A.1 SAS EXAMPLES

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, GLIMMIX, and CATMOD. PROC FREQ performs basic analyses for two-way and three-way contingency tables. PROC GENMOD fits generalized linear models using ML or Bayesian methods, cumulative link models for ordinal responses, zero-inflated Poisson regression models for count data, and GEE analyses for marginal models. PROC LOGISTIC gives ML fitting of binary response models, cumulative link models for ordinal responses, and baseline-category logit models for nominal responses. (PROC SURVEYLOGISTIC fits binary and multi-category regression models to survey data by incorporating the sample design into the analysis and using the method of pseudo ML.) PROC CATMOD fits baseline-category logit models and can fit a variety of other models using weighted least squares. PROC NLMIXED gives ML fitting of generalized linear mixed models, using adaptive Gauss–Hermite quadrature. PROC GLIMMIX also fits such models with a variety of fitting methods.

The examples in this appendix show SAS code for version 9.3. We focus on basic model fitting rather than the great variety of options. For more detail, see

Stokes, Davis, and Koch (2012) *Categorical Data Analysis Using SAS*, 3rd ed. Cary, NC: SAS Institute.

Allison (2012) *Logistic Regression Using SAS: Theory and Application,* 2nd edition. Cary, NC: SAS Institute.

For examples of categorical data analyses with SAS for many data sets in my text *An Introduction to Categorical Data Analysis*, see the useful site

`www.ats.ucla.edu/stat/examples/icda/`

set up by the UCLA Statistical Computing Center. A useful SAS site on-line with details about the options as well as many examples for each PROC is at

`support.sas.com/rnd/app/da/stat/procedures/CategoricalDataAnalysis.html`.

In the SAS code below, The @@ symbol in an input line 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. (But, of course, if you are already a SAS user, you know this and much more!)

Chapter 1: Introduction

With PROC FREQ for a 1×2 table of counts of successes and failures for a binomial variate, confidence limits for the binomial proportion include Agresti–Coull, Jeffreys (i.e., Bayes with beta(0.5, 0.5) prior), score (Wilson), and Clopper–Pearson exact method. The keyword BINOMIAL and the EXACT statement yields binomial tests. Table 1 shows code for confidence intervals for the example in the text Section 1.4.3 about estimating the proportion of people who are vegetarians, when 0 of 25 people in a sample are vegetarian. The AC option gives the Agresti–Coull interval, and the WILSON option gives the score-test-based interval.

Table 1: **SAS Code for Confidence Intervals for a Proportion**

```
-------------------------------------------------------------------------
data veg;
input response $ count;
datalines;
no   25
yes  0
;
proc freq data=veg; weight count;
tables response / binomial(ac wilson exact jeffreys) alpha=.05;
run;
-------------------------------------------------------------------------
```

Chapters 2–3: Two-Way Contingency Tables

Table 2 uses SAS to analyze Table 3.2 in *Categorical Data Analysis*, on education and belief in God. 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 you enter the cell counts from the contingency table instead of subject-level data. PROC FREQ can conduct Pearson and likelihood-ratio 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 (3.16) with a "nonzero correlation" test (CMH1). You can also perform chi-squared tests using PROC GENMOD (using loglinear models discussed in Chapters 9-10), as shown. Its RESIDUALS option provides cell residuals. The output labeled "Std Pearson Residual" is the standardized residual.

For creating mosaic plots in SAS, see `www.datavis.ca` and `www.datavis.ca/books/vcd/`.

With PROC FREQ, for 2×2 tables the MEASURES option in the TABLES statement provides confidence intervals 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. Using RISKDIFF(CL=(MN)) gives the interval based on inverting a score test, as suggested by Miettinen and Nurminen (1985), which is much preferred over a Wald interval. See Table 3 for an example.

