Categorical Data Analysis, Second Edition. Alan Agresti
Copyright © 2002 John Wiley & Sons, Inc.
ISBN: 0-471-36093-7

APPENDIX A

Using Computer Software to Analyze Categorical Data

In this appendix we discuss statistical software for categorical data analysis, with emphasis on SAS. We begin by mentioning major software that can perform the analyses discussed in this book. Then we illustrate, by chapter, SAS code for the analyses. Information about other packages (such as S-Plus, R, SPSS, and Stata), as well as updated information about SAS, is at the Web site (*www.stat.ufl.edu/~aa/cda/cda.html.*) Section A.2 on SAS also lists other software for analyses not currently available in SAS.

A.1 SOFTWARE FOR CATEGORICAL DATA ANALYSIS

A.1.1 SAS

SAS is general-purpose software for a wide variety of statistical analyses. The main procedures (PROCs) for categorical data analyses are FREQ, GENMOD, LOGISTIC, NLMIXED, and CATMOD.

PROC FREQ computes measures of association and their estimated standard errors. It also performs generalized Cochran–Mantel–Haenszel tests of conditional independence, and exact tests of independence in $I \times J$ tables.

PROC GENMOD fits generalized linear models. It fits cumulative link models for ordinal responses. It can perform GEE analyses for marginal models. One can form one's own variance function and allow scale parameters, making it suitable for quasi-likelihood analyses.

PROC LOGISTIC gives ML fitting of binary response models, cumulative link models for ordinal responses, and baseline-category logit models for nominal responses. It incorporates model selection procedures, regression diagnostic options, and exact conditional inference. PROC PROBIT also conducts ML fitting of binary and cumulative link models as well as quantal response models that permit a strictly positive probability as the linear predictor decreases to $-\infty$.

PROC CATMOD fits baseline-category logit models. It is also useful for WLS fitting of a wide variety of models for categorical data.

PROC NLMIXED fits generalized linear mixed models (GLMMs). It approximates the likelihood using adaptive Gauss–Hermite quadrature.

Other programs run on SAS that are not specifically supported by the SAS Institute. For further details about SAS for categorical data analyses, see the very helpful guide by Stokes et al. (2000). Also useful are SAS publications on logistic regression (Allison 1999) and graphics (Friendly 2000).

A.1.2 Other Software Packages

Most major statistical software has procedures for categorical data analyses. For instance, see SPSS (*SPSS Regression Models 10.0* by M. J. Norusis, SPSS Inc., 1999), Stata (*A Handbook of Statistical Analyses Using Stata*, 2nd ed., by S. Rabe-Hesketh and B. Everitt, CRC Press, Boca Raton, FL, 2000), S-Plus (*Modern Applied Statistics with S-Plus*, 3rd ed., by W. N. Venables and B. D. Ripley, Springer-Verlag, New York, 1999), and the related free package, R, and GLIM (Aitkin et al. 1989). Most major software now follows the lead of GLIM and includes a generalized linear models routine. Examples are PROC GENMOD in SAS and the glm function in R and S-Plus.

For certain analyses, specialized software is better than the major packages. A good example is StatXact (Cytel Software, Cambridge, Massachusetts), which provides exact analysis for categorical data methods and some nonparametric methods. Among its procedures are small-sample confidence intervals for differences and ratios of proportions and for odds ratios, and Fisher's exact test and its generalizations for $I \times J$ tables. It can also conduct exact tests of conditional independence and of equality of odds ratios in $2 \times 2 \times K$ tables, and exact confidence intervals for the common odds ratio in several 2×2 tables. StatXact uses Monte Carlo methods to approximate exact P-values and confidence intervals when a data set is too large for exact inference to be computationally feasible. Its companion, LogXact, performs exact conditional logistic regression.

Other examples of specialized software are SUDAAN for GEE-type analyses that handle clustering in survey data (Research Triangle Institute, Research Triangle Park, North Carolina), Latent GOLD for latent class modeling (Statistical Innovations, Belmont, Massachusetts), MLn (Institute of Education, London) and HLM (Scientific Software, Chicago) for multilevel models, and PASS for power analyses (NCSS Statistical Software, Kaysville, Utah). S-Plus and R functions are also available from individuals or from published work for particular analyses. For instance, *Statistical Models in S* by J. M. Chambers and T. J. Hastie (Wadsworth, Belmont, California, 1993, p. 227) showed the use of S-Plus in quasi-likelihood analyses using the quasi and make.family functions.

TABLE A.1 SAS Code for Chi-Squared, Measures of Association, and Residuals for Education–Religion Data in Table 3.2

```
data table;
    input degree religion $ count @@;
datalines;
1 fund 178      1 mod 138      1 lib 108
2 fund 570      2 mod 648      2 lib 442
3 fund 138      3 mod 252      3 lib 252
    ;
proc freq order=data; weight count;
  tables degree*religion/ chisq expected measures cmh1;
proc genmod order=data; class degree religion;
  model count=degree religion/ dist=poi link=log residuals;
```

A.2 EXAMPLES OF SAS CODE BY CHAPTER

The examples below show SAS code (Version 8.1). We focus on basic model fitting rather than the great variety of options. The material is organized by chapter of presentation. For convenience, data for examples are entered in the form of the contingency table displayed in the text. In practice, one would usually enter data at the subject level. These tables and the full data sets are available at *www.stat.ufl.edu/~aa/cda/cda.html.*

Chapters 1–3: Introduction, Two-Way Contingency Tables

Table A.1 uses SAS to analyze Table 3.2. The @@ symbol indicates that each line of data contains more than one observation. Input of a variable as characters rather than numbers requires an accompanying $ label in the INPUT statement. PROC FREQ forms the table with the TABLES statement, ordering row and column categories alphanumerically. To use instead the order in which the categories appear in the data set (e.g., to treat the variable properly in an ordinal analysis), use the ORDER = DATA option in the PROC statement. The WEIGHT statement is needed when one enters the cell counts instead of subject-level data. PROC FREQ can conduct chi-squared tests of independence (CHISQ option), show its estimated expected frequencies (EXPECTED), provide a wide assortment of measures of association and their standard errors (MEASURES), and provide ordinal statistic (3.15) with a "nonzero correlation" test (CMH1). One can also perform chi-squared tests using PROC GENMOD (using loglinear models discussed in the Chapters 8–9 section of this appendix), as shown. Its RESIDUALS option provides cell residuals. The output labeled "StReschi" is the standardized Pearson residual (3.13).

