A SAS Macro for Loglinear Smoothing: Applications and Implications

Tim P. Moses

Alina A. von Davier

Tim P. Moses

Alina A. von Davier

Research & Development

An SAS Macro for Loglinear Smoothing: Applications and Implications

Tim P. Moses and Alina A. von Davier

ETS, Princeton, NJ

April 2006

As part of its educational and social mission and in fulfilling the organization's nonprofit charter and bylaws, ETS has and continues to learn from and also to lead research that furthers educational and measurement research to advance quality and equity in education and assessment for all users of the organization's products and services.

ETS Research Reports provide preliminary and limited dissemination of ETS research prior to publication. To obtain a PDF or a print copy of a report, please visit:

http://www.ets.org/research/contact.html

Abstract

The two purposes of this paper are to provide a SAS IML macro that performs loglinear smoothing and to apply this macro to loglinear smoothing problems that have not been extensively discussed in the test-equating literature. The SAS macro is demonstrated on univariate, bivariate, and trivariate smoothing problems. The univariate and bivariate examples reproduce published results (von Davier, Holland, & Thayer, 2004). The trivariate example extends the bivariate smoothing example to allow for comparisons of subgroups' univariate and bivariate distributions. The implications are that important questions about distribution differences and subpopulation invariance of equating functions can be considered through comparisons and evaluations of complex loglinear models that are easily fit with this SAS IML macro.

Key words: Loglinear smoothing, SAS IML, equating

Acknowledgments

Table of Contents

Introduction

Polynomial loglinear models for one-, two- and higher-way contingency tables (Bock & Yates, 1973; Haberman, 1974a, 1978, 1979) have important applications to measurement and assessment (Hanson, 1991; Holland & Thayer, 1987, 2000; Rosenbaum & Thayer, 1987). Two such applications are test score distribution estimation and comparison (Kolen, 1991; Hanson, 1996). Another application is the estimation and enhancement of test-equating stability (von Davier, Holland, & Thayer, 2004; Holland & Thayer, 1989; Kolen & Brennan, 1995; Livingston, 1993; Skaggs, 2004). In these applications, the polynomial loglinear models are essentially regarded as a smoothing technique that is commonly referred to as loglinear smoothing.

In an effort to make loglinear smoothing more readily available, reports have described how it can be implemented with SAS/STAT PROC GENMOD (Moses & von Davier, 2004; Moses, von Davier, & Casabianca, 2004; SAS Institute, 2002a). PROC GENMOD is flexible and adequate for most simple univariate smoothing problems. However, it can have convergence problems for some bivariate loglinear smoothing problems. Moreover, PROC GENMOD does not directly provide the so-called "C-matrices"—that is, the low-rank matrix factors of the covariance matrix of the estimated probabilities (von Davier et al., 2004; Holland & Thayer, 1989, 2000) that are important computational tools for the standard errors of the smoothed frequencies and the accuracy measures used in the kernel equating framework.

The possibility of developing a SAS IML (SAS Institute, 2002b) macro that implements loglinear smoothing without the limitations of PROC GENMOD was investigated. The purpose of this paper is to describe this new SAS macro (rather than to demonstrate PROC GENMOD) and to apply it to problems that have not been extensively discussed in the literature. The SAS macro performs loglinear smoothing according to Holland and Thayer's (1987, 2000) specifications. It is appropriate for univariate, bivariate, and trivariate frequency distributions of test data, and it converges even when PROC GENMOD fails. This macro also computes the C-matrix factors.

The first major section of this paper reviews the use of loglinear models for smoothing discrete distributions. The second section describes how to obtain and use the SAS macro to fit loglinear smoothing models. The third and fourth sections demonstrate the macro with respect to a simple univariate smoothing problem and a much more complicated 22-parameter bivariate problem, both from von Davier et al. (2004). The fifth section demonstrates the SAS macro on a

trivariate smoothing problem, where the third variable defines a subgroup that provides a basis for comparing bivariate distributions (see also Liou, Cheng, & Li, 2001). The implications of these applications are discussed in terms of future research with a focus on the study of equating methods.

Loglinear Smoothing Models

Assume we have a random variable X that defines the test form X (we use the same notation for a test form and a random variable) with possible values $x_0,...,x_J$, or x_j, with $j = 0,...,J$ (the possible score values) and a corresponding vector of observed score frequencies $n = (n_0,...,n_J)^t$ that sum to the total sample size N. Under some distributional assumptions about n, like multinomial or Poisson distributional assumptions, the vector of the population score probabilities $\boldsymbol{p} = (p_0,...,p_J)^t$ is said to satisfy a loglinear model if

$$\log_e(p_j) = \alpha + u_j + \mathbf{b_j}\boldsymbol{\beta} \tag{1}$$

where the $\{p_j\}$ are assumed to be positive and sum to one, $\mathbf{b_j}$ is a row vector of constants referred to as score functions throughout this text (e.g., x_j^1, x_j^2, x_j^3), $\boldsymbol{\beta}$ is a vector of free parameters, u_j is a known constant that specifies the distribution of the $\{p_j\}$ when $\boldsymbol{\beta} = 0$, and α is a normalizing constant that ensures that the probabilities sum to one.

Under different choices of $\boldsymbol{u}$, $\boldsymbol{B}$ (the matrix of score functions formed by arranging the row vectors, $\mathbf{b_j}$, one on top of the other), or $\boldsymbol{\beta}$, the loglinear model becomes equivalent to the discrete uniform distribution ($\boldsymbol{u} = 0$, $\boldsymbol{\beta} = 0$) or the binomial distribution (see Holland & Thayer, 1987, 2000, for details).

Loglinear models are a class of exponential families of discrete distributions, which can be described in terms of their sample moments. As in Holland and Thayer (1987, 2000), we will make use of this property and of the fact that the u_j are known constants. Therefore in this paper the loglinear model used to fit a univariate distribution is

$$\log_e(p_j) = \alpha + \sum_{i=1}^{I} \beta_i(x_j)^i, \tag{2}$$

where the u_j are set to zero. When the data are test score data, the terms in this model can be defined as follows: the x_j^i are *score functions* of the possible score values of test X (e.g., $x_j^1, x_j^2,$

2

$x_j^3, \ldots, x_j^I$), α is as described above, and the β_i are free parameters to be estimated in the model-fitting process.

The value of *I* determines the number of moments of the actual test score distribution that are preserved in the smoothed distribution. If *I = 1* then the smoothed distribution preserves the first moment (the mean) of the observed distribution. If *I = 4* then the smoothed distribution preserves the first, second, third, and fourth moments (mean, variance, skewness, and kurtosis) of the observed distribution.

The model in (2) can be extended to fit the bivariate distribution of the scores of two tests (call them *X* and *Y*):

$$\log_e(p_{jk}) = \alpha + \sum_{i=1}^{I} \beta_{xi}(x_j)^i + \sum_{h=1}^{H} \beta_{yh}(y_k)^h + \sum_{g=1}^{G} \sum_{f=1}^{F} \beta_{gf}(x_j)^g (y_k)^f, \qquad (3)$$

where p_{jk} is the joint score probability of the score (x_j, y_k; score x_j on test *X* and score y_k on test *Y*). The fitting of (3) produces a smoothed bivariate distribution that preserves *I* moments in the marginal (univariate) distribution of *X*; *H* moments in the marginal (univariate) distribution of *Y*; and a number of cross-moments ($G \leq I$, $F \leq H$) in the bivariate *X-Y* distribution. Model (3) is also appropriate for the smoothing of bivariate distributions with impossible *X-Y* score combinations, *structural zeros*, when the total test score can never be less than the score on the internal anchor test and the anchor score cannot be less than its maximum possible value to a greater extent than the total test score is less than its maximum possible value (see Holland & Thayer for an example, 2000).

Indicator functions can be used to fit both the full univariate distribution and a subset of the distribution (e.g. *teeth* or lumps at different score points) within a single loglinear model. One example of such a model is:

$$\log_e(p_j) = \alpha + \beta_1(x_j)^1 + \beta_2(x_j)^2 + \beta_3 S_j + \beta_4(x_j)^1 S_j, \qquad (4)$$

where the indicator function $S_j = 1$ if *j* belongs to a defined subset of all *j's* and $S_j = 0$ otherwise. S_j denotes the set of score points where the frequencies are systematically lower or higher than most of the test frequencies. Model (4) will preserve the mean and variance of the

total distribution of X (β_1 and β_2), the total frequency in the cells denoted as $S_j = 1$ (β_3), and the mean of the cell values for the cells in $S_j = 1$ (β_4).

One additional smoothing model combines the bivariate model in (3) with the use of indicator functions in (4):

$$
\begin{aligned}
\log_e(p_{jkl}) = {} & \alpha + \beta_l S_l + \sum_{i=1}^{I} \beta_{xi}(x_j)^i + \sum_{h=1}^{H} \beta_{yh}(y_k)^h + \sum_{g=1}^{G}\sum_{f=1}^{F} \beta_{gf}(x_j)^g (y_k)^f \\
& + \sum_{i=1}^{I} \beta_{xiS}(x_j)^i S_l + \sum_{h=1}^{H} \beta_{yhS}(y_k)^h S_l + \sum_{g=1}^{G}\sum_{f=1}^{F} \beta_{gfS}(x_j)^g (y_k)^f S_l
\end{aligned}
\tag{5}
$$

The model in (5) is useful for preserving features in a trivariate distribution, where p_{jkl} is the probability of score (x_j, y_k) in subgroup $S_l = 0$ or 1. Model (5) preserves the subgroups' frequencies, X and Y univariate moments, and XY cross-moments. Simpler versions of (5) can include fewer subgroup-varying terms and can allow the subgroups' distributions to share certain parameters. For example, less β_{xiS} terms can be included, allowing for a certain number of the lowest univariate moments in X to vary by subgroups, but constraining the higher moments to be equal so that they are shared by the subgroups and equal to those of the total distribution.