For tables having small cell counts, the EXACT statement can provide various exact analyses. These include Fisher's exact test (with its two-sided P-value based on the sum of the probabilities that are no greater than the probability of the observed table) and its generalization for $I \times J$ tables, treating variables as nominal, with keyword FISHER. Table 4 analyzes the tea tasting data in Table 3.9 of the textbook. Table 4 also uses PROC LOGISTIC to get a profile-likelihood confidence interval for the odds ratio (CLODDS = PL), viewing the odds ratio as a parameter in a simple logistic regression model with a binary indicator as a predictor. PROC LOGISTIC

Table 2: **SAS Code for Chi-Squared, Measures of Association, and Residuals for Data on Education and Belief in God in Table 3.2**

```
-------------------------------------------------------------------
data table;
    input degree belief $ count @@;
datalines;
1 1  9   1 2  8   1 3 27   1 4  8   1 5  47   1 6 236
2 1 23   2 2 39   2 3 88   2 4 49   2 5 179   2 6 706
3 1 28   3 2 48   3 3 89   3 4 19   3 5 104   3 6 293
  ;
proc freq  order=data; weight count;
  tables degree*belief / chisq expected measures cmh1;
proc genmod  order=data;  class degree belief;
    model count = degree belief / dist=poi link=log residuals;
run;
-------------------------------------------------------------------
```

Table 3: **SAS Code for Confidence Intervals for 2×2 Table**

```
-----------------------------------------------------------------------
data example;
input group $ outcome $ count @@;
datalines;
placebo yes 2  placebo no 18  active yes 7 active no 13
;
proc freq order=data; weight count;
    tables group*outcome / riskdiff(CL=(WALD MN)) measures;
      * MN = Miettinen and Nurminen inverted score test;
run;
-----------------------------------------------------------------------
```

uses FREQ to weight counts, serving the same purpose for which PROC FREQ uses WEIGHT.

The OR keyword gives the odds ratio and its large-sample Wald confidence interval based on (3.2) and the small-sample interval based on the noncentral hypergeometric distribution (16.28). Other EXACT statement keywords include unconditional exact confidence limits for the difference of proportions (for the keyword RISKDIFF), 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). With keyword RISKDIFF in the EXACT statement, SAS seems to construct an exact unconditional interval due to Santner and Snell in 1980 (*JASA*) that is very conservative. Version 9.3 includes the option RISKDIFF(METHOD=FMSCORE), which seems to be based on the Chan

Table 4: **SAS Code for Fisher's Exact Test and Confidence Intervals for Odds Ratio for Tea-Tasting Data in Table 3.9**

```
-----------------------------------------------------------------------
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;
run;
-----------------------------------------------------------------------
```

and Zhang (1999) approach of inverting two separate one-sided score tests, which is less conservative. (Note: FM stands for Farrington-Manning, which is an incorrect attribution by SAS for the score test and the score CI.) See `http://support.sas.com/documentation/cdl/en/procstat/63963/HTML/default/viewer.htm#procstat_freq_a0000000564.htm#procstat.freq.freqrdiffexact` for details and Table 5 for an example for a 2×2 table. (The software StatXact also provides the Agresti and Min (2001) method of inverting a single two-sided score test, which is less conservative yet.) You can use Monte Carlo simulation (option MC) to estimate exact P-values when the exact calculation is too time-consuming.

Chapter 4: Generalized Linear Models

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 function in the LINK option. 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 6 illustrates for the snoring data in Table 4.2 of the textbook. Profile likelihood confidence intervals are provided in PROC GENMOD with the LRCI option.

Table 7 uses PROC GENMOD for count modeling of the horseshoe crab data in Table 4.3 of the textbook. (Note that the complete data set is in the Datasets link `www.stat.ufl.edu/~aa/cda/data.html` at this website.) The variable SATELL in the data set refers to the number of satellites that a female horseshoe crab has. Each observation refers to a single crab. Using width as the predictor, the first two models use Poisson regression and the third model assumes a negative binomial distribution.