Table A.2 analyzes Table 3.8. With PROC FREQ, for 2 × 2 tables the MEASURES option in the TABLES statement provides confidence intervals

TABLE A.2 SAS Code for Fisher's Exact Test and Confidence Intervals for Odds Ratio for Tea-Tasting Data in Table 3.8

```
data fisher;
input poured guess count @@;
datalines;
1 1 3   1 2 1   2 1 1   2 2 3
;
proc freq;   weight count;
  tables poured*guess / measures riskdiff;
  exact fisher or / alpha = .05;
proc logistic descending; freq count;
  model guess = poured / clodds = pl;
```

for the odds ratio (labeled "case-control" on output) and the relative risk, and the RISKDIFF option provides intervals for the proportions and their difference. For tables having small cell counts, the EXACT statement can provide various exact analyses. These include Fisher's exact test and its generalization for $I \times J$ tables, treating variables as nominal, with keyword FISHER. The OR keyword gives the odds ratio and its large-sample confidence interval (3.2) and the small-sample interval based on (3.20). Other EXACT statement keywords include binomial tests for 1×2 tables (keyword BINOMIAL), exact trend tests for $I \times 2$ tables (TREND), and exact chi-squared tests (CHISQ) and exact correlation tests for $I \times J$ tables (MHCHI). One can use Monte Carlo simulation (option MC) to estimate exact P-values when the exact calculation is too time consuming. Table A.2 also uses PROC LOGISTIC to get a profile-likelihood confidence interval for the odds ratio (CLODDS = PL). LOGISTIC uses FREQ to serve the same purpose as PROC FREQ uses WEIGHT.

Other

StatXact provides small-sample confidence intervals for a binomial parameter, the difference of proportions, relative risk, and odds ratio. Blaker (2000) gave S-Plus functions that provide his confidence interval for a binomial parameter.

Chapter 4: Models for Binary Response Variables

PROC GENMOD fits GLMs. It specifies the response distribution in the DIST option ("poi" for Poisson, "bin" for binomial, "mult" for multinomial, "negbin" for negative binomial) and specifies the link in the LINK option. Table A.3 illustrates for Table 4.2. For binomial models with grouped data, the response in the model statements takes the form of the number of "successes" divided by the number of cases.

TABLE A.3 SAS Code for Binary GLMs for Snoring Data in Table 4.2

```
data glm;
input snoring disease total @@;
datalines;
0 24 1379   2 35 638   4 21 213   5 30 254
;
proc genmod; model disease / total = snoring / dist = bin link = identity;
proc genmod; model disease / total = snoring / dist = bin link = logit;
proc genmod; model disease / total = snoring / dist = bin link = probit;
```

TABLE A.4 SAS Code for Poisson and Negative Binomial GLMs for Horseshoe Crab Data in Table 4.3

```
data crab;
input color spine width satell weight;
datalines;
3  3  28.3  8  3.05
4  3  22.5  0  1.55
...
3  2  24.5  0  2.00
;
proc genmod;
  model satell = width / dist = poi link = log;
proc genmod;
  model satell = width / dist = poi link = identity;
proc genmod;
  model satell = width / dist = negbin link = identity;
```

Table A.4 uses GENMOD for count modeling of Table 4.3. Each observation refers to a single crab. Using width as the predictor, the first two models use Poisson regression. The third model uses the identity link assuming a negative binomial distribution.

Table A.5 uses GENMOD for the overdispersed data of Table 4.5. A CLASS statement requests dummy variables for the groups. With no intercept in the model (option NOINT) for the identity link, the estimated parameters are the four group probabilities. The ESTIMATE statement provides an estimate, confidence interval, and test for a contrast of model parameters, in this case the difference in probabilities for the first and second groups. The second analysis uses the Pearson statistic to scale standard errors to adjust for overdispersion. PROC LOGISTIC can also provide overdispersion modeling of binary responses; see Table A.27 in the Chapter 13 part of this appendix.

PROC GAM (starting in Version 8.2) fits generalized additive models.

TABLE A.5 SAS Code for Overdispersion Modeling of Teratology Data in Table 4.5

```
data moore;
  input litter group n y @@;
datalines;
 1 1 10 1     2 1 11 4     3 1 12 9    4 1 4 4    5 1 10 10
...
55 4 14 1    56 4  8 0    58 4 17 0
;
proc genmod;  class group;
  model y/n = group / dist = bin link = identity noint;
estimate ‘pi1- pi2 ’ group 1 -1 0 0;
proc genmod;  class group;
  model y/n = group / dist = bin link = identity noint scale = pearson;
```

Chapters 5 and 6: Logistic Regression

One can fit logistic regression models using either software for GLMs or specialized software for logistic regression. PROC GENMOD uses Newton-Raphson, whereas PROC LOGISTIC uses Fisher scoring. Both yield ML estimates, but SE values use observed information in GENMOD and expected information in LOGISTIC. These are the same for the logit link.

Table A.6 applies GENMOD and LOGISTIC to Table 5.2, when “y” out of “n” crabs had satellites at a given width level. In GENMOD, the LRCI option provides profile likelihood confidence intervals. The ALPHA = option can specify an error probability other than the default of 0.05. The TYPE3 option provides likelihood-ratio tests for each parameter. (In the Chapter 8−9 section we discuss the second GENMOD analysis.)