Fitting Loglinear Smoothing Models

Under the assumption that the vector of the frequencies is multinomial, the estimation of the free parameters (β_i) proceeds by maximizing the following log-likelihood function:

$$
L = \sum_j n_j \log_e(p_j),
\tag{6}
$$

where n_j and p_j are the observed frequencies and the population score probabilities in the *jth* cell, respectively (Holland & Thayer, 1987, 2000).

The maximization of (6) can be accomplished through the use of the Newton-Raphson algorithm (Holland & Thayer, 1987, p. 11). Holland and Thayer specify two criteria for the convergence solution from the algorithm. One criterion involves the maximization of the log-likelihood function; the maximum is said to be attained when the relative change in the log-likelihood is less than some specified value. The second criterion involves the satisfaction of the

likelihood equation for all of the estimated parameters (β), meaning that the relative error in each fitted moment must be less than some specified value. At convergence both criteria should be met.

To add stability to the Newton-Raphson algorithm, the score functions in **B** are transformed so that they sum to zero and their squares sum to one (Holland & Thayer, 1987, 2000; Rosenbaum & Thayer, 1987). Holland and Thayer also suggest specific starting values. The suggested starting values for the parameter estimates are based on converting the observed frequencies into a smoother form with nonzero frequencies at all score points and then computing a function of these converted frequencies and the score functions.

Large-sample standard errors of the estimated parameters (β) can be estimated when the Newton-Raphson algorithm converges to a maximum likelihood solution. The parameter estimates and standard errors that correspond to the higher moments are misleadingly small because they are coefficients of scores raised to high powers. If the comparability of parameter estimates is of interest, a preferable approach to defining score functions in terms of powers would be to define them as orthogonal polynomials (Haberman, 1974a).

At convergence, the variance-covariance matrix of **β** is given as

$$Cov(\widehat{\boldsymbol{\beta}}) = (\boldsymbol{B}^t Cov(\boldsymbol{n})\boldsymbol{B})^{-1}, \tag{7}$$

where $Cov(\boldsymbol{n}) = N(\boldsymbol{D}_p - \boldsymbol{pp}^t)$ and $\boldsymbol{D}_p$ is the diagonal matrix of $\boldsymbol{p}$.

Evaluating the Fit of Loglinear Models

There are several measures that are useful for evaluating the extent to which the smoothed frequencies match the observed frequencies. The likelihood ratio chi-square statistic is given as:

$$G^2 = 2\sum_j n_j \log_e(\frac{n_j}{\hat{p}_j N}), \tag{8}$$

where $\hat{p}_j$ is the smoothed value of p_j based on a particular model. This measure is often used in statistical tests that evaluate the relative fit of nested and competing models (Agresti, 2002; Haberman, 1974b; Hanson, 1996; Holland & Thayer, 2000).

Other measures for overall model fit include the Pearson chi-square statistic,

$$\sum_j \frac{(n_j - \hat{p}_j N)^2}{\hat{p}_j N}, \tag{9}$$

the Freeman-Tukey chi-square statistic,

$$\sum_j \left(\sqrt{n_j} + \sqrt{n_j + 1} - \sqrt{4\hat{p}_j N + 1} \right)^2, \tag{10}$$

and two measures that penalize the overfitting of data, including the Akaike information criterion (AIC; Akaike, 1981, 1987), which adds twice the number of parameters estimated by the model to the likelihood chi-square statistic, and the consistent Akaike information criterion (CAIC; Bozdogan, 1987), which adds 1+log(N) times the number of parameters to the likelihood ratio chi-square statistic. Variants of these measures not computed with the SAS IML macro include other members of Cressie and Read's power-divergence family of chi-square statistics (Read & Cressie, 1988), the Bayesian inference criterion (Schwarz, 1978), and Gilula and Haberman's modification of the AIC (Gilula & Haberman, 1994).

In addition to evaluating overall model fit, it can be useful to compare the smoothed and observed frequencies at each score level using Freeman-Tukey residuals (Freeman & Tukey, 1950),

$$\sqrt{n_j} + \sqrt{n_j + 1} - \sqrt{4\hat{p}_j N + 1}. \tag{11}$$

When a model fits the data well, the Freeman-Tukey residuals are asymptotically normally and randomly distributed with a mean of zero and a variance approaching one. The asymptotic variance of the Freeman-Tukey residuals is less than one and the departure from one depends on the complexity of the model and the sparseness of the data (Bishop, Feinberg, & Holland, 1975 Haberman, 1973, 1974b;). Freeman-Tukey residuals are especially useful for suggesting whether indicator functions or higher moments are warranted in univariate distributions. The residuals become less useful when there are many zeros in the observed frequencies (as in many bivariate problems).

C-Matrices

The estimated variance-covariance matrix of the smoothed probabilities ($\Sigma_{\hat{p}}$) can be used for obtaining their confidence intervals (Holland & Thayer, 1987, 2000) and for computing

standard errors of kernel equating (von Davier et al., 2004). A very useful factorization of the estimated variance-covariance matrix is the C-matrix, defined as:

$$\Sigma_{\hat{p}} = C_p C_p^t,$$ (12)

where C_p is the J by I matrix that can be efficiently computed as:

$$C_p = N^{-1/2} D_{\sqrt{p}} Q.$$ (13)

The diagonal matrix, $D_{\sqrt{p}}$, has the diagonal entries $\sqrt{p_j}$, and Q is the J by I orthogonal matrix that comes from the following QR-factorization:

$$\left[D_{\sqrt{p}} - \sqrt{\hat{p}}\hat{p}^t \right] B = QR.$$ (14)

Q is a J by I matrix with orthogonal columns, R is an I by I upper triangular matrix, and B is the matrix of score functions and shown in (1) (Holland & Thayer, 1987). The QR call routine in SAS returns a J by J matrix (SAS Institute, 2002b), so the SAS IML macro uses the first I columns of the outputted Q in computing the C-matrix (Dongarra, Bunch, Moler, & Stewart, 1979).

A SAS Macro for Loglinear Smoothing

The SAS macro described in this paper is flexible enough to address several of the loglinear smoothing problems described in the literature, including univariate, bivariate, and trivariate problems, and provides all of the fit measures reviewed in the previous section and the C-matrix. The requirements for implementing the macro are: SAS software, SAS IML, and some familiarity with SAS DATA and PROC statements. This macro will be distributed upon request by the first author. The 6-step procedure for implementing the macro within SAS is summarized in Appendix A.

Error Catching

The macro is designed to be user friendly, meaning that if users specify impossible conditions, the macro will output informative messages about what it needs in order to run. For example, if the user misspells their count variable or score functions so that the macro is unable to find them within the specified dataset, the macro will stop running and output one of the following error messages to the SAS log:

7

```
ERROR: Unable To Find Your Count Variable In Your Dataset.

ERROR: Unable To Find Your Score Function Variable(s) In Your Dataset.
```

Additional messages give specific feedback to the user when the macro cannot locate the dataset within the specified library, when the dataset has missing values, or when the user does not list a count variable or any score functions at all.

Limitations

The macro has been found to produce acceptable results for a variety of smoothing problems, but it sometimes fails to converge. When the model contains a large number of parameters (e.g., >10 or 12 moments for some univariate problems), SAS IML is less able to solve the required linear systems that are necessary for computing the Newton-Raphson update for the parameter values. As a result, the macro will terminate and give an error message about singular matrices.

Within SAS IML, the procedure for finding matrices that solve linear systems is intentionally limited by the machine's precision (SAS Institute, 2002b). Even with convergence in the overall solution, the SAS macro will sometimes be unable to compute a matrix inverse required for the standard errors of β. In this situation, everything except the standard errors of β will be produced. Our attempts to work around these constraints considered the use of singular value decomposition to compute the required matrix inverses while using a more liberal singularity criterion. The results of these attempts were converged but incorrect solutions. We therefore treat the SAS-imposed singularity constraint as a necessary balance of the flexible Newton-Raphson algorithm (which allows for the fitting of a variety of different kinds of parameters) and the storage constraints of the SAS system. One promising possibility for improving the convergence rate of the SAS IML macro involves the use of orthogonal polynomials of the scores rather than powers of the scores, a possibility that directly resolves the singularity issues with *B* and, as mentioned earlier, allows for comparisons of the βs.

Convergence Criteria

The strictest usable values for convergence criteria should be no smaller than the square root of machine precision (Press, Teukolsky, Vetterling, & Flannery, 1992, p. 398). Since SAS stores numbers as eight-byte reals, the machine precision is about $1e^{-15}$. Therefore the strictest

convergence criteria in SAS would be $\sqrt{1e^{-15}} \approx 3e^{-8}$. These convergence values may be overly strict, especially because of collinearity issues with **B**. Both criteria are labeled in the macro code so that the user may consider larger values for difficult smoothing problems.

Univariate Smoothing Example

In this section, a univariate smoothing from von Davier et al. (2004, p. 99–105) is reproduced. All of the code and output for this example is provided in the appendices, and the reader is invited to follow along with the analyses and also to evaluate the results in terms of the original work. The data are already in the form of a score distribution, where the test (*Y*) is a 20-item rights-scored test that was taken by 1,455 examinees. Appendix B illustrates how the frequency data are entered into a SAS dataset and also how the score functions needed for a model that preserves the first three moments of the distribution of *Y* are defined. The following model is fit:

$$\log_e(p_j) = \alpha + \beta_1(y_j)^1 + \beta_2(y_j)^2 + \beta_3(y_j)^3 \tag{15}$$

Appendix C shows SAS macro commands and gives the resulting output. Overall, the model fits the data very well, as suggested by the small likelihood ratio chi-square statistic (20.24) relative to the degrees of freedom (17). The dataset outputted from the macro (named "outresults") contains the frequencies and score functions in the original dataset plus smoothed counts, smoothed probabilities, Freeman-Tukey residuals, and the C-matrix. This dataset is shown in Appendix D. Appendix E shows how to obtain the plot of the observed and smoothed frequencies. Appendix F shows how the moment-matching characteristics of the smoothed results can be verified within SAS. This conversion of datasets of smoothed frequencies into datasets of individual scores based on the smoothed frequencies may provide useful inputs into other routines that rely on datasets of individual observations, but it also makes expensive time demands for large frequency tables. The more direct way of computing distribution moments from the probabilities is demonstrated in the trivariate smoothing section.