Table 8 uses PROC GENMOD for the overdispersed teratology-study data of Table 4.7 of the textbook. (Note that the complete data set is in the Datasets link `www.stat.ufl.edu/~aa/cda/data.html` at this website.) A CLASS statement requests

Table 5: **SAS Code for "Exact" Confidence Intervals for 2×2 Table**

```
-------------------------------------------------------------------------
data example;
input group $ outcome $ count @@;
datalines;
placebo yes 2  placebo no 18  active yes 7 active no 13
;
proc freq order=data; weight count;
    tables group*outcome ;
exact or riskdiff ;  * conservative Santner-Snell approach;
run;
proc freq order=data; weight count;
    tables group*outcome ;
exact riskdiff(method=fmscore);
   * exact unconditional inverting two one-sided score tests;
run;
-------------------------------------------------------------------------
```

Table 6: **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 LRCI;
proc genmod; model disease/total = snoring  /  dist=bin  link=probit;
run;
-------------------------------------------------------------------------
```

indicator (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 35 in the Chapter 14 part of this appendix for SAS.

The final PROC GENMOD run in Table 10 fits the Poisson regression model with log link for the grouped data of Tables 4.4 and 5.2. It models the total number of

Table 7: **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 ;
run;
-------------------------------------------------------------
```

Table 8: **SAS Code for Overdispersion Modeling of Teratology Data in Table 4.7**

```
-----------------------------------------------------------------------
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   57 4  6 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;
run;
-----------------------------------------------------------------------
```

satellites at each width level (variable "satell"), using the log of the number of cases as offset.

Chapters 5–7: Logistic Regression and Binary Response Analyses

You 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 the SE values use the inverted observed information matrix in PROC GENMOD and the inverted expected information matrix in PROC LOGISTIC. These are the same for the logit link because it is the canonical link function for the binomial, but differ for other links. With PROC LOGISTIC, logistic regression is the default for binary data. PROC LOGISTIC has a built-in check of whether logistic regression ML estimates exist. It can detect complete separation of data points with 0 and 1 outcomes, in which case at least one estimate is infinite. PROC LOGISTIC can also apply other links, such as the probit.

Table 9 shows logistic regression analyses for the horseshoe crab data of Table 4.3. The variable SATELL in the data set at `www.stat.ufl.edu/~aa/cda/data.html` refers to the number of satellites that a female horseshoe crab has. The logistic models refer to a constructed binary variable Y that equals 1 when a horseshoe crab has satellites and 0 otherwise. With binary data entry, PROC GENMOD and PROC 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 PROC 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 PROC GENMOD statement uses an indicator variable for color, indicating whether a crab is light or dark (color = 4). The fourth PROC GENMOD statement fits the main effects model using all the predictors. PROC LOGISTIC has options for stepwise selection of variables, as the final model statement shows. The LACKFIT option yields the Hosmer–Lemeshow statistic. Predicted probabilities and lower and upper 95% confidence limits for the probabilities are shown in a plot with the PLOTS=ALL option or with PLOTS=EFFECT. The ROC curve is produced using the PLOTS=ROC option.

Table 10 applies PROC GENMOD and PROC LOGISTIC to the grouped data as shown in Table 5.2 of the textbook, when "y" out of "n" crabs had satellites at a given width level. Profile likelihood confidence intervals are provided in PROC GENMOD with the LRCI option and in PROC LOGISTIC with the PLCL option. In PROC GENMOD, the ALPHA = option can specify an error probability other than the default of 0.05, and the TYPE3 option provides likelihood-ratio tests for each parameter. In PROC LOGISTIC, the INFLUENCE option provides Pearson and deviance residuals and diagnostic measures (Pregibon 1981). The STB option provides standardized estimates by multiplying by $s_{x_j}\sqrt{3}/\pi$ (text Section 5.4.7). Following the model statement, Table 10 requests predicted probabilities and lower and upper 95% confidence limits for the probabilities. In the PLOTS option, the standardized residuals are plotted against the linear predictor values. (In the Chapter 9–10 section we discuss the second GENMOD analysis of a loglinear model.)