TABLE A.6 SAS Code for Modeling Grouped Crab Data in Table 5.2

```
data crab;
input width y n satell;  logcases = log(n);
datalines;
22.69  5 14 14
...
30.41 14 14 72
;
proc genmod;
  model y/n = width / dist = bin link = logit 1rci alpha = .01 type3;
proc logistic;
  model y/n = width / influence stb;
  output out = predict p = pi_hat lower = LCL upper = UCL;
proc print data = predict;
proc genmod;
  model satell = width / dist = poi link = log offset = logcases residuals;
```

TABLE A.7 SAS Code for Logit Modeling of AIDS Data in Table 5.5

```
data aids;
input race $ azt $ y n @@;
datalines;
 White Yes 14 107   White No 32 113   Black Yes 11 63   Black No 12 55
;
proc genmod; class race azt;
 model y/n = azt race / dist = bin type3 lrci residuals obstats;
proc logistic; class race azt / param = reference;
 model y/n = azt race / aggregate scale = none clparm = both clodds = both;
 output out = predict p = pi_hat lower = lower upper = upper;
proc print data = predict;
proc logistic; class race azt (ref = first) / param = ref;
 model y/n = azt / aggregate = (azt race) scale = none;
```

With PROC LOGISTIC, logistic regression is the default for binary data. LOGISTIC has a built-in check of whether logistic regression ML estimates exist. It can detect a complete separation of data points with 0 and 1 outcomes. LOGISTIC can also apply other links, such as the probit. Its INFLUENCE option provides Pearson and deviance residuals and diagnostic measures (Pregibon 1981). The STB option provides standardized estimates by multiplying by $s_{x_j}\sqrt{3}/\pi$ (Section 5.4.7 and Note 5.9). Following the model statement, Table A.6 requests predicted probabilities and lower and upper 95% confidence limits for the probabilities.

Table A.7 uses GENMOD and LOGISTIC to fit a logit model with qualitative predictors to Table 5.5. In GENMOD, the OBSTATS option provides various "observation statistics," including predicted values and their confidence limits. The RESIDUALS option requests residuals such as the Pearson and standardized Pearson residuals (labeled "Reschi" and "StReschi"). A CLASS statement requests dummy variables for the factor. By default, in GENMOD the parameter estimate for the last level of each factor equals 0. In LOGISTIC, estimates sum to zero. That is, dummies take the effect coding $(1, -1)$ of 1 when in the category and -1 when not, for which parameters sum to 0. In the CLASS statement in LOGISTIC, the option PARAM = REF requests $(1, 0)$ dummy variables with the last category as the reference level. Also putting REF = FIRST next to a variable name requests its first category as the reference level. The CLPARM = BOTH and CLODDS = BOTH options provide Wald and profile likelihood confidence intervals for parameters and odds ratio effects of explanatory variables. With AGGREGATE SCALE = NONE in the model statement, LOGISTIC reports Pearson and deviance tests of fit; it forms groups by aggregating data into the possible combinations of explanatory variable values, without overdispersion adjustments. Adding variables in parentheses after AGGREGATE (as in the second use of LOGISTIC in Table A.7) specifies the predictors used for forming the table on which to test fit, even when some predictors may have no effect in the model.

TABLE A.8 SAS Code for Logistic Regression Models with Horseshoe Crab Data in Table 4.3

```
data crab;
input color spine width satell weight;
if satell>0 then y = 1; if satell = 0 then y = 0;
if color = 4 then light = 0; if color<4 then light = 1;
datalines;
2 3 28.3 8 3.05
...
2 2 24.5 0 2.00
;
proc genmod descending; class color;
  model y = width color / dist = bin link = logit lrci type3 obstats;
  contrast 'a- d' color 1 0 0 -1;
proc genmod descending;
  model y = width color / dist = bin link = logit;
proc genmod descending;
  model y = width light / dist = bin link = logit;
proc genmod descending; class color spine;
  model y = width weight color spine / dist = bin link = logit type3;
proc logistic descending; class color spine / param = ref;
  model y = width weight color spine / selection = backward lackfit
      outroc = classif1;
proc plot data = classif1; plot _sensit_*_lmspec_ ;
```

Table A.8 shows logistic regression analyses for Table 4.3. The models refer to a constructed binary variable Y that equals 1 when a horseshoe crab has satellites and 0 otherwise. With binary data entry, GENMOD and LOGISTIC order the levels alphanumerically, forming the logit with $(1, 0)$ responses as $\log[P(Y = 0)/P(Y = 1)]$. Invoking the procedure with DESCENDING following the PROC name reverses the order. The first two GENMOD statements use both color and width as predictors; color is qualitative in the first model (by the CLASS statement) and quantitative in the second. A CONTRAST statement tests contrasts of parameters, such as whether parameters for two levels of a factor are identical. The statement shown contrasts the first and fourth color levels. The third GENMOD statement uses a dummy variable for color, indicating whether a crab is light or dark (color = 4). The fourth GENMOD statement fits the main effects model using all the predictors from Table 4.3. LOGISTIC has options for stepwise selection of variables, as the final model statement shows. The LACKFIT option yields the Hosmer−Lemeshow statistic. Using the OUTROC option, LOGISTIC can output a data set for plotting a ROC curve.

Table A.9 analyzes Table 6.9. The CMH option in PROC FREQ specifies the CMH statistic, the Mantel−Haenszel estimate of a common odds ratio and its confidence interval, and the Breslow−Day statistic. FREQ uses the

TABLE A.9 SAS Code for CMH Analysis of Clinical Trial Data in Table 6.9

```
data crab;
input center $ treat response count @@ ;
datalines;
a  1  1 11    a  1  2 25    a  2  1 10    a  2  2 27
...
h  1  1  4    h  1  2  2    h  2  1  6    h  2  2  1
;
proc freq; weight count;
  tables center*treat*response/ cmh chisq;
```

two rightmost variables in the TABLES statement as the rows and columns for each partial table; the CHISQ option yields chi-square tests of independence for each partial table. For $I \times 2$ tables the TREND keyword in the TABLES statement provides the Cochran–Armitage trend test.

Exact conditional logistic regression is available in PROC LOGISTIC with the EXACT statement. It provides ordinary and mid-P-values as well as confidence limits for each model parameter and the corresponding odds ratio with the ESTIMATE = BOTH option. One can also conduct the exact conditional version of the Cochran–Armitage test using the TREND option in the EXACT statement with PROC FREQ. Version 9 of SAS will include asymptotic conditional logistic regression, using a STRATA statement to indicate the stratification parameters to be conditioned out. One can also use PROC PHREG to do this (Stokes et al. 2000).

Models with probit and complementary log-log (CLOGLOG) links are available with PROC GENMOD, PROC LOGISTIC, or PROC PROBIT. O'Brien (1986) gave a SAS macro for computing powers using the noncentral chi-squared distribution.

Other

LogXact provides exact conditional logistic regression and StatXact provides exact inference about the odds ratio in $2 \times 2 \times K$ tables. PASS (NCSS Statistical Software) provides power analyses.