Bivariate Smoothing Example

In this section, the results of a bivariate loglinear problem from von Davier et al. (2004, p. 155–167) are reproduced. These data come from the fall 2001 national administration of a high-volume testing program. The bivariate distribution is of a total test (*X*) with 78 items (*j* = 0

to 78) and an external anchor (*A*) with 35 items (*l* = 0 to 35). The descriptive statistics, based on a sample of 10,634 examinees, are included in Appendix G. The tests are correlated at .88, and out of the 2,844 possible score combinations, 1,502 have zero frequencies. The 22-parameter model to be described next does not converge using SAS PROC GENMOD.

These data exhibit some unusual patterns that suggest special considerations for the loglinear model. First, the two marginal distributions have teeth, a regular pattern of cells with frequencies that are much lower than those of neighboring cells. The teeth are due to the use of rounded formula scores and are at every 5th score from score 5 on for *X* and from score 2 on for *A*. Second, there are *lumps* (very large frequencies) at score 0 in both marginal distributions, due to the rounding of all negative scores to zero. Because these patterns are due to aspects of the test and the processing of its scores and not to randomness in the sample, they are explicitly incorporated into the bivariate loglinear model.

Under the assumption that the sample bivariate frequencies n_{jl}, have an independent, approximate multinomial distribution with population cell probabilities p_{jl}, the fitted model is the following:

$$\log_e(p_{jl}) = \alpha + \beta_x S_{xo} + \beta_a S_{a0} + \sum_{i=1}^{4} \beta_{xi}(x_j)^i + \sum_{h=1}^{4} \beta_{ah}(a_l)^h + \sum_{g=1}^{2} \sum_{f=1}^{2} \beta_{gf}(x_j)^g(a_l)^f$$

$$+ \sum_{e=0}^{3} \beta_{xe} S_{xs}(x_j)^e + \sum_{d=0}^{3} \beta_{ad} S_{as}(a_l)^d$$

(16)

The terms of (16) are defined as follows.

The p_{jl} are the population probabilities of obtaining score x_j on test *X* and score a_l on test *A*. The α is a scaling constant that constrains the sum of all of the p_{jl}'s to 1. The S_{x0} and S_{a0} terms are indicator functions set to 1 when x_j and a_l are 0 and set to 0 when x_j and a_l are not 0. They will preserve the lumps at zero in *X* and *A*. The four x_j^i and a_l^h terms are univariate score functions (the score on tests *X* and *A* to the power of 1, 2, 3, and 4) that will preserve the first four moments of the marginal distributions of *X* and *A*. The four $x_j^g a_l^f$ terms will preserve four different degrees of dependence in *X* and *A*: *XA* (the covariance), XA^2, X^2A, and X^2A^2. The S_{xs} and S_{as} terms are indicator functions equal to 1 when x_j and a_l are teeth scores and zero otherwise. The $S_{xs}(x_j)^e$ and $S_{xs}(a_l)^d$ terms will preserve the total frequencies of the teeth of *X* and *A* when *e* = *d* = 0 and the first three moments of the distributions of the teeth (*e* and *d* go from 1 to 3). All of the βs are free parameters to be estimated by the model-fitting algorithm.

10

von Davier et al (2004) discuss the importance of three of the cross-moments that describe the joint distribution, XA^2, X^2A, and X^2A^2, in the context of equating. In addition to (16), we also considered an alternative model, (17), that preserves everything in (16) except for the three cross-moments (XA^2, X^2A, and X^2A^2). Model (17) is not shown here. Appendix H shows the SAS code from importing the dataset to defining the needed score functions for fitting the models. Appendix I shows how the SAS macro is called for (16) and (17) and also compares the overall fit statistics from both models.

The model fit statistics for each individual model are not χ^2-distributed with such sparse data, but the use of significance tests for comparing the fit statistics of limited models can still be meaningful (Haberman, 1977). The difference between the likelihood ratio chi-square statistics for (16) and (17) is 600 on 3 degrees of freedom, significant beyond $\underline{p} < .0001$. The two fit statistics that penalize for overfitting models (the AIC and CAIC) are also smaller for the more complex model in (16). These results provide support for including the three cross-moment terms in the model (XA^2, X^2A, and X^2A^2.). Finally, we give the smoothed and observed plots of the marginal distributions of X and A in Appendices J and K, where the smoothed frequencies are based on (16) (von Davier et al., p. 156–157). These plots show close matches between the observed and smoothed distributions, especially with respect to the teeth of the distributions.

Trivariate Smoothing Example

Trivariate loglinear models can be used to consider if and how the score distributions of examinee subgroups differ from the overall score distributions. This trivariate smoothing example extends the 22-parameter bivariate model from the previous section in two ways. First, the probabilities to be modeled are defined in terms of the three variables, test X, test A, and a subgroup function. Second, the model is extended to consider how the subgroups' X, A, and XA distributions differ. Four trivariate models are considered, ranging from very simple, where the subgroups share all of their parameters, to very complex, where the subgroups are considered independent and differ in terms of all of their parameters.

The Subgroup

Because the XA exam is a verbal assessment, an important way to subgroup the examinees is based on their experience with English. This is captured by examinees' responses to a question about their first language. On the basis of responses to this question, examinees were classified

into two groups, one group of 6,867 examinees where English was their first and only language, and a second group of 3,767 examinees who were exposed to languages other than English. The descriptive statistics on tests X and A for these two sets of examinees are presented in Appendix L. The XA correlation is .88 for the examinees with English as their first and only language and .89 for the examinees who were exposed to languages other than English. The statistics reveal differences in the marginal distributions for each examinee group. Specifically, the English-only students are a better-performing and a slightly more homogeneous group.

Four Trivariate Models: (18), (19), (20), and (21)

The first loglinear model fit to the trivariate data is based on the well-fitting bivariate model considered in the previous section. To review, this model preserves the first four moments in X and A, lumps at zero in tests X and A, the frequencies and first three moments of the teeth of tests X and A, and four cross-moments in X and A. To this model we add an indicator function (S_c) defined as 1 for the English-only examinees and zero for the other examinees, which preserves the frequencies in each subgroup. The resulting 23-parameter model is:

$$
\begin{aligned}
\log_e(p_{jlc}) &= \alpha + \beta_x S_{xO} + \beta_a S_{a0} + \sum_{i=1}^{4} \beta_{xi}(x_j)^i + \sum_{h=1}^{4} \beta_a(a_l)^h + \sum_{g=1}^{2}\sum_{f=1}^{2} \beta_{gf}(x_j)^g(a_l)^f \\
&+ \sum_{e=0}^{3} \beta_{xe} S_{xs}(x_j)^e + \sum_{d=0}^{3} \beta_{ad} S_{as}(a_l)^d + \beta_c S_c
\end{aligned}
\tag{18}
$$

Model (18) is nested within two more complex models defined to incorporate specific kinds of subgroup differences in the distributions of X, A, and XA. The first of these two models, (19), evaluates the extent to which subgroup distributions differ with respect to the marginal distributions of X and A. We added four parameters to (18) in order to consider the extent to which the fit of (18) can be improved by allowing the means and variances of X and A to differ by subgroup in a 27-parameter model:

$$
\begin{aligned}
\log_e(p_{jlc}) &= \alpha + \beta_x S_{xO} + \beta_a S_{a0} + \sum_{i=1}^{4} \beta_{xi}(x_j)^i + \sum_{h=1}^{4} \beta_{ah}(a_l)^h + \sum_{g=1}^{2}\sum_{f=1}^{2} \beta_{gf}(x_j)^g(a_l)^f \\
&+ \sum_{e=0}^{3} \beta_{xe} S_{xs}(x_j)^e + \sum_{d=0}^{3} \beta_{ad} S_{as}(a_l)^d + \beta_c S_c + \sum_{i=1}^{2} \beta_{x(i+4)} S_c(x_j)^i + \sum_{h=1}^{2} \beta_{a(h+4)} S_c(a_l)^h
\end{aligned}
\tag{19}
$$

Another model, (20), evaluates the extent to which subgroup distributions differ with respect to their bivariate *XA* distributions:

$$\log_e(p_{jlc}) = \alpha + \beta_x S_{xo} + \beta_a S_{a0} + \sum_{i=1}^{4} \beta_{xi}(x_j)^i + \sum_{h=1}^{4} \beta_{ah}(a_l)^h + \sum_{g=1}^{2} \sum_{f=1}^{2} \beta_{gf}(x_j)^g (a_l)^f$$

$$+ \sum_{e=0}^{3} \beta_{xe} S_{xs}(x_j)^e + \sum_{d=0}^{3} \beta_{ad} S_{as}(a_l)^d + \beta_c S_c + \sum_{i=1}^{2} \beta_{x(i+4)} S_c (x_j)^i + \sum_{h=1}^{2} \beta_{a(h+4)} S_c (a_l)^h \qquad (20)$$

$$+ \sum_{g=1}^{2} \sum_{f=1}^{2} \beta_{(g+3)(f+3)} S_c (x_j)^g (a_l)^f$$

Finally, a fourth model, (21), was fit that allowed all of the 22 parameters (excluding the subgroup frequencies parameter) in (18) to differ by subgroup (45 parameters):

$$\log_e(p_{jlc}) = \alpha + \beta_x S_{xo} + \beta_a S_{a0} + \sum_{i=1}^{4} \beta_{xi}(x_j)^i + \sum_{h=1}^{4} \beta_{ah}(a_l)^h + \sum_{g=1}^{2} \sum_{f=1}^{2} \beta_{gf}(x_j)^g (a_l)^f$$

$$+ \sum_{e=0}^{3} \beta_{xe} S_{xs}(x_j)^e + \sum_{d=0}^{3} \beta_{ad} S_{as}(a_l)^d + \beta_c S_c + \sum_{i=1}^{4} \beta_{x(i+4)} S_c (x_j)^i + \sum_{h=1}^{4} \beta_{a(h+4)} S_c (a_l)^h \qquad (21)$$

$$+ \sum_{g=1}^{2} \sum_{f=1}^{2} \beta_{(g+3)(f+3)} S_c (x_j)^g (a_l)^f + \beta_{x0c} S_c S_{xo} + \beta_{a0c} S_c S_{a0} + \sum_{e=0}^{3} \beta_{xec} S_c S_{xs}(x_j)^e$$

$$+ \sum_{d=0}^{3} \beta_{adc} S_c S_{as}(a_l)^d$$

This fourth model allowed for a consideration of the fit of a model that regarded the subgroups as completely independent with respect to their univariate and bivariate distributions and also with respect to their zero-score frequencies and the distributions of their teeth (which address the omitting pattern differences of the examinees). Models (18) through (21) allow for direct evaluations of how the *X* and *A* distributions differ. The SAS code that imports the trivariate data and defines the score functions needed for the four models is shown in Appendix M. Appendix N shows how the three models are fit using the SAS macro and also presents the overall model statistics from the four models.

