Paper SA06_05
Using PROC GENMOD for Loglinear Smoothing

Tim Moses and Alina A. von Davier, Educational Testing Service, Princeton, NJ

ABSTRACT AND INTRODUCTION

The goal of smoothing is to replace an observed frequency distribution with a distribution that preserves some features of the observed data without the irregularities that are attributable to sampling. The type of smoothing covered in this paper involves the fitting of loglinear, Poisson-based models to discrete distributions. Loglinear smoothing can preserve a variety of different features in observed data with a relatively small number of parameters.

In this paper we use SAS/STAT® PROC GENMOD (SAS, 2002) to demonstrate the smoothing of univariate (one variable) and bivariate (two variables and one variable for separate subgroups) frequency distributions. For univariate distributions, we will produce smoothed distributions that preserve 1) the mean, 2) the mean and variance, 3) the mean, variance and skewness, and finally 4) the mean, variance, skewness and kurtosis in the observed distribution of one variable, *X*. For bivariate distributions, we will produce smoothed distributions that preserve three univariate moments in each of the marginal distributions of two variables, *X* and *Y*, as well as the correlation between *X* and *Y*. Finally, the incorporation of indicator functions is used to model overall and subset-specific features of distributions within the same overall model.

LOGLINEAR SMOOTHING MODELS

Assume we have a discrete random variable *X* with possible values $x_0,\ldots,x_J$, or x_j, with $j=0,\ldots,J$ (the possible values), and a corresponding vector of observed frequencies $n = (n_0,\ldots,n_J)^t$ that sum to the total sample size, *N*. Under multinomial or Poisson distributional assumptions about *n*, the vector of the population probabilities $p = (p_0,\ldots,p_J)^t$ is said to satisfy the following loglinear model:

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

where the $\{p_j\}$ are assumed to be positive and sum to 1, $\mathbf{b_j}$ is a row vector of known constants, $\boldsymbol{\beta}$ is a vector of free parameters, u_j is a known constant that specifies the distribution of the $\{p_j\}$ when the vector $\boldsymbol{\beta}$ is set to zero, and α is a normalizing constant that insures that the probabilities sum to one (Holland & Thayer, 1987; 2000). Throughout this paper $\boldsymbol{u}$ will be set to 0 so that the "null" model will be a uniform distribution where the frequencies for all j score values are equal to *N/J*.

For the modeling of test score distributions, we write the loglinear model as:

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

where p_j is the probability of obtaining test score x_j, the x_j^i are "score functions" of the possible score values of test *X* (e.g., x_j^1, $x_j^2, x_j^3,\ldots, x_j^I$), α is as described above, and the β_i are free parameters to be estimated in the model-fitting process.

Loglinear models like (1) have a very useful "moment-matching" property. The maximum likelihood estimates of $\boldsymbol{\beta}$ will force the estimated probabilities $\{\hat{p}_j\}$ to satisfy the following condition (Holland & Thayer, 1987; 2000):

$$\sum_j x_j^i \hat{p}_j = \sum_j x_j^i (n_j / N) \text{ for i = 1 to } I.$$

This means that the *I* fitted moments in the smoothed distribution will equal the observed moments. For example, when *I*=1 in model (1), **x** is a j-by-1 matrix of one score function (x_j^1) and the resulting loglinear model will preserve the first moment (the mean) of the observed distribution. When *I*=4 in model (1), **x** is a j-by-4 matrix of four score functions (x_j^1, x_j^2, x_j^3, and x_j^4) and the resulting loglinear model will preserve the first moment (mean), the second moment (variance), the third moment (skew) and the fourth moment (kurtosis) of the observed distribution.

The model in (1) can be extended to fit the bivariate distribution of the scores of two tests (call them *X* and *Y*):

$$\log_e(p_{jk}) = \alpha + \sum_{i=1}^{I} \beta_{xi} (x_j)^i + \sum_{h=1}^{H} \beta_{yh} (y_k)^h + \sum_{g=1}^{G} \sum_{f=1}^{F} \beta_{gf} (x_j)^g (y_k)^f \,, \qquad (2)$$

where the p_{jk}'s are the joint score probabilities of the score (x_j, y_k) (score x_j on test *X* and score y_k on test *Y*) written in a single-column vector. The fitting of model (2) produces a 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 ($G \le I,\ F \le H$) in the bivariate *XY* distribution. The cross-moments will model the dependence between tests *X* and *Y*, so that when *F=G=0* the resulting loglinear model will produce a smoothed *XY* distribution where the covariance between *X* and *Y* is not preserved. When *F=G=1* the resulting loglinear model will produce a smoothed *XY* distribution where the covariance between *X* and *Y* will be preserved.

When a distribution can be considered as having distinct subsets, the moments of these subsets can be modeled through the use of indicator (0,1) functions which have the same use as "dummy variables" in ordinary least squares regression. One example of a loglinear smoothing model that uses indicator functions is necessary when there is an extremely large number of zero scores in a test. In this case, the histogram can show an overall distribution with a "lump" at the test score of zero. When an indictor function, $I_S(j)$, is defined as 1 at score *j*=0 and 0 otherwise and incorporated into the following model,

$$\log_e(p_j) = \alpha + \beta_1(x_j)^1 + \beta_2(x_j)^2 + \beta_3 I_s(j), \qquad (3)$$

the frequency at score *j*=0 can be preserved (β_3) along with the distribution's overall mean (β_1) and variance (β_2).

Indicator functions can also be used to model distinct distributions. For example, subgroups may be considered to differ with respect their means, variances, skews, etc. These differences can be evaluated by fitting loglinear smoothing models that use indicator functions to define separate subgroups and including product terms of the indicator functions and score functions that allow the subgroups' moments to differ. The following model preserves the overall mean (β_1) and variance (β_2) of a univariate distribution, the frequencies of the subgroups (β_3), and the subgroup-specific means (β_4) and variances (β_5),

$$\log_e(p_{jS}) = \alpha + \beta_1(x_j)^1 + \beta_2(x_j)^2 + \beta_3 I_s(j) + \beta_4(x_j)^1 I_s(j) + \beta_5(x_j)^2 I_s(j). \qquad (4)$$

The resulting smoothed frequencies are said to be much more stable than the observed frequencies, especially when sample sizes are small (Rosenbaum & Thayer, 1987). These smoothed frequencies are very useful for test equating methods that find comparable scores across different test forms based on score frequencies (equipercentile equating methods). When sample sizes are small, there can be a substantial improvement in the stability of frequency-based equating methods as a result of using loglinear-smoothed frequencies rather than observed frequencies (Livingston, 1993).

SAS/STAT PROC GENMOD

The form of the SAS/STAT PROC GENMOD code that fits loglinear models is:

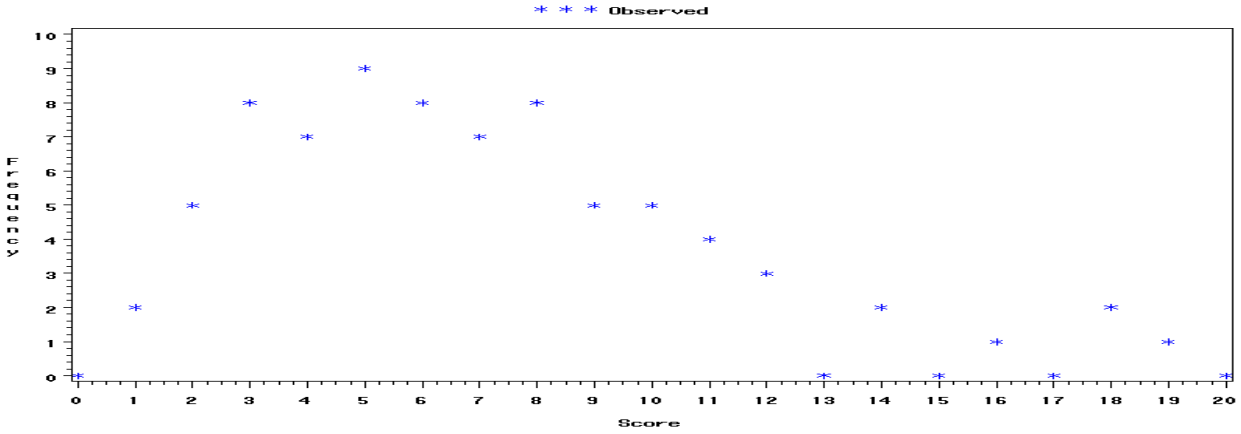
```
proc genmod data=DATA;
output out=RESULTS p=smoothed;
model freq=score1 score2 score3 …. / link=log dist=p type3;
run;
```