Chapter 7: Multinomial Response Models

PROC LOGISTIC fits baseline-category logit models (as of Version 8.2) using the LINK = GLOGIT option. The final response category is the default baseline for the logits. Exact inference is also available using the conditional distribution to eliminate nuisance parameters. PROC CATMOD also fits baseline-category logit models, as Table A.10 shows. CATMOD codes estimates for a factor so that they sum to zero. The PRED = PROB and PRED = FREQ options provide predicted probabilities and fitted values and their standard errors. The POPULATION statement provides the

TABLE A.10 SAS Code for Baseline-Category Logit Models with Alligator Data in Table 7.1

```
data gator;
input lake gender size food count @@;
datalines;
1 1 1 1 7  1 1 1 2 1  1 1 1 3 0  1 1 1 4 0  1 1 1 5 5
...
4 2 2 1 8  4 2 2 2 1  4 2 2 3 0  4 2 2 4 0  4 2 2 5 1
;
proc logistic; freq count; class lake size / param = ref;
  model food(ref = '1') = lake size / link = glogit
      aggregate scale = none;
proc catmod; weight count;
  population lake size gender;
  model food = lake size / pred = freq pred = prob;
```

variables that define the predictor settings. For instance, with "gender" in that statement, the model with lake and size effects is fitted to the full table also classified by gender.

PROC GENMOD can fit the proportional odds version of cumulative logit models using the DIST = MULTINOMIAL and LINK = CLOGIT options. Table A.11 fits it to Table 7.5. When the number of response categories exceeds 2, by default PROC LOGISTIC fits this model. It also gives a score test of the proportional odds assumption of identical effect parameters for each cutpoint. Both procedures use the $\alpha_j + \beta x$ form of the model. Cox (1995) used PROC NLIN for the more general model (7.8) having a scale parameter.

Both GENMOD and LOGISTIC can use other links in cumulative link models. GENMOD uses LINK = CPROBIT for the cumulative probit model and LINK = CCLL for the cumulative complementary log-log model. Table A.11 uses LINK = PROBIT in LOGISTIC to fit a cumulative probit model.

TABLE A.11 SAS Code for Cumulative Logit and Probit Models with Mental Impairment Data in Table 7.5

```
data impair;
input mental ses life;
datalines;
1 1 1
...
4 0 9
;
proc genmod ;
  model mental = life ses / dist = multinomial link = clogit lrci type3;
proc logistic;
  model mental = life ses / link = probit;
```

TABLE A.12 SAS Code for Adjacent-Categories Logit and Mean Response Models and CMH Analysis of Job Satisfaction Data in Table 7.8

```
data jobsat;
input gender income satisf count @@;
count2 = count + .01;
datalines;
1 1 1 1  1 1 2 3  1 1 3 11  1 1 4 2
...
0 4 1 0  0 4 2 1  0 4 3  9  0 4 4 6
;
proc catmod order = data; * ML analysis of adj-cat logit (ACL) model;
    weight count;
    population gender income;
    model satisf =
        (1 0 0  3 3,  0 1 0 2 2,  0 0 1 1 1,
         1 0 0  6 3,  0 1 0 4 2,  0 0 1 2 1,
         1 0 0  9 3,  0 1 0 6 2,  0 0 1 3 1,
         1 0 0 12 3,  0 1 0 8 2,  0 0 1 4 1,
         1 0 0  3 0,  0 1 0 2 0,  0 0 1 1 0,
         1 0 0  6 0,  0 1 0 4 0,  0 0 1 2 0,
         1 0 0  9 0,  0 1 0 6 0,  0 0 1 3 0,
         1 0 0 12 0,  0 1 0 8 0,  0 0 1 4 0)
          /ml pred = freq;
proc catmod order = data; weight count2; * WLS analysis of ACL model;
  response alogits; population gender income; direct gender income;
  model satisf = _response_ gender income;
proc catmod; weight count; * mean response model;
  population gender income; response mean; direct gender income;
  model satisf = gender income / covb;
proc freq; weight count;
  tables gender*income*satisf/ cmh scores = table;
```

One can fit adjacent-categories logit models in CATMOD by fitting equivalent baseline-category logit models. Table A.12 uses it for Table 7.8, where each line of code in the model statement specifies the predictor values (for the three intercepts, income, and gender) for the three logits. The income and gender predictor values are multiplied by 3 for the first logit, 2 for the second, and 1 for the third, to make effects comparable in the two models. PROC CATMOD has options (CLOGITS and ALOGITS) for fitting cumulative logit and adjacent-categories logit models to ordinal responses; however, those options provide weighted least squares (WLS) rather than ML fits. A constant must be added to empty cells for WLS to run. CATMOD treats zero counts as structural zeros, so they must be replaced by small constants when they are actually sampling zeros. The DIRECT statements identify predictors treated as quantitative. The second analysis in Table A.12 uses the ALOGITS option. CATMOD can also fit mean response models using WLS, as the third analysis in Table A.12 shows.

With the CMH option, PROC FREQ provides the generalized CMH tests of conditional independence. The statistic for the “general association” alternative treats X and Y as nominal [statistic (7.20)], the statistic for the "row mean scores differ" alternative treats X as nominal and Y as ordinal, and the statistic for the "nonzero correlation" alternative treats X and Y as ordinal [statistic (7.21)]. Table A.12 analyzes Table 7.8, using scores $(1, 2, 3, 4)$ for each variable.

PROC MDC fits multinomial discrete choice models, with logit and probit links. One can also use PROC PHREG, which is designed for the Cox proportional hazards model for survival analysis, because the partial likelihood for that analysis has the same form as the likelihood for the multinomial model (Allison 1999, Chap. 7; Chen and Kuo 2001).

Other

LogXact provides exact conditional analyses for baseline-category logit models. Joseph Lang (*jblang@stat.uiowa.edu*) has an R function that can fit mean response models by ML.