Comparisons of the fit statistics across the four models suggest that the subgroups differ much more in terms of their marginal *X* and *A* distributions than their bivariate *XA* distributions. When the means and variances of *X* and *A* are allowed to differ by subgroup in (19), the likelihood ratio chi-square measure improves relative to (18) by 114.73 with these additional 4 degrees of freedom (significant beyond $\underline{p} < .0001$). This is almost all of the improvement in the

likelihood ratio chi-square statistic that could be obtained by allowing all of the parameters to vary by subgroup in (21) (132.10). The AIC and CAIC measures also decrease, suggesting that the additional parameters are not overfitting the data. To gain further insight into how the marginal distributions of *X* and *A* differ, Appendices O and P plot them for the two subgroups and also for the total group based on (19). The differences are visible, showing that the English examinee group has higher means and smaller variances on *X* and *A*.

When the subgroups are allowed to differ in terms of their bivariate distribution, the improvement in model fit is not as dramatic. When considering (20) relative to the simpler model in (19), the likelihood ratio chi-square is reduced by 7.66 on 4 degrees of freedom ($\underline{p}$ > .10). The AIC and CAIC statistics actually increase, which suggests that allowing the bivariate *XA* moments to vary by subgroup could be overfitting the very sparse bivariate data. To gain further insight into the subgroup differences in the conditional distribution of *X* given *A*, the conditional means and standard deviations of *X* for each score of *A* are computed and plotted in Appendix Q, based on (20). These conditional statistics are almost exactly the same for the two groups, with the exception at the anchor score of zero, where the English as first language examinees are shown to have larger means and standard deviations on *X*.

Research Implications of This Study

The primary objective of this paper was to make a flexible and powerful SAS IML macro available for loglinear smoothing. The demonstrations provided in this paper show that many different kinds of smoothings can be performed with the macro, including univariate, bivariate, and trivariate problems.

A second objective of this paper is to promote the use of loglinear modeling for comparing distributions and models, in addition to the smoothing of score frequencies. These comparisons of distributions are directly relevant to the consideration of different equating models. When loglinear models are used along with the kernel equating framework (von Davier et al., 2004), alternative tests of the same question can be considered. For example, the comparison of subgroups' distributions that was featured in this paper's trivariate section has an analogous significance test of group-equating functions at test-score levels within kernel equating (through using the discussed trivariate loglinear smoothing results as input and the standard error of equating differences). Comparisons of distribution-level and equated-score-level results are only now being performed (Moses, Yang, & Wilson, 2005). The approaches

have implications for showing the additional noise that equating functions add to score distributions, for assessing the practical implications of differences between equating functions, and for directly testing the distributional assumptions of specific equating approaches (linear vs. curvilinear and population invariance assumptions).

References

Agresti, A. (2002). *Categorical data analysis* (2nd ed.). New York: Wiley.

Akaike, H. (1981). Likelihood of a model and information criteria. *Journal of Econometrics*, *16*, 3–14.

Akaike, H. (1987). Factor analysis and AIC. *Psychometrika*, *52*, 317–332.

Bishop, Y. M. M., Feinberg, S. E., & Holland, P. W. (1975). *Discrete multivariate analysis: Theory and practice.* Cambridge, MA: MIT Press.

Bock, R. D., & Yates, G. (1973). *MULTIQUAL: Log-linear analysis of nominal or ordinal qualitative data by the method of maximum likelihood*. Chicago: International Education Services.

Bozdogan, H. (1987). Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika*, *52*, 345–370.

von Davier, A. A., Holland, P. W., & Thayer, D. T. (2004). *The kernel method of test equating*. New York: Springer-Verlag.

Dongarra, J. J., Bunch, J. R., Moler, C. B., & Stewart, G. W. (1979). *LINPACK user's guide*. Philadelphia, PA: SIAM.

Freeman, M. F., & Tukey, J. W. (1950). Transformations related to the angular and the square root. *Annals of Mathematical Statistics*, *21*, 607–611.

Gilula, Z., & Haberman, S. J. (1994). Conditional log-linear models for analyzing categorical panel data. *Journal of the American Statistical Association*, *89*, 645–656.

Haberman, S. J. (1973). The analysis of residuals in cross-classification tables. *Biometrics*, *29*, 205–220.

Haberman, S. J. (1974a). Log-linear models for frequency tables with ordered classifications. *Biometrics*, *30*, 589–600.

Haberman, S. J. (1974b). *The analysis of frequency data*. Chicago: University of Chicago Press.

Haberman, S. J. (1977). Log-linear models and frequency tables with small expected cell counts. *The Annals of Statistics*, *5*(6), 1148–1169.

Haberman, S. J. (1978). *Analysis of qualitative data: Vol. 1. Introductory topics.* New York: Academic Press.

Haberman, S. J. (1979). *Analysis of qualitative data: Vol. 2. New developments.* New York: Academic Press.

Hanson, B. A. (1991). A comparison of bivariate smoothing methods in common-item equipercentile equating. *Applied Psychological Measurement*, *15*(4), 391–408.

Hanson, B. A. (1996). Testing for differences in test score distributions using log-linear models. *Applied Measurement in Education*, *9*, 305–321.

Holland, P. W., & Thayer, D. T. (1987). *Notes on the use of log-linear models for fitting discrete probability distributions* (PSR Technical Rep. No. 87–79; ETS RR-87-31). Princeton, NJ: ETS.

Holland, P. W., & Thayer, D. T. (1989). *The kernel method of equating score distributions* (PSR Technical Rep. 89–84; ETS RR-89-07). Princeton, NJ: ETS.

Holland, P. W., & Thayer, D. T. (2000). Univariate and bivariate loglinear models for discrete test score distributions. *Journal of Educational and Behavioral Statistics*, *25*, 133–183.

Kolen, M. J. (1991). Smoothing methods for estimating test score distributions. *Journal of Educational Measurement*, *28*(3), 257–282.

Kolen, M. J., & Brennan, R. L. (1995). *Test equating, methods and practices.* New York: Springer-Verlag.

Liou, M., Cheng, P. E., & Li, M. (2001). Estimating comparable scores using surrogate variables. *Applied Psychological Measurement*, *25*(2), 197–207.

Livingston, S. (1993). Small-sample equatings with log-linear smoothing. *Journal of Educational Measurement*, *30*, 23–39.

Moses, T., & von Davier, A. A. (2004). Using PROC GENMOD for loglinear smoothing. In W. Stinson & E. Westerlund (Co-chairs), *NorthEast SAS Users Group, Inc. seventeenth annual conference proceedings*. Retrieved February 20, 2006, from http://www.nesug.org/html/Proceedings/nesug04.pdf

Moses, T., von Davier, A. A., & Casabianca, J. (2004). *Loglinear smoothing: An alternative numerical approach using SAS* (ETS RR-04-27). Princeton, NJ: ETS.

Moses, T., Yang, W., & Wilson, C. (2005, April). *Using kernel equating to check the statistical equivalence of nearly identical test editions*. Paper presented at the annual meeting of National Council on Measurement in Education Conference, Montreal.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical recipes in C: The art of scientific computing* (2nd ed.). New York: Cambridge University Press.

Read, T., & Cressie, N. (1988). *Goodness-of-fit statistics for discrete multivariate data.* New York: Springer.

Rosenbaum, P. R., & Thayer, D. (1987). Smoothing the joint and marginal distributions of scored two-way contingency tables in test equating. *British Journal of Mathematical and Statistical Psychology*, *40*, 43–49.

SAS Institute. (2002a). SAS/STAT software: The GENMOD procedure, Version 9 [Computer software]. Cary, NC: SAS Institute.

SAS Institute. (2002b). *SAS/IML 9 user's guide*. Carey, NC: SAS Institute.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.

Skaggs, G. (2004, April). *Passing score stability when equating with very small samples*. Paper presented at the annual meeting of the American Educational Research Association, San Diego CA.

The Steps for Implementing the SAS Macro for Loglinear Smoothing

1. Enter the dataset into SAS. We assume that most of the time this initial dataset lists the test score(s) for each individual examinee.

2. Convert the dataset from 1) into a frequency dataset that lists the counts for each observed test score or observed combination of test scores.

3. Because not every test score or combination of test scores is always attained in an observed sample, an additional, empty frequency table should be created that includes all possible scores or score combinations. For bivariate data of total tests and internal anchors, this step should exclude structural zeros, the impossible score combinations (e.g. if scored as "number rights," the anchor score cannot be greater than the total test score).