The first line invokes the GENMOD procedure for a desired dataset called 'DATA'. This dataset must be in the form of a frequency table rather than a listing of each individual observation, with the score functions ($x_j^1, x_j^2, x_j^3, \ldots, x_j^I$) already created in a SAS Data step. The second line asks for the smoothed frequencies { $\hat{p}_j N$} and the observed frequencies and score functions to be written to a dataset (called 'RESULTS') that can be used to evaluate the results. The third line specifies the model to be fit, where the observed frequencies are to be related to some number of score functions. An intercept that corresponds to α and a scale parameter that is constrained to 1 are included by default in the model. After the required backslash (/), the appropriate link function for 'linking' the frequencies to the score functions (log) and distribution (p=Poisson) are requested. While theoretical derivations show that models based on Poisson and multinomial distributional assumptions produce the same results (Fisher, 1922; Haberman, 1974; Bishop, Fienberg, & Holland, 1975), the Poisson distribution is recommended due to its much greater flexibility in SAS. This flexibility comes from GENMOD treating the score functions as continuous rather than as categorical (as done by the multinomial-based SAS/STAT® PROC CATMOD), so that the scores raised to different powers will lead to preserving different moments. The user can request a sequential modeling process (type1), where the score functions are entered in one-at-a-time, or a non-sequential modeling process (type3).

SMOOTHING UNIVARIATE DISTRIBUTIONS

Assume that test *X* is a 20-item test that has been administered to 77 examinees. The distribution of hypothetical data is plotted in Figure 1 using SAS/GRAPH® PROC GPLOT. The small sample size results in a score distribution that is an unstable representation of the population distribution. The frequencies make abrupt and inconsistent changes between score values 3-9, and no examinees obtain certain scores (0, 13, 15, 17, 20). Despite the instability in this sampled distribution, it should still be representative of the population distribution in terms of its lower-order moments (its mean, variance and skew). Loglinear smoothing should be useful for eliminating the jaggedness, estimating the probability of attaining scores that are possible to attain but were not earned by this particular sample, and preserving some features that are thought to be representative of the population distribution. Appendix 1 contains all of the data and SAS code used for the example.

Figure 1. Observed Frequencies



The data entry step in SAS involves entering the scores and frequencies. Then score functions are defined so that five loglinear models of the form in (1) can be fit to the data, preserving 0, 1, 2, 3 and 4 moments of the distribution.

```
data llin;
input score freq;
cards;
0 0
1 2
2 5
3 8
4 7
5 9
6 8
7 7
8 8
9 5
10 5
11 4
12 3
13 0
14 2
15 0
16 1
17 0
18 2
19 1
20 0
;

data llin;set llin;
score2=score**2;
score3=score**3;
score4=score**4;
```

The following shows the SAS code and output for the three-moment fit.

```
proc genmod data=llinout;
output out=llinout p=p3;
model freq=score score2 score3/link=log dist=p type3;
title '3 Moments';
run;
```

3 Moments
The GENMOD Procedure
Model Information

Data Set	WORK.LLINOUT	Predicted Values and Diagnostic Statistics
Distribution	Poisson	
Link Function	Log	
Dependent Variable	freq	
Observations Used	21	

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	17	14.7116	0.8654
Scaled Deviance	17	14.7116	0.8654
Pearson Chi-Square	17	11.1755	0.6574
Scaled Pearson X2	17	11.1755	0.6574
Log Likelihood		49.7100	

Algorithm converged.

Analysis Of Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-0.0252	0.5493	-1.1018	1.0515	0.00	0.9635
score	1	0.8850	0.2278	0.4386	1.3314	15.10	0.0001
score2	1	-0.1026	0.0275	-0.1566	-0.0487	13.90	0.0002
score3	1	0.0029	0.0009	0.0010	0.0047	9.34	0.0022
Scale	0	1.0000	0.0000	1.0000	1.0000		

The results of all five models are summarized in Table 1, which compares the four moments for the observed distribution and the five smoothed distributions, the degrees of freedom for the smoothing models (21 score values – the number of moments preserved – 1), the deviance and the Pearson Chi-Square statistics.

Table 1. Results from the Observed and Five Models

stats	Observed	_0_Moments	_1_Moments	_2_Moments	_3_Moments	_4_Moments
mean	7.06494	10.0000	7.0649	7.0649	7.0649	7.0649
stdev	3.96195	6.0553	5.6188	3.9620	3.9620	3.9620
skew	0.94922	-0.0000	0.6057	0.3362	0.9492	0.9492
kurtosis	0.79734	-1.2055	-0.6937	-0.3307	1.0197	0.7973
DF	.	20.0000	19.0000	18.0000	17.0000	16.0000
Deviance	.	68.0421	49.5019	23.6294	14.7116	14.1029
PearsonChiSquare	.	55.2727	35.6782	25.8511	11.1755	10.2712

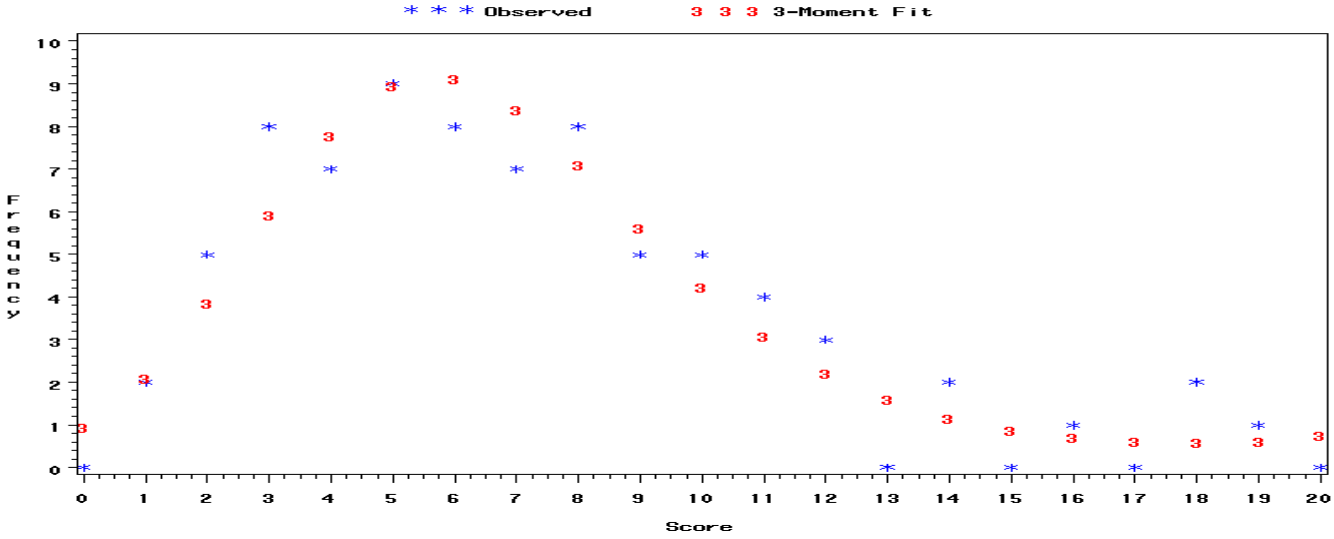
The deviance and the Pearson Chi-Square are two indicators provided by PROC GENMOD for how poorly the smoothed frequencies represent the observed frequencies (SAS, 2002, p. 1,537-1,538):

$$\text{Deviance} = 2\sum_j [n_j \log_e(\frac{n_j}{\hat{p}_j N}) - (n_j - \hat{p}_j N)]. \quad (5)$$

$$\text{Pearson Chi-Square} = \sum_j \frac{(n_j - \hat{p}_j N)^2}{\hat{p}_j N}. \quad (6)$$

The difference in the deviance statistics relative to the difference in the degrees of freedom when comparing nested models can be used to evaluate the null hypothesis that a simpler model that preserves fewer moments is the true model and a more complicated model that preserves additional moments is just fitting noise. The reduction in the deviance statistic is large for the 3-moment fit relative to the 2-moment fit (8.9178, 1 degree of freedom). When evaluated based on a chi-square distribution with one degree of freedom, the improvement obtained from preserving the third moment is statistically significant ($p < .01$). In contrast, the reduction in the deviance statistic is very small for the 4-moment fit relative to the 3-moment fit (.6087, 1 degree of freedom), suggesting that preserving the fourth moment does not significantly improve the model fit ($p > .10$). Figure 2 plots the observed and smoothed frequencies for the 3-moment fit.