Chapters 8 and 9: Loglinear Models

Table A.13 uses GENMOD to fit model (AC, AM, CM) to Table 8.3. Table A.14 uses GENMOD for table raking of Table 8.15. Table A.15 uses GENMOD to fit the linear-by-linear association model (9.6) and the row effects model (9.8) to Table 9.3 (with column scores $1, 2, 4, 5$). The defined

TABLE A.13 SAS Code for Fitting Loglinear Models to Drug Survey Data in Table 8.3

```
data drugs;
input a c m count @@;
datalines;
1 1 1 911   1 1 2 538   1 2 1 44   1 2 2 456
2 1 1   3   2 1 2  43   2 2 1  2   2 2 2 279
;
proc genmod;  class a c m;
  model count = a c m a*m a*c c*m / dist = poi link = log lrci type3 obstats;
```

TABLE A.14 SAS Code for Raking Table 8.15

```
data rake;
input school atti count @@;
log_c = log(count); pseudo = 100 / 3;
data lines;
1 1 209   1 2 101   1 3 237
...
;
proc genmod; class school atti;
  model pseudo = school atti / dist = poi link = log offset = log_c obstats;
```

TABLE A.15 SAS Code for Fitting Association Models to GSS Data in Table 9.3

```
data sex;
input premar birth u v count @@; assoc = u*v ;
datalines;
1 1 1 1 38   1 2 1 2 60   1 3 1 4 68   1 4 1 5 81
...
;
proc genmod; class premar birth;
  model count = premar birth assoc / dist = poi link = log;
proc genmod; class premar birth;
  model  count = premar birth premar*v / dist = poi link = log;
```

variable “assoc” represents the cross-product of row and column scores, which has β parameter as coefficient in model (9.6). Table A.6 uses GENMOD to fit the Poisson regression model with log link for the grouped data of Table 5.2. It models the total number of satellites at each width level (variable “satell”), using the log of the number of cases as offset.

Correspondence analysis is available with PROC CORRESP.

Other

Prof. Joseph Lang (*jblang@stat.uiowa.edu*) has R and S-Plus functions for ML fitting of the generalized loglinear model (8.18). Becker (1990) gave a FORTRAN program that fits the $RC(M)$ model.

Chapter 10: Models for Matched Pairs

Table A.16 analyzes Table 10.1. For square tables, the AGREE option in PROC FREQ provides the McNemar chi-squared statistic for binary matched pairs, the X^2 test of fit of the symmetry model (also called *Bowker’s test*),

TABLE A.16 SAS Code for McNemar’s Test and Comparing Proportions for Matched Samples in Table 10.1

```
data matched;
input first second count @@;
datalines;
1 1 794   1 2 150   2 1 86   2 2 570
;
proc freq; weight count;
  tables first*second / agree;  exact mcnem;
proc catmod; weight count;
  response marginals;
  model first*second = (1  0 ,
                        1  1 ;
```

TABLE A.17 SAS Code for Testing Marginal Homogeneity with Migration Data in Table 10.6

```
data migrate;
input then $ now $ count  m11  m12  m13  m21  m22  m23  m31  m32  m33  m44  m1  m2  m3;
datalines;
  ne  ne 11607  1  0  0  0  0  0  0  0  0  0  0  0  0
  ne  mw   100  0  1  0  0  0  0  0  0  0  0  0  0  0
  ne   s   366  0  0  1  0  0  0  0  0  0  0  0  0  0
  ne   w   124 -1 -1 -1  0  0  0  0  0  0  0  1  0  0
  mw  ne    87  0  0  0  1  0  0  0  0  0  0  0  0  0
  mw  mw 13677  0  0  0  0  1  0  0  0  0  0  0  0  0
  mw   s   515  0  0  0  0  0  1  0  0  0  0  0  0  0
  mw   w   302  0  0  0 -1 -1 -1  0  0  0  0  0  1  0
   s  ne   172  0  0  0  0  0  0  1  0  0  0  0  0  0
   s  mw   225  0  0  0  0  0  0  0  1  0  0  0  0  0
   s   s 17819  0  0  0  0  0  0  0  0  1  0  0  0  0
   s   w   270  0  0  0  0  0  0 -1 -1 -1  0  0  0  1
   w  ne    63 -1  0  0 -1  0  0 -1  0  0  0  1  0  0
   w  mw   176  0 -1  0  0 -1  0  0 -1  0  0  0  1  0
   w   s   286  0  0 -1  0  0 -1  0  0 -1  0  0  0  1
   w   w 10192 0  0  0  0  0  0  0  0  0  1  0  0  0
       ;
proc genmod;
  model count = m11  m12  m13  m21  m22  m23  m31  m32  m33  m44  m1  m2  m3
      / dist = poi  link = identity;
proc catmod;  weight count;  response marginals;
  model then*now = _response_ /freq;
  repeated time 2;
```

and Cohen's kappa and weighted kappa with SE values. The MCNEM keyword in the EXACT statement provides a small-sample binomial version of McNemar's test. PROC CATMOD can provide the confidence interval for the difference of proportions. The code forms a model for the marginal proportions in the first row and the first column, specifying a model matrix in the model statement that has an intercept parameter (the first column) that applies to both proportions and a slope parameter that applies only to the second; hence the second parameter is the difference between the second and first marginal proportions.

PROC LOGISTIC can conduct conditional logistic regression.

Table A.17 shows ways of testing marginal homogeneity for Table 10.6. The GENMOD code shows the Lipsitz et al. (1990) approach, expressing the I^2 expected frequencies in terms of parameters for the $(I - 1)^2$ cells in the first $I - 1$ rows and $I - 1$ columns, the cell in the last row and last column, and $I - 1$ marginal totals (which are the same for rows and columns). Here, m11 denotes expected frequency μ_{11}, m1 denotes $\mu_{1+} = \mu_{+1}$, and so on. This parameterization uses formulas such as $\mu_{14} = \mu_{1+} - \mu_{11} - \mu_{12} - \mu_{13}$ for terms in the last column or last row. CATMOD provides the Bhapkar test (10.16) of marginal homogeneity, as shown.