4. Merge the datasets in 2) and 3) and convert any missing counts to zero,

5. Define all of the score functions needed to preserve the desired moments in the smoothing model.

6. Fit the model with the SAS macro. This step will require only two lines of code, which are the following:

```
%include '[filelocation]\loglinmacro.sas';
%loglin(libname=,data=,count=,scoref=,output=);
```

The first line calls in the file of the macro from an accessible drive location and only needs to be run once at the beginning of the SAS session. The second line runs the macro, where library for the dataset can be optionally named after `libname=`, and a dataset within the library that contains the score frequencies is listed after `data=`, the observed counts to be smoothed are listed after `count=`, and the score functions that correspond to the moments that are to be preserved are listed after `scoref=`. Several measures of model fit will be printed and saved to an outputted dataset named `fitoutput`, where `output` is specified after `output=`. A second outputted dataset, named after `output=`, will include the test score values and score functions, the observed counts, the smoothed counts, the smoothed probabilities, the Freeman-Tukey residuals, and the C-matrix.

Univariate Example: Entering the Frequency Table and Defining the Score Functions

```
data llin;
input count y;
cards;
0 0
4 1
11 2
16 3
18 4
34 5
63 6
89 7
87 8
129 9
124 10
154 11
125 12
131 13
109 14
98 15
89 16
66 17
54 18
37 19
17 20
;

data llin;set llin;
y2=y**2;
y3=y**3;
run;
```

Appendix C

Univariate Example: Running the Smoothing Macro and Its Results

```
%include 'H:\ApSplreq\tpm\loglinearsmoothing\aera\loglinmacro.sas';
%loglin(data=llin,count=count,scoref=y y2 y3,output=outresults);
```

                              RESULTS

                  The following score functions were included:

                              SCOREFUNCTIONS

                          Y   Y2   Y3

                                  ITER

            The solution converged in   6 iterations.

                              MODELFIT

                Likelihood Ratio Chi-square   20.24
                Pearson Chi-square   18.35
                Freeman-Tukey Chi-square   20.09
                AIC   28.24
                CAIC   53.37
                Degrees of Freedom   17.00

                            BETAESTIMATES
                         Betas   StdErrors

                 Y   0.8389425393320 0.0917352777665
                 Y2 -.0453897017972 0.0087387126508
                 Y3 0.0005035366392 0.0002577475833

                              ALPHA

                           -6.748485

Appendix D

Univariate Example: The Dataset Outputted From Running the SAS Smoothing Macro

```
proc print data=outresults noobs;run;
```

count	y	y2	y3	smoothedcounts	smoothedprobs	ftresiduals	cY	cY2	cY3
0	0	0	0	1.706	0.001173	-1.79729	-.000090589	-.000184074	-.000285245
4	1	1	1	3.775	0.002594	0.22371	-.000183128	-.000330256	-.000429494
11	2	4	8	7.649	0.005257	1.15954	-.000336071	-.000527477	-.000544529
16	3	9	27	14.242	0.009788	0.50944	-.000560484	-.000744582	-.000545922
18	4	16	64	24.436	0.016794	-1.33537	-.000849736	-.000915463	-.000345930
34	5	25	125	38.752	0.026634	-0.74332	-.001170120	-.000949703	0.000093103
63	6	36	216	56.978	0.039160	0.80741	-.001459501	-.000764005	0.000707042
89	7	49	343	77.905	0.053543	1.23973	-.001638764	-.000324827	0.001318586
87	8	64	512	99.354	0.068284	-1.25213	-.001634929	0.000318990	0.001690498
129	9	81	729	118.543	0.081473	0.96117	-.001407797	0.001035082	0.001628788
124	10	100	1000	132.724	0.091219	-0.74696	-.000968368	0.001644944	0.001083547
154	11	121	1331	139.868	0.096129	1.18530	-.000379933	0.001985664	0.000188722
125	12	144	1728	139.154	0.095638	-1.20858	0.000259296	0.001967721	-.000784962
131	13	169	2197	131.097	0.090101	0.01328	0.000844677	0.001602719	-.001539598
109	14	196	2744	117.307	0.080623	-0.75635	0.001293062	0.000991380	-.001864801
98	15	225	3375	100.000	0.068728	-0.17559	0.001560260	0.000281782	-.001703649
89	16	256	4096	81.457	0.055984	0.84239	0.001644003	-.000380778	-.001148582
66	17	289	4913	63.596	0.043709	0.32865	0.001574774	-.000892001	-.000383198
54	18	324	5832	47.732	0.032806	0.91084	0.001400550	-.001204595	0.000393841
37	19	361	6859	34.545	0.023742	0.44973	0.001171820	-.001323284	0.001029168
17	20	400	8000	24.180	0.016619	-1.51966	0.000930977	-.001287237	0.001442615

Univariate Example: Plotting the Observed and Smoothed Counts

```
axis1 label=(angle=-90 rotate=90 'Frequency' font='Times New Roman'
height=2in);
axis2 order=(0 to 20 by 1) label=('Score' height=100in );
Legend1 label=(height=1 position=top justify=center '')
       value=('Observed' 'Smoothed') position=(top center);
symbol1 color=green interpol=none width=1 value=triangle height=1;
symbol2 color=blue interpol=none width=1 value=circle height=1;
proc gplot data=outresults;
plot count*y smoothedcounts*y / overlay vaxis=axis1 haxis=axis2
legend=legend1;
title 'Figure 1. Observed and Smoothed Frequencies';
run;quit;
```


Figure 1. Observed and Smoothed Frequencies
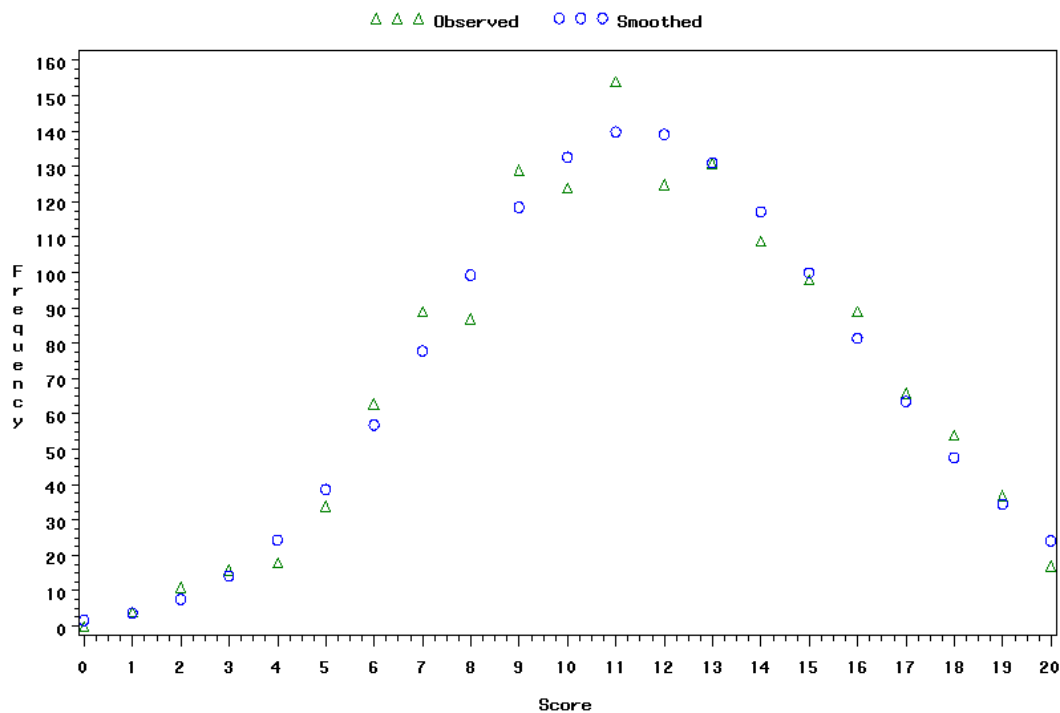
Appendix F

Univariate Example: Verifying That the Smoothed Distribution Preserves the First

Three Moments of the Observed Distribution

```
/*These commands create large datasets based on the actual and fitted frequencies of X.
Then i compare the moments of these datasets.*/
data yobserved;set outresults;
do i=1 to 1000*count;
output;
end;
drop i;
data ysmoothed;set outresults;
do i=1 to 1000*smoothedcounts;
output;
end;
drop i;
data yobssmooth;merge yobserved(rename=y=yobserved) ysmoothed(rename=y=ysmoothed);
proc means data=yobssmooth mean std skew kurt;
var yobserved ysmoothed;
title 'Moments based on the actual and smoothed frequencies of Y.';
run;
```

```
                    Moments based on the actual and smoothed frequencies of Y.
                                    The MEANS Procedure
            Variable              Mean            Std Dev         Skewness          Kurtosis
            ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
            yobserved        11.5931271          3.9342663       -0.0626866        -0.4988359
            ysmoothed        11.5931404          3.9342451       -0.0626767        -0.4277949
            ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
```

Appendix G

Bivariate Example: The Descriptive Statistics of the Marginal Distributions of X and A

```
proc means data=neatxa mean std skew kurt min max;
var x a;
title 'Table 1: Descriptive statistics of the two tests, X and A (N=10,634).';
run;
```

Table 1: Descriptive statistics of the two tests, X and A (N=10,634).

The MEANS Procedure

Variable	Mean	Std Dev	Skewness	Kurtosis	Minimum	Maximum
X	39.2656573	17.1952746	-0.1014811	-0.7825456	0	78.0000000
A	17.0539778	8.3332670	-0.0096494	-0.8534236	0	35.0000000

Appendix G

Bivariate Example: The Descriptive Statistics of the Marginal Distributions of X and A

```
proc means data=neatxa mean std skew kurt min max;
var x a;
title 'Table 1: Descriptive statistics of the two tests, X and A (N=10,634).';
run;
```

Table 1: Descriptive statistics of the two tests, X and A (N=10,634).