Figure 2. Observed and Smoothed Frequencies



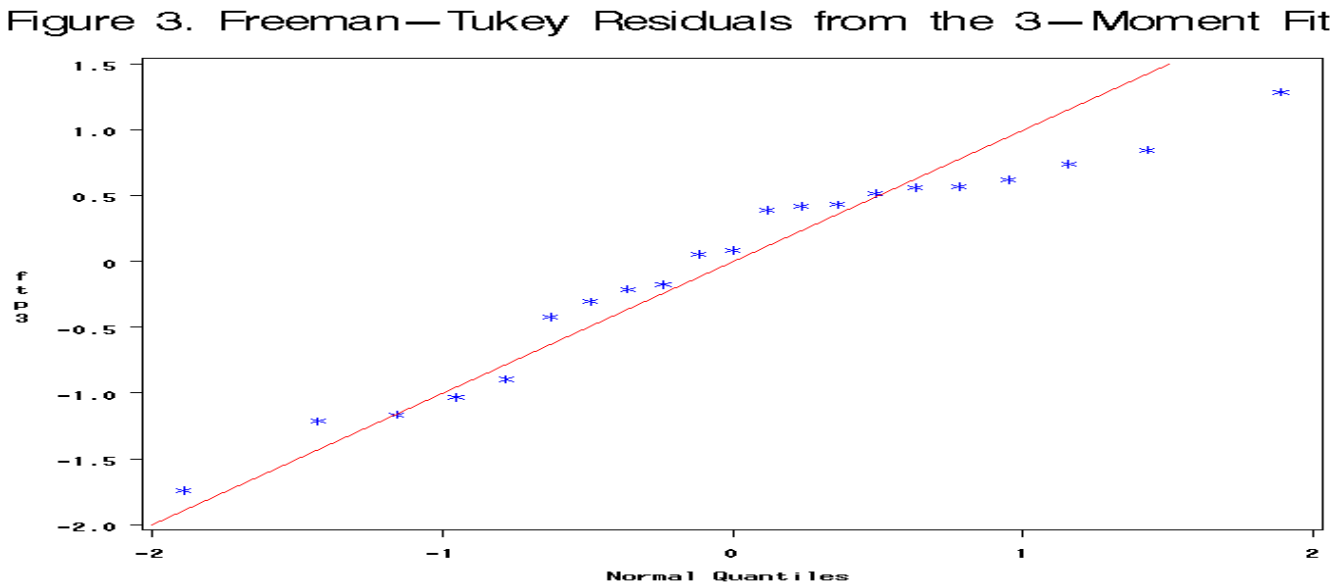
To evaluate the 3-moment fit in more detail, the observed and smoothed frequencies from the 3-moment fit are shown in Table 2 along with Freeman-Tukey (1950) residuals:

$$\text{Freeman-Tukey Residuals} = \sqrt{n_j} + \sqrt{n_j + 1} - \sqrt{4\hat{p}_j N + 1}. \qquad (7)$$

An acceptable model will produce Freeman-Tukey residuals that are approximately normally-distributed around a mean of zero, not too extreme, and without a systematic pattern. These residuals appear to meet these criteria. The distribution of the Freeman-Tukey residuals can also be evaluated graphically. Figure 3 shows the Quantile-Quantile plot produced by SAS/BASE ® PROC UNIVARIATE. This Q-Q plot compares the ordered residuals with the theoretical quantiles from a normal distribution (horizontal axis). Perfectly-normal Freeman-Tukey residuals would lie on the diagonal line.

Table 2. Observed and Smoothed Frequencies and Freeman-Tukey Residuals from the 3-Moment Fit

score	Observed	Smoothed	FTResiduals	score	Observed	Smoothed	FTResiduals
0	0	0.97516	-1.21374	11	4	3.12492	0.56188
1	2	2.13841	0.05537	12	3	2.25190	0.56857
2	5	3.88593	0.61816	13	0	1.62756	-1.74048
3	8	5.95415	0.84680	14	2	1.20042	0.73759
4	7	7.82710	-0.20987	15	0	0.91935	-1.16273
5	9	8.98204	0.08542	16	1	0.74388	0.42034
6	8	9.15538	-0.30521	17	0	0.64707	-0.89427
7	7	8.43415	-0.41960	18	2	0.61566	1.28544
8	8	7.14503	0.38967	19	1	0.65197	0.51477
9	5	5.66371	-0.17807	20	0	0.78187	-1.03162
10	5	4.27433	0.43147				



SMOOTHING BIVARIATE DISTRIBUTIONS

For the example of bivariate smoothing, the analyses of Rosenbaum and Thayer (1987) are reproduced. This problem requires the smoothing of a joint distribution of a 19 item test (*X*) and an 11 item test (*Y*). These tests were administered to a sample of 100 examinees. As with the univariate example, the frequencies are said to be very unstable because they are based on a small sample of examinees. The following analyses demonstrate how to produce more stable estimates of these frequencies through fitting loglinear smoothing models of the form in (2) using PROC GENMOD. Appendix 2 contains all of the data and SAS code used for this example.

The data entry step in SAS involves entering the entire frequency table of the 240 possible *XY* score combinations and their observed frequencies at these combinations. The score functions are then defined so that Rosenbaum and Thayer's (1987) model can be considered.

```
data xy;
input x y freq;
cards;
1 0 1
1 2 1
1 3 3
2 1 2
2 2 2
2 3 1
3 1 1
3 2 3
3 3 1
```

……The entire table of observed (nonzero) bivariate frequencies is entered…….

```
data xym;
do x=0 to 19 by 1;
do y=0 to 10 by 1;
output;
end;
output;
end;

proc sort data=xy;by x y;run;
proc sort data=xym;by x y;run;
data rt;merge xym xy;by x y;

data rt; set rt;
x2=x**2;
x3=x**3;
y2=y**2;
y3=y**3;
xy=x*y;
if freq=. then freq=0;
```

The Rosenbaum and Thayer (1987) model will preserve the means, variances, skews of *X* and *Y* and the *XY* covariance.

```
proc genmod data=rt;
output out=rtfit p=rtfit;
model freq=x x2 x3 y y2 y3 xy /link=log dist=p type3;
title 'Rosenbaum and Thayers (1987) Model';
run;
```

Rosenbaum and Thayers (1987) Model

The GENMOD Procedure
Model Information

Data Set	WORK.RT
Distribution	Poisson
Link Function	Log
Dependent Variable	freq
Observations Used	240

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	232	139.7193	0.6022
Scaled Deviance	232	139.7193	0.6022
Pearson Chi-Square	232	187.2932	0.8073
Scaled Pearson X2	232	187.2932	0.8073
Log Likelihood		-112.2186	

Algorithm converged.

Analysis Of Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-0.6166	0.5392	-1.6734	0.4402	1.31	0.2528
x	1	0.0631	0.1986	-0.3262	0.4523	0.10	0.7507
x2	1	-0.0809	0.0252	-0.1303	-0.0315	10.30	0.0013
x3	1	0.0008	0.0008	-0.0008	0.0023	0.95	0.3308
y	1	0.5632	0.3656	-0.1533	1.2797	2.37	0.1234
y2	1	-0.2483	0.0775	-0.4002	-0.0964	10.27	0.0014
y3	1	0.0021	0.0042	-0.0061	0.0103	0.25	0.6151
xy	1	0.2138	0.0338	0.1475	0.2800	39.97	<.0001
Scale	0	1.0000	0.0000	1.0000	1.0000		

NOTE: The scale parameter was held fixed.

For comparative purposes a second model that preserves the same univariate moments but does not preserve the *XY* covariance is also considered, referred to as the independence model. Comparisons of the results from the two smoothing models show the importance of the *XY* covariance for the fit of the model. The smoothed bivariate frequencies from Rosenbaum and Thayer's model (Table 4) are much closer to the observed bivariate frequencies (Table 3) than the smoothed frequencies obtained from the independence model (Table 5). The smoothed frequencies in the off-diagonal cells of Table 4 are essentially zero, which is expected when a high, positive covariance is preserved (the *XY* covariance and correlation in these data are 12.89 and .862, respectively).