TABLE A.18 SAS Code Showing Square-Table Analysis of Table 10.5

```
data sex;
input premar extramar symm qi count @@;
unif = premar*extramar;
datalines;
1 1 1 1 144    1 2 2 5  2    1 3 3 5  0    1 4  4 5 0
2 1 2 5  33    2 2 5 2  4    2 3 6 5  2    2 4  7 5 0
3 1 3 5  84    3 2 6 5 14    3 3 8 3  6    3 4  9 5 1
4 1 4 5 126    4 2 7 5 29    4 3 9 5 25    4 4 10 4 5
;
proc genmod; class symm;
  model count = symm / dist = poi  link = log; * symmetry;
proc genmod; class extramar premar symm;
  model count = symm extramar premar / dist = poi  link = log; *QS;
proc genmod; class symm;
  model count = symm extramar premar / dist = poi  link = log; * ordinal QS;
proc genmod; class extramar premar qi;
  model count = extramar premar qi / dist = poi  link = log; * quasi indep;
proc genmod; class extramar premar;
  model count = extramar premar unif / dist = poi  link = log;
data sex2;
input score below above @@; trials = below + above;
datalines;
1 33 2   1 14 2   1 25 1   2 84 0   2 29 0   3 126 0
;
proc genmod data = sex2;
  model above / trials = score / dist = bin link = logit noint;
 proc genmod data = sex2;
  model above / trials = /dist = bin  link = logit noint;
proc genmod data = sex2;
  model above / trials = /dist = bin  link = logit;
```

Table A.18 shows various square-table analyses of Table 10.5. The "symm" factor indexes the pairs of cells that have the same association terms in the symmetry and quasi-symmetry models. For instance, "symm" takes the same value for cells $(1, 2)$ and $(2, 1)$. Including this term as a factor in a model invokes a parameter λ_{ij} satisfying $\lambda_{ij} = \lambda_{ji}$. The first model fits this factor alone, providing the symmetry model. The second model looks like the third except that it identifies "premar" and "extramar" as class variables (for quasi-symmetry), whereas the third model statement does not (for ordinal quasi-symmetry). The fourth model fits quasi-independence. The "qi" factor invokes the δ_i parameters. It takes a separate level for each cell on the main diagonal and a common value for all other cells. The fifth model fits the quasi-uniform association model (10.29).

The bottom of Table A.18 fits square-table models as logit models. The pairs of cell counts (n_{ij}, n_{ji}), labeled as "above" and "below" with reference to the main diagonal, are six sets of binomial counts. The variable defined as "score" is the distance $(u_j - u_i) = j - i$. The first two cases are symmetry

TABLE A.19 SAS Code for Fitting Bradley–Terry Model to Table 10.10

```
data baseball;
input  wins  games  milw  detr  toro  newy  bost  clev  balt;
datalines;
7  13  1 -1  0  0  0  0  0
...
6  13  0  0  0  0  0  1 -1
;
proc genmod;
  model  wins/games=milw  detr  toro  newy  bost  clev  balt /
  dist=bin  link=logit noint covb;
```

and ordinal quasi-symmetry. Neither model contains an intercept (NOINT), and the ordinal model uses “score” as the predictor. The third model allows an intercept and is the conditional symmetry model (10.28).

Table A.19 uses GENMOD for logit fitting of the Bradley–Terry model to Table 10.10 by forming an artificial explanatory variable for each team. For a given observation, the variable for team i is 1 if it wins, -1 if it loses, and 0 if it is not one of the teams for that match. Each observation lists the number of wins (“wins”) for the team with variate-level equal to 1 out of the number of games (“games”) against the team with variate-level equal to -1. The model has these artificial variates, one of which is redundant, as explanatory variables with no intercept term. The COVB option provides the estimated covariance matrix of parameter estimators.

Chapter 11: Analyzing Repeated Categorical Response Data

Table A.20 uses GENMOD for the likelihood-ratio test of marginal homogeneity for Table 11.1, where for instance *m11p* denotes μ_{11+}. The marginal homogeneity model expresses the eight cell expected frequencies in terms of

TABLE A.20 SAS Code for Testing Marginal Homogeneity with Crossover Study of Table 11.1

```
data crossover;
input  a  b  c  count  m111  m11p  m1p1  mp11  m1pp  m222  @@;
datalines;
1  1  1  6   1  0  0  0  0  0   1  1  2  16 -1  1  0  0  0  0
1  2  1  2  -1  0  1  0  0  0   1  2  2   4  1 -1 -1  0  1  0
2  1  1  2  -1  0  0  1  0  0   2  1  2   4  1 -1  0 -1  1  0
2  2  1  6   1  0 -1 -1  1  0   2  2  2   6  0  0  0  0  0  1
;
proc genmod;
  model  count=m111  m11p  m1p1  mp11  m1pp  m222/dist=poi  link=identity;
proc catmod;  weight  count;  response  marginals;
  model  a*b*c=_response_ /freq;
  repeated  drug  3;
```

TABLE A.21 SAS Code for Marginal Modeling of Depression Data in Table 11.2

```
data depress;
input  case  diagnose  drug  time  outcome  @@;  *  outcome = 1 is normal;
datalines;
  1  0  0  0  1    1  0  0  1  1    1  0  0  2  1
...
340  1  1  0  0  340  1  1  1  0  340  1  1  2  0
;
proc genmod descending;  class case;
  model; outcome = diagnose  drug  time  drug*time / dist = bin  link = logit  type3;
  repeated  subject = case / type = exch  corrw;
proc n1mixed  qpoints = 200;
  parms alpha = -.03  beta1 = -1.3  beta2 = -.06  beta3 = .48  beta4 = 1.02  sigma = .066;
  eta = alpha + beta1*diagnose  + beta2*drug  + beta3*time  + beta4*drug*time  + u;
  p = exp(eta) / (1 + exp(eta));
  model  outcome ~ binary(p);
  random  u ~ normal(0,  sigma*sigma)   subject = case;
```