The MEANS Procedure

Variable	Mean	Std Dev	Skewness	Kurtosis	Minimum	Maximum
X	39.2656573	17.1952746	-0.1014811	-0.7825456	0	78.0000000
A	17.0539778	8.3332670	-0.0096494	-0.8534236	0	35.0000000

Bivariate Example: Implementing Steps 1-5 of the SAS Routine

```
/*Step 1: Inputting the dataset of Individual Observations*/
data neatxa; infile 'H:\bivariate.dat';
input @9 X 2.
      @11 A 2.;

/*Step 2: Obtaining the Observed Bivariate Frequencies.*/
proc freq data=neatxa noprint;tables X*A / out=neatxa;run;

/*Step 3: Defining all possible score combinations.*/
data outxa;do x=0 to 78 by 1;do a=0 to 35 by 1;output;end;end;

/*Step 4: Merging 2 and 3*/
data neat;merge neatxa outxa;by x a; if count=. then count=0;

/*Step 5: Defining the Score Functions for the Desired Models.*/
data neat;set neat;
/*The score functions for X.*/
x2=x**2;
x3=x**3;
x4=x**4;
/*The score functions for A.*/
a2=a**2;
a3=a**3;
a4=a**4;
/*The X-A Cross-Moments.*/
ax=a*x;
a2x=a*a*x;
ax2=a*x*x;
a2x2=a*a*x*x;
/*The lumps at zero for X and A.*/
if x=0 then IX0=1;else IX0=0;
if a=0 then IA0=1;else IA0=0;

/*The teeth of X.*/
do i=5 to 75 by 5;if X=i then IXS=1;end;
if IXS=. then IXS=0;

/*The teeth of A.*/
do j=2 to 32 by 5;if a=j then IAS=1;end;
if IAS=. then IAS=0;
```

```
/*The univariate moments for the teeth of X and A.*/
IXSx=IXS*x;
IXSx2=IXS*x2;
IXSx3=IXS*x3;
IASa=IAS*a;
IASa2=IAS*a2;
IASa3=IAS*a3;
```

Appendix I

Bivariate Example: The SAS Code and Output for Fitting (16) and (17)

```
/*Model 16*/
%loglin(data=neat,count=count,scoref=X X2 X3 X4 A A2 A3 A4 IX0 IA0 IXS IXSX
IXSX2 IXSX3 IAS IASA IASA2 IASA3 ax ax2 a2x a2x2,output=model16);


/*Model 17*/
%loglin(data=neat,count=count,scoref=X X2 X3 X4 A A2 A3 A4 IX0 IA0 IXS IXSX
IXSX2 IXSX3 IAS IASA IASA2 IASA3 ax,output=model17);

data results;merge fitmodel16 fitmodel17;
proc print data=results noobs;run;
```

The SAS System

STATS	MODEL16	MODEL17
Likelihood Ratio Chi-square	1966.93	2566.90
Pearson Chi-square	6466.90	2.4227E14
Freeman-Tukey Chi-square	1632.96	1780.86
AIC	2012.93	2606.90
CAIC	2203.18	2772.33
Degrees of Freedom	2821.00	2824.00

Bivariate Example: Plotting the Observed and Smoothed Counts of X

```
proc means data=model16 noprint;var count smoothedcounts;class x;output
out=xgraph sum=observed smoothed;run;
data xgraph;set xgraph;if x=. then delete;

axis1 order=(0 to 250 by 50) label=(angle=-90 rotate=90 'Frequency'
font='Times New Roman' height=2in);
axis2 order=(0 to 78 by 5) label=('Score' height=100in );
Legend1 label=(height=1 position=top justify=center '')
      value=('Observed' 'Smoothed') position=(top center);
proc gplot data=xgraph;
plot observed*x='plus' smoothed*x='square' / overlay vaxis=axis1 haxis=axis2
legend=legend1;
title 'Figure 2. Observed and Smoothed Frequencies';
run;quit;
```


Figure 2. Observed and Smoothed Frequencies of X
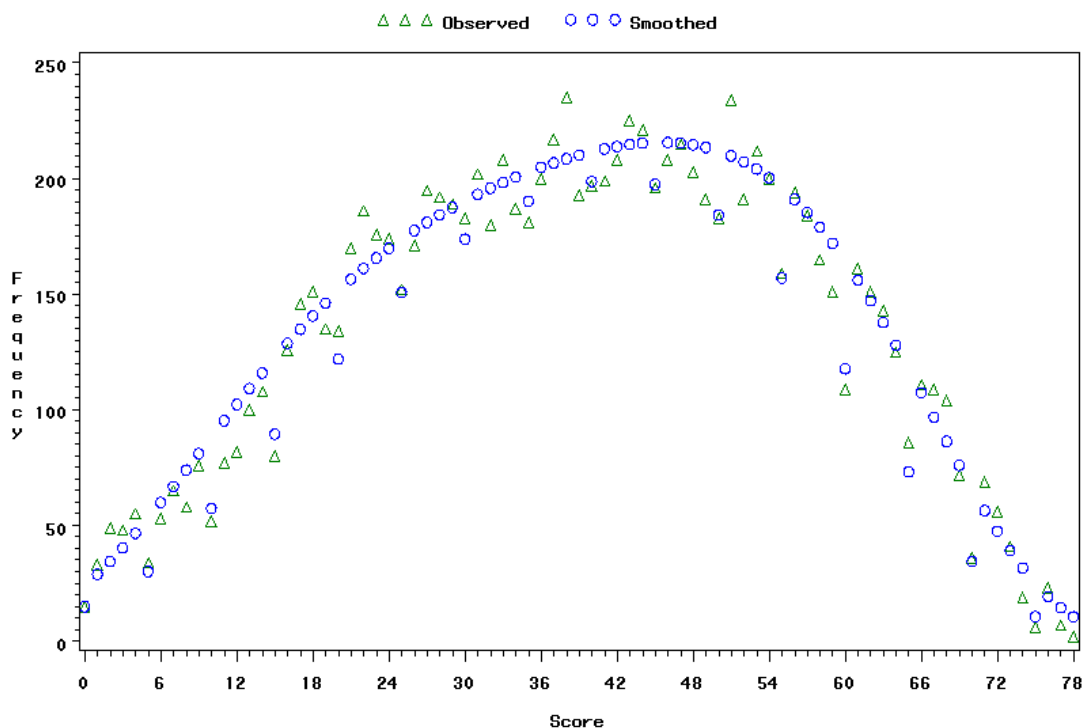
Appendix K

Bivariate Example: Plotting the Observed and Smoothed Counts of A

```
proc means data=model16 noprint;var count smoothedcounts;class a;output
out=agraph sum=observed smoothed;run;
data agraph;set agraph;if a=. then delete;

axis1 order=(0 to 500 by 100) label=(angle=-90 rotate=90 'Frequency'
font='Times New Roman' height=2in);
axis2 order=(0 to 35 by 5) label=('Score' height=100in );
Legend1 label=(height=1 position=top justify=center '')
     value=('Observed' 'Smoothed') position=(top center);
proc gplot data=agraph;
plot observed*a='plus' smoothed*a='square' / overlay vaxis=axis1 haxis=axis2
legend=legend1;
title 'Figure 3. Observed and Smoothed Frequencies';
run;quit;
```


Figure 3. Observed and Smoothed Frequencies of A
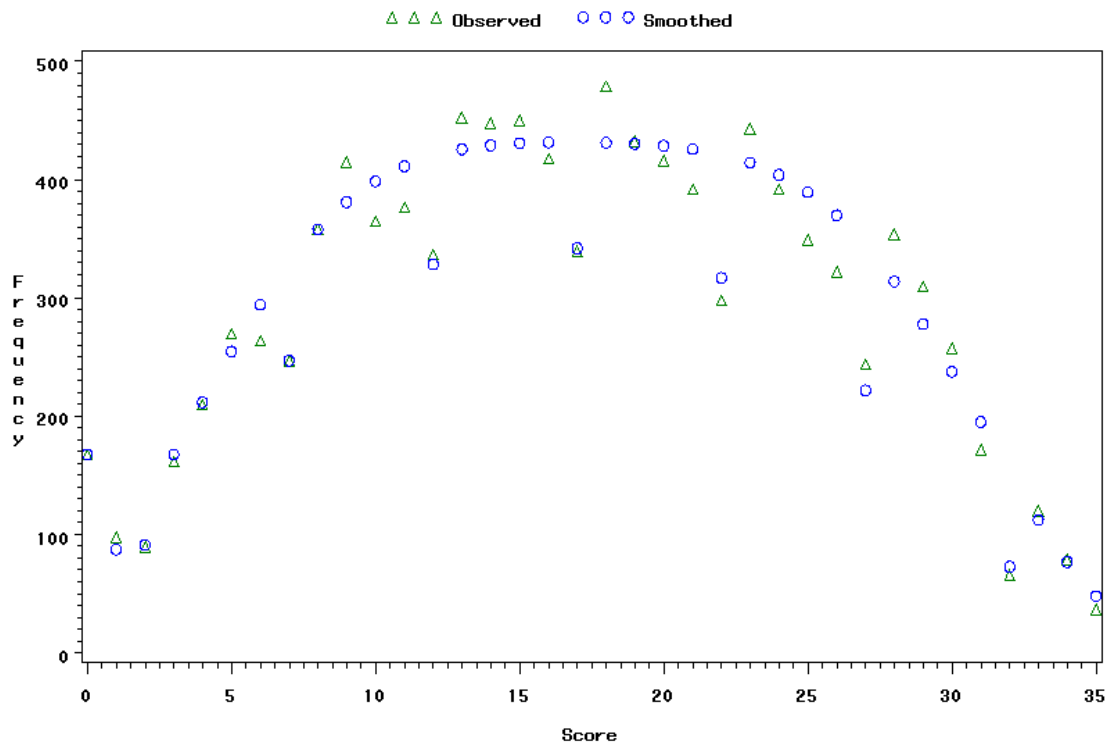
Appendix L