Table 3. Observed XY Distribution

x	y0	y1	y2	y3	y4	y5	y6	y7	y8	y9	y10	y11
0	.	.	.	.	.	.	.	.	.	.	.	.
1	1	.	1	3	.	.	.	.	.	.	.	.
2	.	2	2	1	.	.	.	.	.	.	.	.
3	.	1	3	1	.	.	.	.	.	.	.	.
4	1	1	.	1	.	5	.	.	.	.	.	.
5	.	.	1	2	1	3	1	.	.	.	.	.
6	.	.	2	2	2	1	.	.	.	.	.	.
7	.	.	.	.	.	1	.	.	.	.	.	.
8	.	.	.	2	1	3	1	1	.	.	.	.
9	.	.	1	.	1	.	3	1	.	.	.	.
10	.	.	.	.	1	.	.	1	1	.	.	.
11	.	.	1	.	.	1	2	1	1	2	.	.
12	.	.	.	.	.	.	1	.	.	1	.	.
13	.	.	.	.	.	1	3	3	2	2	.	.
14	.	.	.	.	.	.	.	1	1	.	.	.
15	.	.	.	.	.	.	.	1	1	2	1	.
16	.	.	.	.	.	.	1	.	1	.	.	1
17	.	.	.	.	.	.	.	.	1	3	3	.
18	.	.	.	.	.	.	.	1	.	.	1	1
19	.	.	.	.	.	.	.	.	.	.	.	3

Table 4. Smoothed XY Distribution from the Rosenbaum and Thayer model

x	y0	y1	y2	y3	y4	y5	y6	y7	y8	y9	y10	y11
0	0.54	0.74	0.63	0.33	0.11	0.02	0.00	0.00	0.00	0.00	0.00	0.00
1	0.53	0.90	0.95	0.62	0.26	0.07	0.01	0.00	0.00	0.00	0.00	0.00
2	0.45	0.94	1.22	0.99	0.51	0.17	0.04	0.00	0.00	0.00	0.00	0.00
3	0.32	0.84	1.35	1.35	0.86	0.35	0.09	0.02	0.00	0.00	0.00	0.00
4	0.20	0.65	1.29	1.60	1.25	0.63	0.21	0.04	0.01	0.00	0.00	0.00
5	0.11	0.43	1.06	1.63	1.59	0.99	0.40	0.11	0.02	0.00	0.00	0.00
6	0.05	0.25	0.76	1.46	1.75	1.35	0.68	0.22	0.05	0.01	0.00	0.00
7	0.02	0.13	0.48	1.13	1.69	1.62	1.00	0.41	0.11	0.02	0.00	0.00
8	0.01	0.06	0.27	0.78	1.43	1.70	1.30	0.66	0.22	0.05	0.01	0.00
9	0.00	0.02	0.13	0.47	1.07	1.57	1.49	0.93	0.39	0.11	0.02	0.00
10	0.00	0.01	0.06	0.25	0.71	1.29	1.52	1.17	0.60	0.21	0.05	0.01
11	0.00	0.00	0.02	0.12	0.42	0.95	1.38	1.32	0.84	0.36	0.10	0.02
12	0.00	0.00	0.01	0.05	0.22	0.62	1.12	1.32	1.04	0.55	0.20	0.05
13	0.00	0.00	0.00	0.02	0.11	0.37	0.82	1.20	1.17	0.77	0.34	0.11
14	0.00	0.00	0.00	0.01	0.05	0.19	0.54	0.98	1.18	0.96	0.53	0.20
15	0.00	0.00	0.00	0.00	0.02	0.09	0.32	0.72	1.08	1.09	0.75	0.36
16	0.00	0.00	0.00	0.00	0.01	0.04	0.18	0.49	0.91	1.13	0.96	0.57
17	0.00	0.00	0.00	0.00	0.00	0.02	0.09	0.30	0.69	1.07	1.13	0.82
18	0.00	0.00	0.00	0.00	0.00	0.01	0.04	0.17	0.49	0.94	1.22	1.10
19	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.09	0.32	0.76	1.22	1.36

Table 5. Smoothed XY Distribution from the independence model

x	y0	y1	y2	y3	y4	y5	y6	y7	y8	y9	y10	y11
0	0.07	0.12	0.20	0.26	0.31	0.32	0.30	0.27	0.22	0.19	0.15	0.13
1	0.09	0.17	0.26	0.35	0.41	0.42	0.40	0.35	0.30	0.25	0.21	0.17
2	0.11	0.21	0.33	0.44	0.51	0.53	0.50	0.44	0.38	0.31	0.26	0.22
3	0.13	0.25	0.39	0.52	0.61	0.63	0.60	0.53	0.45	0.37	0.31	0.26
4	0.15	0.28	0.44	0.59	0.69	0.72	0.68	0.60	0.51	0.42	0.35	0.30
5	0.16	0.31	0.48	0.65	0.75	0.78	0.74	0.65	0.55	0.46	0.38	0.32
6	0.17	0.32	0.51	0.68	0.79	0.82	0.78	0.69	0.58	0.48	0.40	0.34
7	0.17	0.33	0.51	0.69	0.80	0.83	0.79	0.70	0.59	0.49	0.40	0.34
8	0.17	0.32	0.51	0.68	0.79	0.82	0.78	0.69	0.58	0.48	0.40	0.34
9	0.16	0.31	0.49	0.66	0.77	0.80	0.75	0.67	0.56	0.47	0.39	0.33
10	0.16	0.30	0.47	0.62	0.73	0.76	0.72	0.63	0.53	0.44	0.37	0.31
11	0.15	0.28	0.44	0.59	0.68	0.71	0.67	0.59	0.50	0.42	0.34	0.29
12	0.14	0.26	0.41	0.55	0.64	0.66	0.63	0.55	0.47	0.39	0.32	0.27
13	0.13	0.24	0.38	0.51	0.59	0.61	0.58	0.51	0.44	0.36	0.30	0.25
14	0.12	0.22	0.35	0.47	0.55	0.57	0.54	0.48	0.41	0.34	0.28	0.24
15	0.11	0.21	0.33	0.45	0.52	0.54	0.51	0.45	0.38	0.32	0.26	0.22
16	0.11	0.20	0.32	0.42	0.49	0.51	0.49	0.43	0.36	0.30	0.25	0.21
17	0.10	0.19	0.31	0.41	0.48	0.50	0.47	0.42	0.35	0.29	0.24	0.21
18	0.10	0.19	0.30	0.41	0.48	0.49	0.47	0.41	0.35	0.29	0.24	0.20
19	0.10	0.20	0.31	0.42	0.49	0.50	0.48	0.42	0.36	0.30	0.25	0.21

Table 6 compares the univariate and bivariate moments of *X* and *Y* based on the observed and smoothed frequencies. When the *XY* covariance is preserved (the Rosenbaum-Thayer model) the deviance and Pearson Chi-Square statistics are 130.121 and 192.788 smaller than when the *XY* covariance is not preserved. This indicates that the *XY* covariance is an important feature in the observed distribution that should be preserved. Note that the extremely small and large fit statistics relative to their degrees of freedom indicates that these statistics are probably not χ^2-distributed, though the differences between them when based on differing and nested models (what a likelihood-ratio χ^2 test would evaluate) is thought to be more closely χ^2-distributed (Agresti, 2002). Finally, when evaluating the fit of bivariate distributions, it can be helpful to compare conditional observed and smoothed moments. Table 7 shows the conditional means of *Y* at each value of *X* in the observed and two smoothed distributions. Table 7 shows that when the *XY* covariance is preserved, the smoothed conditional means of *Y* increase with *X* like the observed conditional means.

Table 6. Results from the Observed and Two Models

stats	Observed	Rosenbaum_ Thayer_ Model	Independence_ Model
xmean	9.350	9.350	9.350
xstdev	5.228	5.228	5.228
xskew	0.141	0.141	0.141
xkurt	-1.150	-1.007	-0.984
ymean	5.620	5.620	5.620
ystdev	2.859	2.859	2.859
yskew	0.087	0.087	0.087
ykurt	-0.914	-0.886	-0.820
corrxy	0.862	0.862	0.000
DF	.	232.000	233.000
Deviance	.	139.719	269.840
PearsonChiSquare	.	187.293	380.081

Table 7. Observed and Smoothed Conditional Mean of Y Given X

x	Observed	Rosenbaum_ Thayer_ Model	Independence_ Model
0	.	1.50279	5.62000
1	2.2000	1.82719	5.62000
2	1.8000	2.19377	5.62000
3	2.0000	2.59669	5.62000
4	3.6250	3.02837	5.62000
5	4.1250	3.48127	5.62000
6	3.2857	3.94930	5.62000
7	5.0000	4.42839	5.62000
8	4.7500	4.91630	5.62000
9	5.1667	5.41198	5.62000
10	6.3333	5.91484	5.62000
11	6.5000	6.42396	5.62000
12	7.5000	6.93708	5.62000
13	7.0909	7.44934	5.62000
14	7.5000	7.95210	5.62000
15	8.6000	8.43299	5.62000
16	8.3333	8.87785	5.62000
17	9.2857	9.27426	5.62000
18	9.3333	9.61475	5.62000
19	11.0000	9.89804	5.62000

INDICATOR FUNCTIONS AND LOGLINEAR SMOOTHING MODELS

Two final examples demonstrate the use of indicator functions to address special features of a univariate distribution. These features are a “lump at zero” and subgroup differences. Both use variations of the dataset used for the univariate example. Complete SAS code is not given for these examples, but the specific differences that are beyond what was needed for the simpler univariate example are given.