TABLE A.22 SAS Code for GEE and Random Intercept Cumulative Logit Analysis of Insomnia Data in Table 11.4

```
data francom;
  input  case  treat  time  outcome  @@;
datalines;
  1  1  0  1    1  1  1  1
...
239  0  0  4  239  0  1  4
;
proc genmod;  class case;
  model outcome = treat time treat*time / dist = multinomial
      link = clogit;
  repeated subject = case / type = indep corrw;
proc n1mixed  qpoints = 40;
  bounds  i2>0;  bounds i3>0;
  eta1 = i1 + treat*beta1 + time*beta2 + treat*time*beta3 + u;
  eta2 = i1 + i2 + treat*beta1 + time*beta2 + treat*time*beta3 + u;
  eta3 = i1 + i2 + i3 + treat*beta1 + time*beta2 + treat*time*beta3 + u;
  p1 = exp(eta) / (1 + exp(eta1));
  p2 = exp(eta2) / (1 + exp(eta2)) - exp(eta1) / (1 + exp(eta1));
  p3 = exp(eta3) / (1 + exp(eta3)) - exp(eta2) / (1 + exp(eta2));
  p4 = 1 - exp(eta3) / (1 + exp(eta3));
  11 = y1*log(p1) + y2*log(p2) + y3*log(p3) + y4*log(p4);
  model y1 ~ general(11);
  estimate 'interc2 ' i1 + i2; * this is alpha_2 in model, and
      i1 is alpha_1;
  estimate 'interc3 ' i1 + i2 + i3; * this is alpha_3 in model;
  random u ~ normal(0, sigma*sigma) subject = case;
```

μ_{111}, μ_{11+}, μ_{1+1}, μ_{+11}, μ_{1++}, and μ_{222} (since $\mu_{+1+} = \mu_{++1} = \mu_{1++}$). Note, for instance, that $\mu_{112} = \mu_{11+} - \mu_{111}$ and $\mu_{122} = \mu_{111} + \mu_{1++} - \mu_{11+} - \mu_{1+1}$. CATMOD provides the generalized Bhapkar test (11.5) of marginal homogeneity.

Table A.21 uses GENMOD to analyze Table 11.2 using GEE. Possible working correlation structures are TYPE = EXCH for exchangeable, TYPE = AR for autoregressive, TYPE = INDEP for independence, and TYPE = UNSTR for unstructured. Output shows estimates and standard errors under the naive working correlation and based on the sandwich matrix incorporating the empirical dependence. Alternatively, the working association structure in the binary case can use the log odds ratio (e.g., using LOGOR = EXCH for exchangeability). The type 3 option in GEE provides score tests about effects. See Stokes et al. (2000, Sec. 15.11) for the use of GEE with missing data.

Table A.22 uses GENMOD to implement GEE for a cumulative logit model for Table 11.4. For multinomial responses, independence is currently the only working correlation structure.

Other

Joseph Lang (*jblang@stat.uiowa.edu*) has R and S-Plus functions for ML fitting of marginal models through the generalized loglinear model (11.8), using the constraint approach with Lagrange multipliers. The program MAREG (Kastner et al. 1997) provides GEE fitting and ML fitting of marginal models with the Fitzmaurice and Laird (1993) approach, allowing multicategory responses. See *www.stat.uni-muenchen.de/~andreas/mareg/winmareg.html.*

Chapter 12: Random Effects: Generalized Linear Mixed Models

PROC NLMIXED extends GLMs to GLMMs by including random effects. Table A.23 analyzes the matched pairs model (12.3). Table A.24 analyzes the election data in Table 12.2.

TABLE A.23 SAS Code for Fitting Model (12.3) for Matched Pairs to Table 12.1

```
data matched;
input case occasion response count @@;
datalines;
2  0  1 794    1  1  1 794    2  0  1 150    2  1  0 150
3  0  0  86    3  1  1  86    4  0  0 570    4  1  0 570
;
proc n1mixed;
  eta = alpha + beta*occasion + u;  p = exp(eta) / (1 + exp(eta));
  model  response ~ binary(p);
  random u ~ normal(0, sigma*sigma) subject = case;
  replicate count;
```

TABLE A.24 SAS Code for GLMM Analysis of Election Data in Table 12.2

```
data vote;
input y n;
case = _n_;
datalines;
 1   5
16  32
...
 1   4
;
proc n1mixed;
  eta = alpha + u;  p = exp(eta) / (1 + exp(eta));
  model y ~ binomial(n,p);
  random u ~ normal (0, sigma*sigma) subject = case;
  predict p out = new;
proc print data = new;
```

TABLE A.25 SAS Code for GLMM Modeling of Opinions in Table 10.13

```
data new;
input sex poor single any count;
datalines;
1  1  1  1 342
...
2  0  0  0 457
;
data new;  set new;
  sex = sex- 1;  case = _n_;
  q1 = 1; q2 = 0; resp = poor; output;
  q1 = 0, q2 = 1; resp = single;  output;
  q1 = 0; q2 = 0; resp = any; output;
drop poor single any;
proc n1mixed  qpoints = 50;
  parms alpha = 0 beta1 = .8 beta2 = .3 gamma = 0 sigma = 8.6;
  eta = alpha + beta1*q1 + beta2*q2 + gamma*sex + u;
  p = exp(eta) / (1 + exp(eta));
  model resp ~ binary(p);
  random u ~ normal(0, sigma*sigma) subject = case;
  replicate count;
```

TABLE A.26 SAS Code for GLMM for Leading Crowd Data in Table 12.8

```
data crowd;
input  mem1  att1  mem2  att2  count;
datalines;
  1  1  1  1 458
...
  0  0  0  0 554
;
data new;  set crowd;
  case = _n_;
  x1m = 1;  x1a = 0;  x2m = 0;  x2a = 0;  var = 1;  resp = mem1;  output;
  x1m = 0;  x1a = 1;  x2m = 0;  x2a = 0;  var = 0;  resp = att1;  output;
  x1m = 0;  x1a = 0;  x2m = 1;  x2a = 0;  var = 1;  resp = mem2;  output;
  x1m = 0;  x1a = 0;  x2m = 0;  x2a = 1;  var = 0;  resp = att2; output;
  drop  mem1  att1  mem2  att2;
proc  nlmixed  data = new;
  eta = beta1m*x1m + beta1a*x1a + beta2m*x2m + beta2a*x2a + um*var +
     ua*(1-var);
  p = exp(eta) / (1 + exp(eta));
  model resp ~ binary(p);
  random  um  ua ~ normal([0,0],[s1*s1, cov12, s2*s2]) subject = case;
  replicate count;
  estimate 'mem change' beta2m-beta1m; estimate 'att change'
     beta2a-beta1a;
```

Table A.25 fits model (12.10) to Table 10.13. This shows how to set initial values and set the number of quadrature points for Gauss–Hermite quadrature (e.g., QPOINTS =). One could let SAS fit without initial values but then take that fit as initial values in further runs, increasing QPOINTS until estimates and standard errors converge to the necessary precision.