Trivariate Example: Descriptive Statistics of the Subgroups' Marginal Distributions on X and A (Step 1)

```
/*Step 1: Inputting the dataset of Individual Observations*/
data neatxa; infile 'H:\trivariate.dat';
input @7 efl $1.
      @9 X 2.
      @11 A 2.;

data neatxa;set neatxa;
length English $9.;
if x<0 then x=0;
if a<0 then a=0;
if efl='A' then do;ie=1; English='First';end;
else do;ie=0; English='Not First';end;

proc means data=neatxa mean std skew kurt min max;
var x a;
class English;
title 'Table 2. Comparing the Distributions of X and A for the Subgroups.';
run;
```

```
             Table 2. Comparing the Distributions of X and A for the Subgroups.
                                   The MEANS Procedure
  English      N Obs    Variable          Mean         Std Dev       Minimum        Maximum
  ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
  First         6867    X            40.2018349      16.7402678             0     77.0000000
                        A            17.5545362       8.1164723             0     35.0000000
  Not First     3767    X            37.5590656      17.8716481             0     78.0000000
                        A            16.1414919       8.6414012             0     35.0000000
  ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ
```

Trivariate Example: The Data Processing Steps (2-5)

```
data neatxa1;set neatxa;if ie=1;run;
data neatxa0;set neatxa;if ie=0;run;

/*Step 2: Obtaining the Observed Bivariate Frequencies.*/
proc freq data=neatxa1 noprint;tables X*A / out=neatxa1;run;
proc freq data=neatxa0 noprint;tables X*A / out=neatxa0;run;

/*Step 3: Defining all possible score combinations.*/
data outxa;do x=0 to 78 by 1;do a=0 to 35 by 1;output;end;end;

/*Step 4: Merging 2 and 3*/
data neat0;merge neatxa0 outxa;by x a; if count=. then count=0;ie=0;
data neat1;merge neatxa1 outxa;by x a; if count=. then count=0;ie=1;

data neat;set neat0 neat1;

/*Step 5: Defining the Score Functions for the Desired Models.*/
data neat;set neat;
/*The score functions for X.*/
x2=x**2;
x3=x**3;
x4=x**4;
/*The score functions for A.*/
a2=a**2;
a3=a**3;
a4=a**4;
/*The X-A Cross-Moments.*/
ax=a*x;
a2x=a*a*x;
ax2=a*x*x;
a2x2=a*a*x*x;
/*The lumps at zero for X and A.*/
if x=0 then IX0=1;else IX0=0;
if a=0 then IA0=1;else IA0=0;

/*The teeth of X.*/
do i=5 to 75 by 5;if X=i then IXS=1;end;

/*The teeth of A.*/
do j=2 to 32 by 5;if a=j then IAS=1;end;
if IXS=. then IXS=0;
```

```
if IAS=. then IAS=0;

/*The univariate moments for the teeth of X and A.*/
IXSx=IXS*x;
IXSx2=IXS*x2;
IXSx3=IXS*x3;
IASa=IAS*a;
IASa2=IAS*a2;
IASa3=IAS*a3;
run;

/*Step 5, Subgroups: Defining the Subgroup Functions for the Desired Models.*/
data neat;set neat;
/*The score functions for X.*/
iex1=x*ie;
iex2=x2*ie;
iex3=x3*ie;
iex4=x4*ie;
/*The score functions for A.*/
iea1=a*ie;
iea2=a2*ie;
iea3=a3*ie;
iea4=a4*ie;
/*The X-A Cross-Moments.*/
ieax=ie*a*x;
iea2x=ie*a*a*x;
ieax2=ie*a*x*x;
iea2x2=ie*a*a*x*x;
/*The lumps at zero for X and A.*/
if x=0 then IX0=1;else IX0=0;
if a=0 then IA0=1;else IA0=0;

ieIX0=ie*IX0;
ieIA0=ie*IA0;

/*The teeth of X.*/
do i=5 to 75 by 5;if X=i then IXS=1;end;

/*The teeth of A.*/
do j=2 to 32 by 5;if a=j then IAS=1;end;

if IXS=. then IXS=0;
if IAS=. then IAS=0;
```

```
ieixs=ie*ixs;
ieias=ie*ias;

/*The univariate moments for the teeth of X and A.*/
IXSx=IXS*x;
IXSx2=IXS*x2;
IXSx3=IXS*x3;
IASa=IAS*a;
IASa2=IAS*a2;
IASa3=IAS*a3;

ieIXSx=ie*IXS*x;
ieIXSx2=ie*IXS*x2;
ieIXSx3=ie*IXS*x3;
ieIASa=ie*IAS*a;
ieIASa2=ie*IAS*a2;
ieIASa3=ie*IAS*a3;
run;
```

Appendix N

Trivariate Example: Fitting the Four Models and Outputting Their Results (Step 6)

```
/* Model 18 */
%include 'H:\ApSplreq\tpm\loglinearsmoothing\aera\loglinmacro.sas';
%loglin(data=neat,count=count,scoref=X X2 X3 X4 A A2 A3 A4 IX0 IA0 IXS IXSX IXSX2 IXSX3 IAS IASA IASA2 IASA3
ax ax2 a2x a2x2 ie,output=model18);

/* Model 19 */
%loglin(data=neat,count=count,scoref=X X2 X3 X4 A A2 A3 A4 IX0 IA0 IXS IXSX IXSX2 IXSX3 IAS IASA IASA2 IASA3
ax ax2 a2x a2x2 ie ieX1 ieX2 ieA1 ieA2,output=model19);

/* Model 20 */
%loglin(data=neat,count=count,scoref=X X2 X3 X4 A A2 A3 A4 IX0 IA0 IXS IXSX IXSX2 IXSX3 IAS IASA IASA2 IASA3
ax ax2 a2x a2x2 ie ieX1 ieX2 ieA1 ieA2 ieax ieax2 iea2x iea2x2,output=model20);

/* Model 21 */
%loglin(data=neat,count=count,scoref=X X2 X3 X4 A A2 A3 A4 IX0 IA0 IXS IXSX IXSX2 IXSX3 IAS IASA IASA2 IASA3
ax ax2 a2x a2x2 ie ieX1 ieX2 ieX3 ieX4 ieA1 ieA2 ieA3 ieA4 ieIX0 ieIA0 ieIXS ieIXSX ieIXSX2 ieIXSX3 ieIAS
ieIASA ieIASA2 ieIASA3 ieax ieax2 iea2x iea2x2,output=model21);

data results;merge fitmodel18 fitmodel19 fitmodel20 fitmodel21;
proc print data=results noobs;run;
```

STATS	MODEL18	MODEL19	MODEL20	MODEL21
Likelihood Ratio Chi-square	3624.05	3514.30	3502.36	3491.95
Pearson Chi-square	12372.13	11635.84	11577.26	11945.93
Freeman-Tukey Chi-square	2932.99	2826.82	2812.39	2802.62
AIC	3672.05	3570.30	3566.36	3583.95
CAIC	3870.57	3801.91	3831.06	3964.45
Degrees of Freedom	5664.00	5660.00	5656.00	5642.00

Appendix O

Trivariate Example: Plotting the Marginal Distributions of X

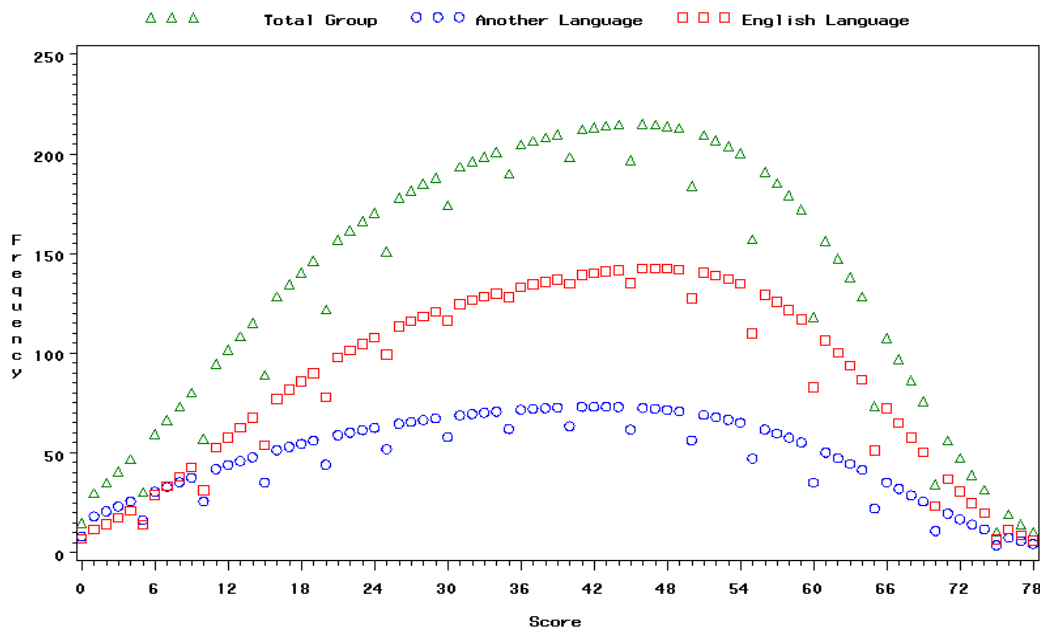## for the Total Group and the Subgroups

```
proc means data=model21 noprint;var smoothedcounts;class x ie;output
out=xgraph sum=smoothed;run;

data xgraph;set xgraph;if x=. then delete;

proc sort data=xgraph;by ie;run;

axis1 order=(0 to 250 by 50) label=(angle=-90 rotate=90 'Frequency'
font='Times New Roman' height=2in);
axis2 order=(0 to 78 by 6) label=('Score' height=100in );
Legend1 label=(height=1 position=top justify=center '')
      value=('Total Group' 'Another Language' 'English Language')
position=(top center);
symbol1 color=green interpol=none width=1 value=triangle height=1;
symbol2 color=blue interpol=none width=1 value=circle height=1;
symbol3 color=red interpol=none width=1 value=square height=1;
proc gplot data=xgraph;
plot smoothed*x=ie / vaxis=axis1 haxis=axis2 legend=legend1;
title 'Figure 4. Comparing Subgroup and Total Smoothed Frequencies on X';
run;quit;
```