LUMP AT ZERO PROBLEM

For the first example, we deal with a common case encountered in test data where a large group of examinees gets the zero score on the test. The dataset used in the univariate example is used, but ten instead of zero examinees obtain the zero score. This makes the score of zero the most common score. Two models are considered to smooth the frequencies. The first model is the selected model from the univariate example, which preserved the mean, variance and skew of the overall distribution but would ignore the lump at zero. A second model adds an indicator function to the first model so that in addition to preserving the first three overall moments, the large frequency at the score of zero can also be preserved. The defining of the indicator function and then the fitting of the second model involve the following SAS code:

```
data llin; set llin; if score=0 then do; freq=10; i=1; end; if i=. then i=0;

proc genmod data=llin;
output out=llinout p=lump;
model freq=score score2 score3 i/link=log dist=p type3;
title 'Preserve the Lump';
run;
```

The importance of preserving the lump at zero can be seen in the overall fit statistics, the smoothed distributions, and the residuals of the smoothed distributions. The deviance statistic and degrees of freedom go from 21.14 and 17 when the lump is ignored to 12.37 and 16 when the lump is preserved. Figure 4 plots the observed and smoothed frequencies from the two models and shows what the overall fit statistics suggest, that preserving the lump enhances the fit of all the smoothed frequencies and not just for the score of zero. Below Figure 4 is Table 8 which gives the Freeman-Tukey residuals from each model and shows that when the lump is ignored (ftnl, the second column), 12 of the 21 residuals are larger than when the lump is preserved (ftl, the third column).

Figure 4. Ignoring and Preserving the Lump at Score Zero

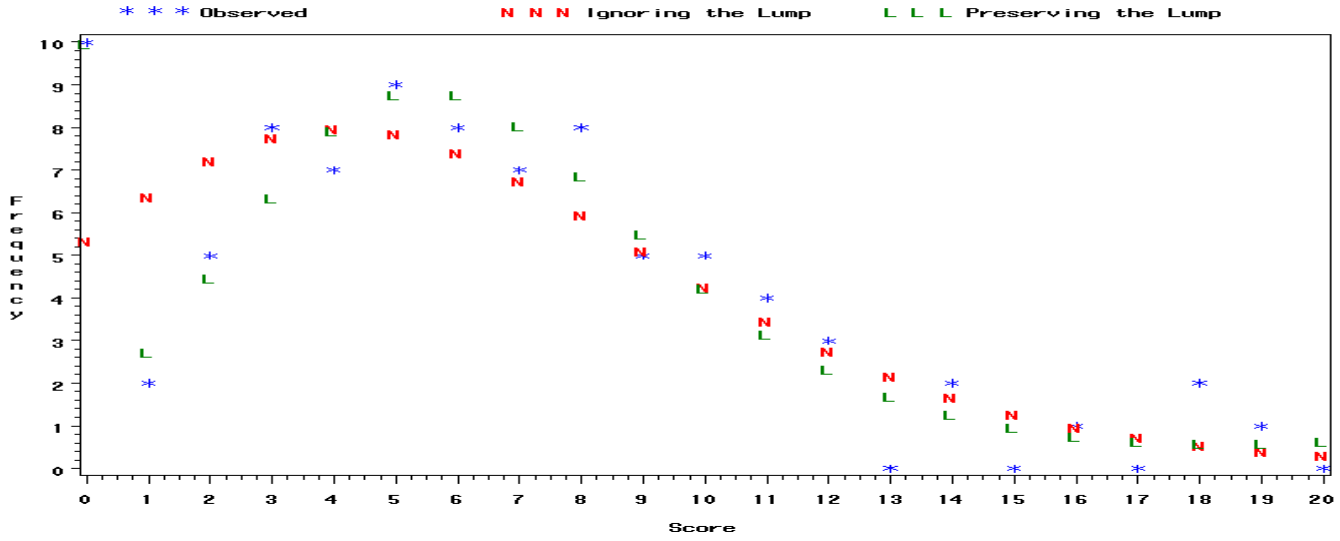


Table 8.
Comparing Freeman-Tukey Residuals From Preserving vs. Ignoring the Lump at Zero

score	ftnl	ftl
0	1.73658	0.07578
1	-2.01331	-0.32203
2	-0.78894	0.32064
3	0.15660	0.67391
4	-0.27195	-0.25234
5	0.46093	0.14491
6	0.27904	-0.19085
7	0.16626	-0.29798
8	0.83053	0.48333
9	0.04426	-0.13017
10	0.42641	0.43142

score	ftnl	ftl
11	0.36599	0.52221
12	0.24252	0.50245
13	-2.12936	-1.81929
14	0.34832	0.65797
15	-1.50044	-1.23396
16	0.17490	0.36446
17	-1.01483	-0.92864
18	1.32071	1.28029
19	0.74535	0.55186
20	-0.54137	-0.92726

SUBGROUP PROBLEM

The second example of indicator functions illustrates how loglinear smoothing models can be used to compare subgroups' distributions. For this problem, very different subgroup distributions are created from the univariate dataset. A second dataset is created as a mirror image of the first dataset so that in this second subgroup the mean is much higher and the skew is negative rather than positive. The descriptive statistics for the two subgroups are shown in Table 9.

Table 9. Test Score Statistics for the Subgroups

Subgroup	n	mean	stdev	skew	kurtosis
1	77	7.0649	3.96195	0.94922	.79734
2	77	12.9351	3.96195	-0.94922	.79734

The following SAS code shows how the scores are reversed to create the data for the second subgroup. Then PROC GENMOD is used to consider a model that preserves the overall mean, variance and skew as well as the subgroup-specific frequencies, means, variances and skews.

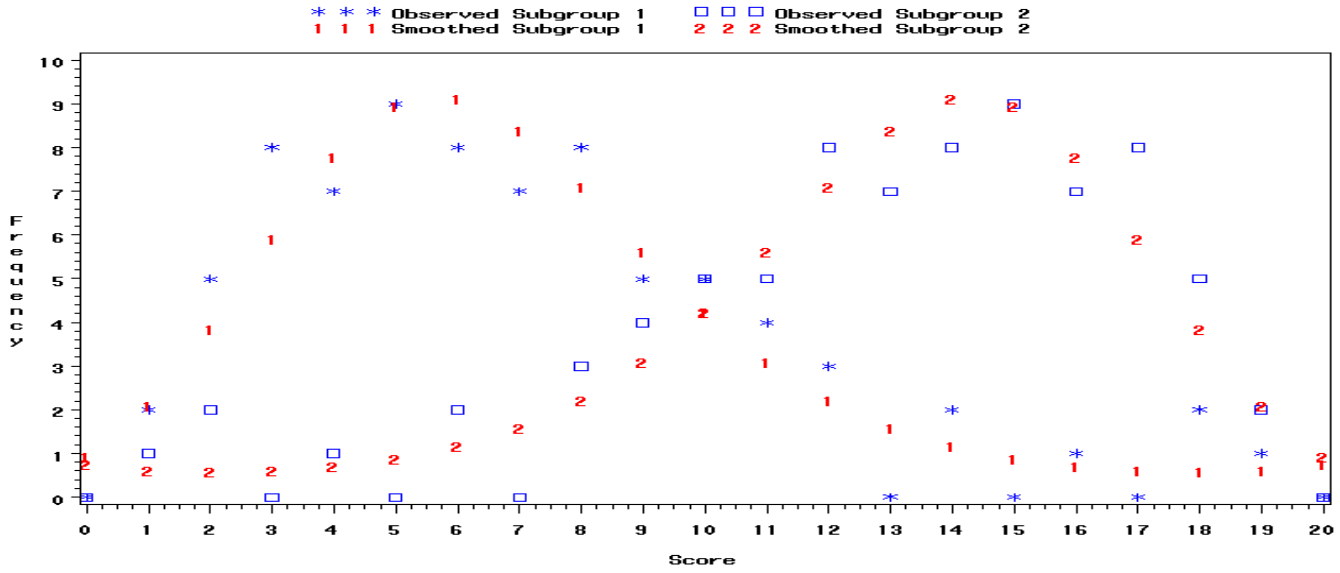
```
data llin1;set llin;
i=0;

data llin2;set llin;
scoreq=20-score;
score=scoreg;
i=1;
drop scoreg;
run;

data llin;set llin1 llin2;
score2=score**2;
score3=score**3;
iscore=i*score;
iscore2=i*score2;
iscore3=i*score3;
run;

proc genmod data=llin;
output out=llinout p=pf3;
model freq=score score2 score3 i iscore iscore2 iscore3/link=log dist=p type3;
run;
```

The fit statistics of four models of increasing complexity are compared in Table 10. These preserve the separate frequencies, then the means, then the variances, and then the skews of the separate subgroups. The improvement in fit corresponds exactly to the differences in the subgroup statistics. That is, for the subgroup characteristics that are exactly the same (variances), no improvement in fit is seen in models that preserve them over models that ignore them. For subgroup characteristics that are very different (means and skews), large improvements are shown in the fit statistics when models preserve them. Finally Figure 5 plots the two observed and smoothed distributions from the model that allows the means, standard deviations and skews to differ in the subgroups.

