equating procedures. In P. W. Holland & D. B. Rubin (Eds.), *Test equating* (pp. 9{49). New York, NY: Academic.

Holland, P. W., & Dorans, N. J. (2006). Linking and equating. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 187-220). Westport, CT: Greenwood.

Kolen, M. J., & Brennan, R. L. (1987). Linear equating models for the common item nonequivalent population design. *Applied Psychological Measurement*, 11, 263-277.

Kolen, M. J. (1988). Traditional equating methodology. *Educational Measurement: Issues and Practice*, Winter, 29-36.

Kolen, M. J. (2004). CIPE (*Windows Console Version*). [Computer software and manual]. Iowa City, IA: Center for Advanced Studies in Measurement and Assessment, The University of Iowa. (Available on http://www.education.uiowa.edu/casma).

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

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

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please contact Isaac Li at:

University of South Florida
4202 East Fowler Ave. EDU 162
Tampa, FL 33620
Work Phone: 407-264-2993
Fax: 813-974-4495
Email: ili@mail.usf.edu