Table A.21 uses NLMIXED for Table 11.2. Table A.22 uses NLMIXED for ordinal modeling of Table 11.4, defining a general multinomial log likelihood. Table A.26 shows a correlated bivariate random effect analysis of Table 12.8. Agresti et al. (2000) showed NLMIXED examples for clustered data, Agresti and Hartzel (2000) showed code for multicenter trials such as Table 12.5, and Hartzel et al. (2001a) showed code for multicenter trials with an ordinal response. The Web site for the journal *Statistical Modelling* shows NLMIXED code for an adjacent-categories logit model and a nominal model at the data archive for Hartzel et al. (2001b). Chen and Kuo (2001) discussed fitting multinomial logit models, including discrete-choice models, with random effects.

Other

MLn (Institute of Education, London) and HLM (Scientific Software, Chicago) fit multilevel models. MIXOR is a FORTRAN program for ML

TABLE A.27 SAS Code for Overdispersion Analysis of Table 4.5

```
data moore;
input  litter  group  n  y  @@;
  z2=0;  z3=0;  z4=0;
  if group=2 then z2=1;  if group=3 then z3=1;  if group=4
      then z4=1;
datalines;
 1  1 10  1     2  1 11  4     3  1 12  9     4  1  4  4
...
55  4 14  1    56  4  8  0    57  4  6  0    58  4 17  0
;
proc logistic;
  model y/n=z2 z3 z4/scale=williams;
proc logistic;
  model y/n=z2 z3 z4/scale=pearson;
proc n1mixed  qpoints=200;
  eta=alpha+beta2*z2+beta3*z3+beta4*z4+u;
  p=exp(eta)/(1+exp(eta));
  model y ~ binomial(n,p);
  random u ~ normal(0, sigma*sigma) subject=litter;
```

TABLE A.28 SAS Code for Fitting Models to Murder Data in Table 13.6

```
data new;
input white black other response;
datalines;
1070  119  55   0
  60   16   5   1
...
   1    0   0   6
;
data new; set new; count=white; race=0; output;
  count=black; race=1; output; drop white black other;
data new2; set new; do i=1 to count; output; end; drop i;
proc genmod data=new2;
  model response=race/dist=negbin link=log;
proc genmod data=new2;
  model response=race/dist=poi  link=log scale=pearson;
data new; set new; case=_n_;
proc n1mixed data=new  qpoints=400;
  parms alpha=-3.7 beta=1.90 sigma=1.6;
  eta=alpha+beta*race+u; mu=exp(eta);
  model response ~ poisson(mu);
  random u ~ normal(0, sigma*sigma) subject=case;
  replicate count;
```

fitting of binary and ordinal random effects models available from Don Hedeker (*www.uic.edu/~hedeker/mix.html*).

Chapter 13: Other Mixture Models for Categorical Data

PROC LOGISTIC provides two overdispersion approaches for binary data. The SCALE = WILLIAMS option uses variance function of the beta-binomial form (13.10), and SCALE = PEARSON uses the scaled binomial variance (13.11). Table A.27 illustrates for Table 4.5. That table also uses NLMIXED for adding litter random intercepts.

For Table 13.6, Table A.28 uses GENMOD to fit a negative binomial model and a quasi-likelihood model with scaled Poisson variance using the Pearson statistic, and NLMIXED to fit a Poisson GLMM. PROC NLMIXED can also fit negative binomial models.

Other

Latent GOLD (developed by J. Vermunt and J. Magidson for Statistical Innovations, Belmont, Massachusetts) can fit a wide variety of mixture models, including latent class models, nonparametric mixtures of logistic regression, and some Rasch mixture models.

Categorical Data Analysis, Second Edition. Alan Agresti
Copyright © 2002 John Wiley & Sons, Inc.
ISBN: 0-471-36093-7

APPENDIX B

Chi-Squared Distribution Values

	Right-Tailed Probability						
df	0.250	0.100	0.050	0.025	0.010	0.005	0.001
1	1.32	2.71	3.84	5.02	6.63	7.88	10.83
2	2.77	4.61	5.99	7.38	9.21	10.60	13.82
3	4.11	6.25	7.81	9.35	11.34	12.84	16.27
4	5.39	7.78	9.49	11.14	13.28	14.86	18.47
5	6.63	9.24	11.07	12.83	15.09	16.75	20.52
6	7.84	10.64	12.59	14.45	16.81	18.55	22.46
7	9.04	12.02	14.07	16.01	18.48	20.28	24.32
8	10.22	13.36	15.51	17.53	20.09	21.96	26.12
9	11.39	14.68	16.92	19.02	21.67	23.59	27.88
10	12.55	15.99	18.31	20.48	23.21	25.19	29.59
11	13.70	17.28	19.68	21.92	24.72	26.76	31.26
12	14.85	18.55	21.03	23.34	26.22	28.30	32.91
13	15.98	19.81	22.36	24.74	27.69	29.82	34.53
14	17.12	21.06	23.68	26.12	29.14	31.32	36.12
15	18.25	22.31	25.00	27.49	30.58	32.80	37.70
16	19.37	23.54	26.30	28.85	32.00	34.27	39.25
17	20.49	24.77	27.59	30.19	33.41	35.72	40.79
18	21.60	25.99	28.87	31.53	34.81	37.16	42.31
19	22.72	27.20	30.14	32.85	36.19	38.58	43.82
20	23.83	28.41	31.41	34.17	37.57	40.00	45.32
25	29.34	34.38	37.65	40.65	44.31	46.93	52.62
30	34.80	40.26	43.77	46.98	50.89	53.67	59.70
40	45.62	51.80	55.76	59.34	63.69	66.77	73.40
50	56.33	63.17	67.50	71.42	76.15	79.49	86.66
60	66.98	74.40	79.08	83.30	88.38	91.95	99.61
70	77.58	85.53	90.53	95.02	100.4	104.2	112.3
80	88.13	96.58	101.8	106.6	112.3	116.3	124.8
90	98.65	107.6	113.1	118.1	124.1	128.3	137.2
100	109.1	118.5	124.3	129.6	135.8	140.2	149.5

Source: Calculated using *StaTable*, Cytel Software, Cambridge, MA.