Table 10. Fit Statistics of Models that Preserve Subgroup Characteristics

stats	Subgroup Frequencies	Subgroup Frequencies & Means	Subgroup Frequencies, Means and Variances	Subgroup Frequencies, Mean, Variances and Skews
DF	37	36	35	34
Deviance	111.66	47.26	47.26	29.42
PearsonChiSquare	95.14	51.70	51.70	22.35

Figure 5. Observed and Smoothed Frequencies For the Subgroups



CONCLUSIONS

SAS/STAT PROC GENMOD can be used for many loglinear smoothing problems, including other important smoothing problems that were not covered in this paper but are covered elsewhere (Moses, von Davier & Casabianca, 2004; Moses & von Davier, 2005). It can be recommended for general use in the smoothing of univariate distributions. With respect to bivariate distributions, PROC GENMOD can have convergence problems. Except for some troublesome bivariate situations, PROC GENMOD can be a flexible smoothing tool that is easy to learn and use.

A SAS® IML macro is currently available (Moses & von Davier, 2005) that implements loglinear smoothing in ways that are more closely-suited to issues with testing data (Holland & Thayer, 2000) than PROC GENMOD. Specifically, the Newton algorithm for maximizing the likelihood function is based on the variance-covariance matrix of the smoothed frequencies rather than on the variance-covariance matrix of the parameter estimates. The Newton algorithm also uses different criteria for identifying convergence. These differences make the IML macro's convergence more likely than GENMOD's convergence for highly-correlated bivariate data. In addition, the IML macro produces estimates of the variance-covariance matrix of the smoothed frequencies in a factored "C-matrix".

Loglinear smoothing is interesting to contrast with other types of smoothing, especially kernel smoothing (Ramsay, 1991). Kernel smoothing uses a weighted and moving average approach to smoothing frequencies. It will produce a "continuized" version of an originally-discrete frequency distribution. In contrast to loglinear smoothing, a stronger form of kernel smoothing that uses a large bandwidth parameter can produce a smoothed frequency distribution with moments that can deviate from those of the original distribution. Kernel smoothing is more suited to simple univariate distributions and is considered non-parametric, meaning that the plausibility of specific and complex features existing in distributions cannot be evaluated with statistical hypothesis tests as they can with loglinear smoothing models. An application that utilizes the unique benefits of loglinear and kernel smoothing is the kernel method of test equating (von Davier, Holland & Thayer, 2004).

REFERENCES

Agresti, A. (2002). Categorical Data Analysis, Second Edition. New York: Wiley.

Bishop, Y. M. M., Fienberg, S. E., & Holland, P. W. (1975). Discrete Multivariate Analysis. MIT Press: Cambridge.

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

Fisher, R. A. (1922). On the interpretation of χ^2 from contingency tables, and the calculation of p. Journal of the Royal Statistical Society, 85, 87-94.

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

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

Holland, P. W. & Thayer, D. T. (1987). Notes on the use of log-linear models for fitting discrete probability distributions. (Technical Report 87-79). Princeton, NJ: Educational Testing Service.

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.

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

Moses T., & von Davier, A. A. (2005). A SAS macro for loglinear smoothing: Applications and implications. Paper presented at the American Educational Research Association, Montreal, CA.

Moses, T., von Davier, A. A., & Casabianca, J. (2004). Loglinear smoothing: An alternative numerical approach using SAS. (Research Report 04-27), Princeton, NJ: Educational Testing Service.

Ramsay, J. O. (1991). Kernel smoothing approaches to nonparametric item characteristic curve estimation. Psychometrika, 56(4), 611-630.

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. SAS/STAT Software: (2002). The GENMOD procedure, Version 9. Cary, NC: SAS Institute.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Tim Moses
Educational Testing Service
Princeton, NJ 08541
Phone: (609) 683-2208
Fax: (609) 683-2130
E-mail: tmoses@ets.org