Figure 4. Comparing Subgroup and Total Smoothed Frequencies on X

Appendix P

Trivariate Example: Plotting the Marginal Distributions of A

**for the Total Group and the Subgroups**

```
proc means data=model21 noprint;var smoothedcounts;class a ie;output
out=agraph sum=smoothed;run;

data agraph;set agraph;if a=. then delete;

proc sort data=agraph;by ie;run;

axis1 order=(0 to 500 by 100) label=(angle=-90 rotate=90 'Frequency'
font='Times New Roman' height=2in);
axis2 order=(0 to 35 by 5) label=('Score' height=100in );
Legend1 label=(height=1 position=top justify=center '')
        value=('Total Group' 'Another Language' 'English Language')
position=(top center);
symbol1 color=green interpol=none width=1 value=triangle height=1;
symbol2 color=blue interpol=none width=1 value=circle height=1;
symbol3 color=red interpol=none width=1 value=square height=1;
proc gplot data=agraph;
plot smoothed*a=ie / vaxis=axis1 haxis=axis2 legend=legend1;
title 'Figure 5. Comparing Subgroup and Total Smoothed Frequencies on A';
run;quit;
```


Figure 5. Comparing Subgroup and Total Smoothed Frequencies on A
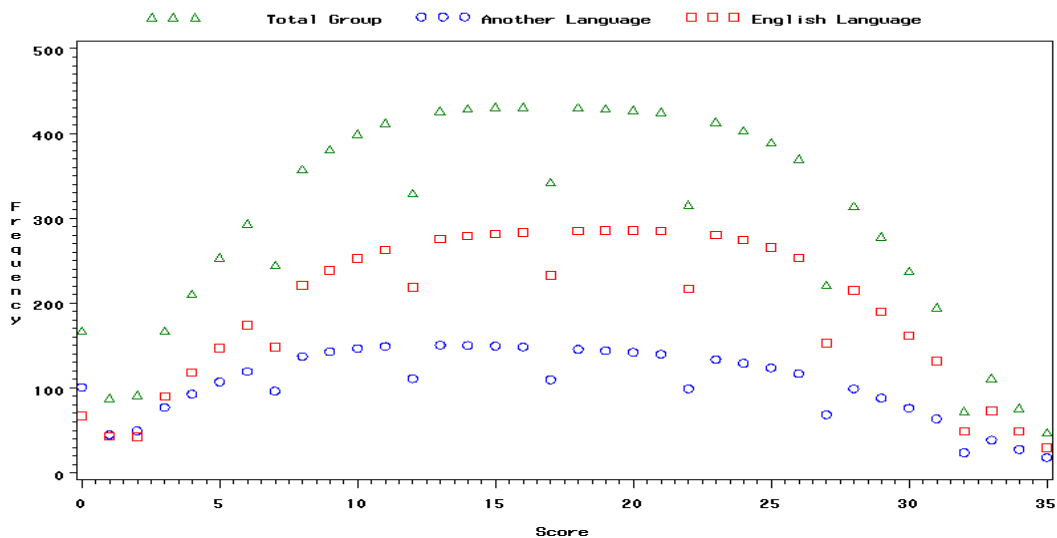
Appendix Q

Trivariate Example: Plotting the Conditional Moments of X Given A for the Subgroups

```
/*Comparing Subgroup Differences on the Conditional Smoothed Moments.*/
/*Programming the Moments*/
data subgroups_pse0;set model20;if ie=0;run;
data subgroups_pse1;set model20;if ie=1;run;

proc means data=subgroups_pse0 noprint;var smoothedcounts;class a;output
out=total0 sum=total;run;
proc means data=subgroups_pse1 noprint;var smoothedcounts;class a;output
out=total1 sum=total;run;

data total0;set total0;if a=. then delete;
data total1;set total1;if a=. then delete;

proc sort data=subgroups_pse0;by a;run;
proc sort data=subgroups_pse1;by a;run;

data subgroups_pse0;merge subgroups_pse0 total0;by a;
sp=smoothedcounts/total;
xsp=x*sp;
run;

data subgroups_pse1;merge subgroups_pse1 total1;by a;
sp=smoothedcounts/total;
xsp=x*sp;
run;

proc means data=subgroups_pse0 noprint;var xsp;class a;output out=mean0
sum=xsmoothedmean;run;
proc means data=subgroups_pse1 noprint;var xsp;class a;output out=mean1
sum=xsmoothedmean;run;

data mean0;set mean0;if a=. then delete;
data mean1;set mean1;if a=. then delete;

data subgroups_pse0;merge subgroups_pse0 mean0;by a;
x2sp=(x-xsmoothedmean)**2*sp;
run;

data subgroups_pse1;merge subgroups_pse1 mean1;by a;
x2sp=(x-xsmoothedmean)**2*sp;
run;
```

```
proc means data=subgroups_pse0 noprint;var x2sp;class a;output out=variances0
sum=xsmoothedvar;run;
proc means data=subgroups_pse1 noprint;var x2sp;class a;output out=variances1
sum=xsmoothedvar;run;

data variances0;set variances0;if a=. then delete;
data variances1;set variances1;if a=. then delete;

data subgroups_pse0;merge subgroups_pse0 variances0;by a;
xssd=sqrt(xsmoothedvar);

data subgroups_pse1;merge subgroups_pse1 variances1;by a;
xssd=sqrt(xsmoothedvar);

proc means data=subgroups_pse0 noprint;var xssd;class a;output out=sd0
mean=xsmoothedsd;run;
proc means data=subgroups_pse1 noprint;var xssd;class a;output out=sd1
mean=xsmoothedsd;run;

data sd0;set sd0;if a=. then delete;
data sd1;set sd1;if a=. then delete;

data xgivena0;merge mean0 sd0;by a;ie=0;run;
data xgivena1;merge mean1 sd1;by a;ie=1;run;

data xgivena;set xgivena0 xgivena1;run;

axis1 order=(0 to 80 by 10) label=(angle=-90 rotate=90 'Conditional Means'
font='Times New Roman' height=2in);
axis2 order=(0 to 35 by 5) label=('Score' height=100in );
Legend1 label=(height=1 position=top justify=center '')
      value=('Another Language' 'English Language') position=(top center);
symbol1 color=blue interpol=none width=1 value=circle height=1;
symbol2 color=red interpol=none width=1 value=square height=1;;
proc gplot data=xgivena;
plot xsmoothedmean*a=ie /vaxis=axis1 haxis=axis2 legend=legend1;
title 'Figure 6. Comparing the Subgroup Differences in Smoothed E(X|A).';
run;quit;
```
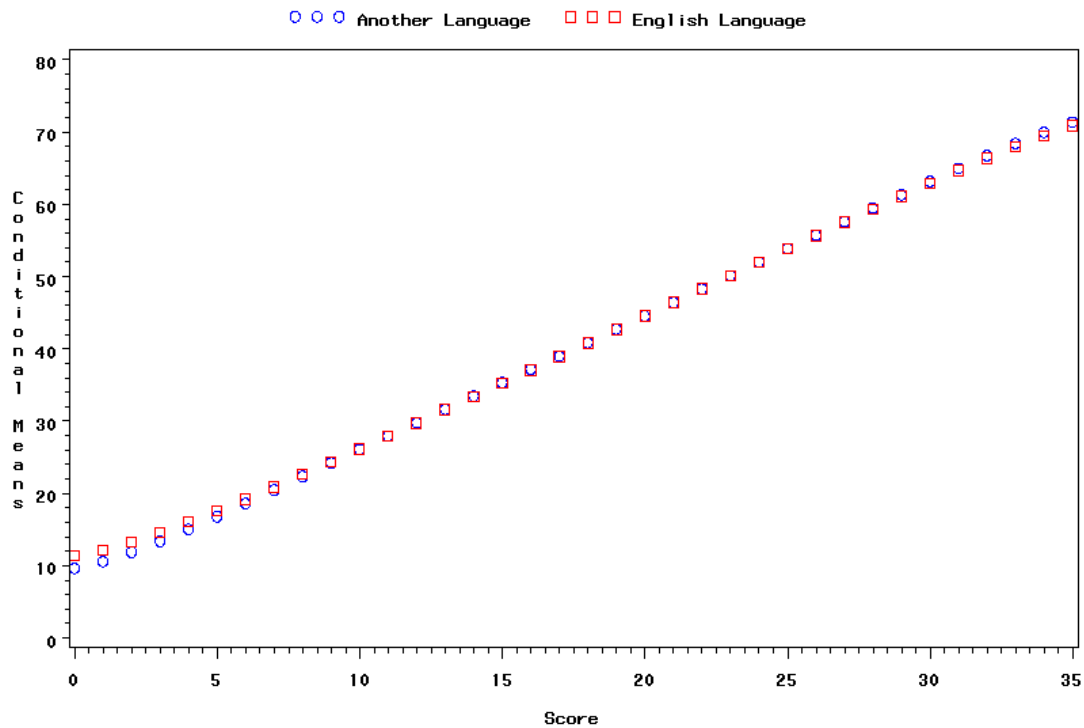
Figure 6. Comparing the Subgroup Differences in Smoothed E(X¦A)


```
axis1 order=(0 to 20 by 5) label=(angle=-90 rotate=90 'Conditional Standard
Deviations' font='Times New Roman' height=2in);
axis2 order=(0 to 35 by 5) label=('Score' height=100in );
Legend1 label=(height=1 position=top justify=center '')
       value=('Another Language' 'English Language') position=(top center);
symbol1 color=blue interpol=none width=1 value=circle height=1;
symbol2 color=red interpol=none width=1 value=square height=1;
proc gplot data=xgivena;
plot xsmoothedsd*a=ie /vaxis=axis1 haxis=axis2 legend=legend1;
title 'Figure 7. Comparing the Subgroup Differences in Smoothed SD(X|A).';
run;quit;
```

Figure 7. Comparing the Subgroup Differences in Smoothed SD(X¦A)