APPENDIX 1: SAS CODE FOR SMOOTHING UNIVARIATE DISTRIBUTIONS

```
data llin;
input score freq;
cards;
0 0
1 2
2 5
3 8
4 7
5 9
6 8
7 7
8 8
9 5
10 5
11 4
12 3
13 0
14 2
15 0
16 1
17 0
18 2
19 1
20 0
;
run;

axis1 label=(angle=-90 rotate=90 'Frequency') order=(0 to 10 by 1);
axis2 order=(0 to 20 by 1) label=('Score' height=3in );
Legend1 label=(height=1 position=top justify=center '')
       value=('Observed' ) position=(top center);
symbol1 color=blue interpol=none width=1 value=star height=1;
proc gplot data=llin;
plot freq*score / overlay vaxis=axis1 haxis=axis2 legend=legend1;
title 'Figure 1. Observed Frequencies';
run;quit;

data llin;set llin;
score2=score**2;
score3=score**3;
score4=score**4;

ods output Genmod.ModelFit=model0;
proc genmod data=llin;
output out=llinout p=p0;
model freq=/link=log dist=p type3;
title '0 Moments';
run;
data model0d;set model0;if criterion NE 'Deviance' then delete;drop valuedf criterion;rename
value=Deviance;
data model0p;set model0;if criterion NE 'Pearson Chi-Square' then delete;drop valuedf criterion;rename
value=PearsonChiSquare;

ods output Genmod.ModelFit=model1;
proc genmod data=llinout;
output out=llinout p=p1;
model freq=score/link=log dist=p type3;
title '1 Moment';
run;
data model1d;set model1;if criterion NE 'Deviance' then delete;drop valuedf criterion;rename
value=Deviance;
data model1p;set model1;if criterion NE 'Pearson Chi-Square' then delete;drop valuedf criterion;rename
value=PearsonChiSquare;

ods output Genmod.ModelFit=model2;
proc genmod data=llinout;
output out=llinout p=p2;
model freq=score score2/link=log dist=p type3;
title '2 Moments';
run;
data model2d;set model2;if criterion NE 'Deviance' then delete;drop valuedf criterion;rename
value=Deviance;
data model2p;set model2;if criterion NE 'Pearson Chi-Square' then delete;drop valuedf criterion;rename
value=PearsonChiSquare;

ods output Genmod.ModelFit=model3;
proc genmod data=llinout;
output out=llinout p=p3;
model freq=score score2 score3/link=log dist=p type3;
title '3 Moments';
run;
```

```
data model3d;set model3;if criterion NE 'Deviance' then delete;drop valuedf criterion;rename
value=Deviance;
data model3p;set model3;if criterion NE 'Pearson Chi-Square' then delete;drop valuedf criterion;rename
value=PearsonChiSquare;

ods output Genmod.ModelFit=model4;
proc genmod data=llinout;
output out=llinout p=p4;
model freq=score score2 score3 score4/link=log dist=p type3;
title '4 Moments';
run;
data model4d;set model4;if criterion NE 'Deviance' then delete;drop valuedf criterion;rename
value=Deviance;
data model4p;set model4;if criterion NE 'Pearson Chi-Square' then delete;drop valuedf criterion;rename
value=PearsonChiSquare;

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' '3-Moment Fit') position=(top center);
symbol1 color=blue interpol=none width=1 value=star height=1;
symbol2 color=red interpol=none width=1 value=3 height=1;
proc gplot data=llinout;
plot freq*score p3*score / overlay vaxis=axis1 haxis=axis2 legend=legend1;
title 'Figure 2. Observed and Smoothed Frequencies';
run;quit;



proc means data=llinout noprint;var freq;output out=ntot sum=ntot;run;
data llinout;if _n_=1 then set ntot;set llinout;
scoreobs=score*freq/ntot;
scorep0=score*p0/ntot;
scorep1=score*p1/ntot;
scorep2=score*p2/ntot;
scorep3=score*p3/ntot;
scorep4=score*p4/ntot;
run;
proc means data=llinout noprint;var scoreobs scorep0-scorep4;output out=means sum=meanobs mean0-
mean4;run;

data llinout;if _n_=1 then set means;set llinout;
score2obs=(score-meanobs)**2*freq/ntot;
score2p0=(score-mean0)**2*p0/ntot;
score2p1=(score-mean1)**2*p1/ntot;
score2p2=(score-mean2)**2*p2/ntot;
score2p3=(score-mean3)**2*p3/ntot;
score2p4=(score-mean4)**2*p4/ntot;
run;
proc means data=llinout noprint;var score2obs score2p0-score2p4;output out=variances sum=varobs var0-
var4;run;

data llinout;if _n_=1 then set variances;set llinout;
score3obs=((score-meanobs)**3*freq/ntot)/(varobs**(3/2));
score3p0=((score-mean0)**3*p0/ntot)/(var0**(3/2));
score3p1=((score-mean1)**3*p1/ntot)/(var1**(3/2));
score3p2=((score-mean2)**3*p2/ntot)/(var2**(3/2));
score3p3=((score-mean3)**3*p3/ntot)/(var3**(3/2));
score3p4=((score-mean4)**3*p4/ntot)/(var4**(3/2));
run;
proc means data=llinout noprint;var score3obs score3p0-score3p4;output out=skews sum=skewobs skew0-
skew4;run;

data llinout;set llinout;
score4obs=((score-meanobs)**4*freq/ntot)/(varobs**2);
score4p0=((score-mean0)**4*p0/ntot)/(var0**2);
score4p1=((score-mean1)**4*p1/ntot)/(var1**2);
score4p2=((score-mean2)**4*p2/ntot)/(var2**2);
score4p3=((score-mean3)**4*p3/ntot)/(var3**2);
score4p4=((score-mean4)**4*p4/ntot)/(var4**2);
run;
proc means data=llinout noprint;var score4obs score4p0-score4p4;output out=kurts sum=kurtobs kurt0-
kurt4;run;

data stats;merge means variances skews kurts;run;


data statso;set stats(rename=meanobs=mean rename=varobs=stdev rename=skewobs=skew
rename=kurtobs=kurtosis);
_TYPE_=-1;
keep mean stdev skew kurtosis _type_;
stdev=sqrt(stdev);
kurtosis=kurtosis-3;
run;
```

```
data stats0;set stats(rename=mean0=mean rename=var0=stdev rename=skew0=skew rename=kurt0=kurtosis);
_TYPE_=0;
keep mean stdev skew kurtosis _type_;
stdev=sqrt(stdev);
kurtosis=kurtosis-3;
run;
data stats0;merge stats0 model0d model0p;run;

data stats1;set stats(rename=mean1=mean rename=var1=stdev rename=skew1=skew rename=kurt1=kurtosis);
_TYPE_=1;
keep mean stdev skew kurtosis _type_;
stdev=sqrt(stdev);
kurtosis=kurtosis-3;
run;
data stats1;merge stats1 model1d model1p;run;

data stats2;set stats(rename=mean2=mean rename=var2=stdev rename=skew2=skew rename=kurt2=kurtosis);
_TYPE_=2;
keep mean stdev skew kurtosis _type_;
stdev=sqrt(stdev);
kurtosis=kurtosis-3;
run;
data stats2;merge stats2 model2d model2p;run;

data stats3;set stats(rename=mean3=mean rename=var3=stdev rename=skew3=skew rename=kurt3=kurtosis);
_TYPE_=3;
keep mean stdev skew kurtosis _type_;
stdev=sqrt(stdev);
kurtosis=kurtosis-3;
run;
data stats3;merge stats3 model3d model3p;run;

data stats4;set stats(rename=mean4=mean rename=var4=stdev rename=skew4=skew rename=kurt4=kurtosis);
_TYPE_=4;
keep mean stdev skew kurtosis _type_;
stdev=sqrt(stdev);
kurtosis=kurtosis-3;
run;
data stats4;merge stats4 model4d model4p;run;

proc format;
       value name            -1='Observed'
                             0='0 Moments'
                             1='1 Moments'
                             2='2 Moments'
                             3='3 Moments'
                             4='4 Moments'
;
run;

data results;set statso stats0 stats1 stats2 stats3 stats4;
format _type_ name.;

proc transpose data=results out=results;id _type_;
data results;set results; rename _name_=stats;

proc print data=results noobs;title 'Table 1. Results from the Observed and Five Models';run;

data llinout;set llinout;
ftp3=sqrt(freq)+sqrt(freq+1)-sqrt(4*p3+1);

data llinout3;set llinout;keep score freq p3 ftp3;rename freq=Observed;rename p3=Smoothed;rename
ftp3=FTResiduals;
proc print data=llinout3 noobs;
title 'Table 2. Observed and Smoothed Frequencies and Freeman-Tukey Residuals from the 3-Moment Fit';
run;

proc univariate data=llinout;
var ftp3;
qqplot ftp3 / normal(mu=0 sigma=1 l=1);
title 'Figure 3. Freeman-Tukey Residuals from the 3-Moment Fit';
run;
```

APPENDIX 2: SAS CODE FOR SMOOTHING BIVARIATE DISTRIBUTIONS

```
data xy;
input x y freq;
cards;
1 0 1
1 2 1
1 3 3
2 1 2
2 2 2
2 3 1
3 1 1
3 2 3
3 3 1
4 0 1
4 1 1
4 3 1
4 5 5
5 2 1
5 3 2
5 4 1
5 5 3
5 6 1
6 2 2
6 3 2
6 4 2
6 5 1
7 5 1
8 3 2
8 4 1
8 5 3
8 6 1
8 7 1
9 2 1
9 4 1
9 6 3
9 7 1
10 4 1
10 7 1
10 8 1
11 2 1
11 5 1
11 6 2
11 7 1
11 8 1
11 9 2
12 6 1
12 9 1
13 5 1
13 6 3
13 7 3
13 8 2
13 9 2
14 7 1
14 8 1
15 7 1
15 8 1
15 9 2
15 10 1
16 6 1
16 8 1
16 11 1
17 8 1
17 9 3
17 10 3
18 7 1
18 10 1
18 11 1
19 11 3
;

data xym;
do x=0 to 19 by 1;
do y=0 to 10 by 1;
output;
```

```
end;
output;
end;

proc sort data=xy;by x y;run;
proc sort data=xym;by x y;run;
data rt;merge xym xy;by x y;run;


%macro stack;
       %do i=0 %to 11;
              data table3&i;set rt;if y=&i;keep x freq;
              data table3&i;set table3&i;rename freq=y&i;
       %end;
%mend stack;
%stack;
data table3;merge %macro merge; %do i=0 %to 11;
table3&i
%end;;
by x;
%mend merge;
%merge;
proc print data=table3 noobs;title 'Table 3. Observed XY Distribution';run;


data rt; set rt;
x2=x**2;
x3=x**3;
y2=y**2;
y3=y**3;
xy=x*y;
if freq=. then freq=0;
drop percent;

ods output Genmod.ModelFit=modelrt;
proc genmod data=rt;
output out=rtfit p=rtfit;
model freq=x x2 x3 y y2 y3 xy /link=log dist=p type3;
title 'Rosenbaum and Thayers (1987) Model';
run;
data modelrtd;set modelrt;if criterion NE 'Deviance' then delete;drop valuedf criterion;rename
value=Deviance;
data modelrtp;set modelrt;if criterion NE 'Pearson Chi-Square' then delete;drop valuedf criterion;rename
value=PearsonChiSquare;

%macro stack;
       %do i=0 %to 11;
              data table4&i;set rtfit;if y=&i;keep x rtfit;rtfit=round(rtfit,.01);
              data table4&i;set table4&i;rename rtfit=y&i;
       %end;
%mend stack;
%stack;
data table4;merge %macro merge; %do i=0 %to 11;
table4&i
%end;;
by x;
%mend merge;
%merge;
proc print data=table4 noobs;title 'Table 4. Smoothed XY Distribution from the Rosenbaum and Thayer
model';run;

ods output Genmod.ModelFit=modelindep;
proc genmod data=rtfit;
output out=rtfit p=indepfit;
model freq=x x2 x3 y y2 y3 /link=log dist=p type3;
title 'X-Y Independence Model';
run;
data modelindepd;set modelindep;if criterion NE 'Deviance' then delete;drop valuedf criterion;rename
value=Deviance;
data modelindepp;set modelindep;if criterion NE 'Pearson Chi-Square' then delete;drop valuedf
criterion;rename value=PearsonChiSquare;


%macro stack;
       %do i=0 %to 11;
```

```
               data table5&i;set rtfit;if y=&i;keep x indepfit;indepfit=round(indepfit,.01);
               data table5&i;set table5&i;rename indepfit=y&i;
       %end;
%mend stack;
%stack;
data table5;merge %macro merge; %do i=0 %to 11;
table5&i
%end;;
by x;
%mend merge;
%merge;
proc print data=table5 noobs;title 'Table 5. Smoothed XY Distribution from the independence model';run;


proc means data=rtfit noprint;var freq;output out=ntot sum=ntot;run;

proc means data=rtfit noprint;var freq rtfit indepfit;class x; output out=rtfitx sum=freqx rtfitx
indepfitx;run;
data rtfitx;set rtfitx;if x=. then delete;run;

proc means data=rtfit noprint;var freq rtfit indepfit;class y; output out=rtfity sum=freqy rtfity
indepfity;run;
data rtfity;set rtfity;if y=. then delete;run;

data rtfitx;if _n_=1 then set ntot;set rtfitx;
xobs=x*freqx/ntot;
xrt=x*rtfitx/ntot;
xindep=x*indepfitx/ntot;
run;
proc means data=rtfitx noprint;var xobs xrt xindep;
output out=xmeans sum=xmeanobs xmeanrt xmeanindep;run;

data rtfitx;if _n_=1 then set xmeans;set rtfitx;
x2obs=(x-xmeanobs)**2*freqx/ntot;
x2rt=(x-xmeanrt)**2*rtfitx/ntot;
x2indep=(x-xmeanindep)**2*indepfitx/ntot;
run;
proc means data=rtfitx noprint;var x2obs x2rt x2indep;
output out=xvariances sum=xstdobs xstdrt xstdindep;run;

data rtfitx;if _n_=1 then set xvariances;set rtfitx;
x3obs=((x-xmeanobs)**3*freqx/ntot)/(xstdobs**(3/2));
x3rt=((x-xmeanrt)**3*rtfitx/ntot)/(xstdrt**(3/2));
x3indep=((x-xmeanindep)**3*indepfitx/ntot)/(xstdindep**(3/2));
run;
proc means data=rtfitx noprint;var x3obs x3rt x3indep;
output out=xskews sum=xskewobs xskewrt xskewindep;run;

data rtfitx;set rtfitx;
x4obs=((x-xmeanobs)**4*freqx/ntot)/(xstdobs**2);
x4rt=((x-xmeanrt)**4*rtfitx/ntot)/(xstdrt**2);
x4indep=((x-xmeanindep)**4*indepfitx/ntot)/(xstdindep**2);
run;
proc means data=rtfitx noprint;var x4obs x4rt x4indep;
output out=xkurts sum=xkurtobs xkurtrt xkurtindep;run;

data xstats;merge xmeans xvariances xskews xkurts;run;

data rtfity;if _n_=1 then set ntot;set rtfity;
yobs=y*freqy/ntot;
yrt=y*rtfity/ntot;
yindep=y*indepfity/ntot;
run;
proc means data=rtfity noprint;var yobs yrt yindep;
output out=ymeans sum=ymeanobs ymeanrt ymeanindep;run;

data rtfity;if _n_=1 then set ymeans;set rtfity;
y2obs=(y-ymeanobs)**2*freqy/ntot;
y2rt=(y-ymeanrt)**2*rtfity/ntot;
y2indep=(y-ymeanindep)**2*indepfity/ntot;
run;
proc means data=rtfity noprint;var y2obs y2rt y2indep;
output out=yvariances sum=ystdobs ystdrt ystdindep;run;

data rtfity;if _n_=1 then set yvariances;set rtfity;
```

```
y3obs=((y-ymeanobs)**3*freqy/ntot)/(ystdobs**(3/2));
y3rt=((y-ymeanrt)**3*rtfity/ntot)/(ystdrt**(3/2));
y3indep=((y-ymeanindep)**3*indepfity/ntot)/(ystdindep**(3/2));
run;
proc means data=rtfity noprint;var y3obs y3rt y3indep;
output out=yskews sum=yskewobs yskewrt yskewindep;run;

data rtfity;set rtfity;
y4obs=((y-ymeanobs)**4*freqy/ntot)/(ystdobs**2);
y4rt=((y-ymeanrt)**4*rtfity/ntot)/(ystdrt**2);
y4indep=((y-ymeanindep)**4*indepfity/ntot)/(ystdindep**2);
run;
proc means data=rtfity noprint;var y4obs y4rt y4indep;
output out=ykurts sum=ykurtobs ykurtrt ykurtindep;run;

data ystats;merge ymeans yvariances yskews ykurts;run;

data rtfit;if _n_=1 then set ntot;set rtfit;
xypxyobs=x*y*freq/ntot;
xypxyrtfit=x*y*rtfit/ntot;
xypxyindepfit=x*y*indepfit/ntot;
proc means data=rtfit noprint;var xypxyobs xypxyrtfit xypxyindepfit;
output out=covxy sum=covxyobs covxyrtfit covxyindepfit;
run;

data allstats;merge xstats ystats covxy;
xstdobs=sqrt(xstdobs);
xstdrt=sqrt(xstdrt);
xstdindep=sqrt(xstdindep);
ystdobs=sqrt(ystdobs);
ystdrt=sqrt(ystdrt);
ystdindep=sqrt(ystdindep);
corrxyobs=(covxyobs-xmeanobs*ymeanobs)/(xstdobs*ystdobs);
corrxyrt=(covxyrtfit-xmeanrt*ymeanrt)/(xstdrt*ystdrt);
corrxyindep=(covxyindepfit-xmeanindep*ymeanindep)/(xstdindep*ystdindep);
xkurtobs=xkurtobs-3;
xkurtrt=xkurtrt-3;
xkurtindep=xkurtindep-3;
ykurtobs=ykurtobs-3;
ykurtrt=ykurtrt-3;
ykurtindep=ykurtindep-3;

data observed;set allstats;
rename xmeanobs=xmean;
rename xstdobs=xstdev;
rename xskewobs=xskew;
rename xkurtobs=xkurt;
rename ymeanobs=ymean;
rename ystdobs=ystdev;
rename yskewobs=yskew;
rename ykurtobs=ykurt;
rename corrxyobs=corrxy;
run;
data observed;set observed;
keep xmean xstd xskew xkurt ymean ystd yskew ykurt corrxy _type_;
_type_=0;
run;

data rt;merge allstats modelrtd modelrtp;
rename xmeanrt=xmean;
rename xstdrt=xstdev;
rename xskewrt=xskew;
rename xkurtrt=xkurt;
rename ymeanrt=ymean;
rename ystdrt=ystdev;
rename yskewrt=yskew;
rename ykurtrt=ykurt;
rename corrxyrt=corrxy;
run;
data rt;set rt;
keep xmean xstd xskew xkurt ymean ystd yskew ykurt corrxy _type_ df deviance pearsonchisquare;
_type_=1;
run;

data indepfit;merge allstats modelindepd modelindepp;
```

```
rename xmeanindep=xmean;
rename xstdindep=xstdev;
rename xskewindep=xskew;
rename xkurtindep=xkurt;
rename ymeanindep=ymean;
rename ystdindep=ystdev;
rename yskewindep=yskew;
rename ykurtindep=ykurt;
rename corrxyindep=corrxy;
run;
data indepfit;set indepfit;
keep xmean xstd xskew xkurt ymean ystd yskew ykurt corrxy _type_ df deviance pearsonchisquare;
_type_=2;
run;

proc format;
       value name            0='Observed'
                             1='Rosenbaum Thayer Model'
                             2='Independence Model'
;
run;
data results;set observed rt indepfit;
format _type_ name.;
proc transpose data=results out=results ;id _type_;
data results;set results;rename _name_=stats;
observed=round(observed,.001);
Rosenbaum_Thayer_Model=round(Rosenbaum_Thayer_Model,.001);
Independence_Model=round(Independence_Model,.001);
proc print data=results noobs;title 'Table 6. Results from the Observed and Two Models';run;


/*X Conditional Means*/
proc sort data=rtfit;by x;run;
data rtfit;merge rtfit rtfitx;by x;
if freqx ne 0 then yobserved=y*freq/freqx;
yrt=y*rtfit/rtfitx;
yindep=y*indepfit/indepfitx;
run;

proc means data=rtfit noprint;var yobserved yrt yindep;class x; output out=condmean sum=yobserved yrt
yindep;run;
data condmean;set condmean;if _type_=0 then delete;drop _freq_ _type_;
proc print data=condmean;
title 'Table 7. Observed and Smoothed Conditional Mean of Y Given X';
run;
```