Table 11 uses PROC GENMOD and PROC LOGISTIC to fit a logistic model with

Table 9: **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
4  3  22.5  0  1.55
...
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;
ods graphics on;
ods html;
proc logistic descending plots=all;
  class color spine / param=ref;
  model y = width weight color spine / selection=backward lackfit;
run;
ods html close;
ods graphics off;
run;
-------------------------------------------------------------------------
```

Table 10: **SAS Code for Modeling Grouped Crab Data in Tables 4.4 and 5.2**

```
-------------------------------------------------------------------------
data crab;
input width y n satell;  logcases=log(n);
datalines;
22.69  5 14  14
...
30.41 14 14  72
;
ods graphics on;
ods html;
proc genmod;
  model y/n = width / dist=bin link=logit lrci alpha=.01 type3;
proc genmod plots=stdreschi(xbeta);;
  model satell = width / dist=poi link=log offset=logcases residuals;
proc logistic data=crab plots=all;
  model y/n = width / influence stb;
  output out=predict  p=pi_hat  lower=LCL  upper=UCL;
proc print  data=predict;
run;
ods html close;
ods graphics off;
run;
-------------------------------------------------------------------------
```

Table 11: **SAS Code for Logistic Modeling of AIDS Data in Table 5.6**

```
-------------------------------------------------------------------------
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;
run;
-------------------------------------------------------------------------
```

qualitative predictors to the AIDS and AZT study of Table 5.6. In PROC GENMOD, the OBSTATS option provides various "observation statistics," including predicted values and their confidence limits. The RESIDUALS option requests residuals such as the Pearson residuals and standardized residuals (labeled "Std Pearson Residual"). A CLASS statement requests indicator variables for the factor. By default, in PROC GENMOD the parameter estimate for the last level of each factor equals 0. In PROC LOGISTIC, estimates sum to zero. That is, dummies take the effect coding $(1, -1)$, with values of 1 when in the category and -1 when not, for which parameters sum to 0. In the CLASS statement in PROC LOGISTIC, the option PARAM = REF requests (1, 0) indicator variables with the last category as the reference level. 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, PROC 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 PROC LOGISTIC in Table 11) 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 12 analyzes the clinical trial data of Table 6.9 of the textbook. 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 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. The EQOR option in an EXACT statement

Table 12: **SAS Code for Cochran–Mantel–Haenszel Test and Related Analyses of Clinical Trial Data in Table 6.9**

```
-----------------------------------------------------------------------
data cmh;
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; exact eqor;
run;
-----------------------------------------------------------------------
```

provides an exact test for equal odds ratios proposed by Zelen (1971). O'Brien (1986) gave a SAS macro for computing powers using the noncentral chi-squared distribution.

Models with probit and complementary log-log (CLOGLOG) links are available with PROC GENMOD, PROC LOGISTIC, or PROC PROBIT. PROC SURVEYLOGISTIC fits binary regression models to survey data by incorporating the sample design into the analysis and using the method of pseudo ML (with a Taylor series approximation). It can use the logit, probit, and complementary log-log link functions.

For the logit link, PROC GENMOD can perform exact conditional logistic analyses, with the EXACT statement. It is also possible to implement the small-sample tests with mid-P-values and confidence intervals based on inverting tests using mid-P-values. The option CLTYPE = EXACT | MIDP requests either the exact or mid-P confidence intervals for the parameter estimates. By default, the exact intervals are produced.

Exact conditional logistic regression is also available in PROC LOGISTIC with the EXACT statement.

PROC GAM fits generalized additive models.

Table 13 shows the use of Bayesian methods for the analysis described in Section 7.2.2. (Note that the complete data set is in the Datasets link `www.stat.ufl.edu/~aa/cda/data.html` at this website.) Variances can be set for normal priors (e.g., VAR = 100), the length of the Markov chain can be set by NMC, and DIAGNOSTICS=MCERROR gives the amount of Monte Carlo error in the parameter estimates at the end of the MCMC fitting process.

The Firth penalized likelihood approach to reducing bias in estimation of logistic regression parameters is available with the FIRTH option in PROC LOGISTIC, as shown in Table 13.

Table 13: **SAS Code for Bayesian Modeling Example in Section 7.2.2 of Data on Endometrial Cancer in Table 7.2**

```
-------------------------------------------------------------------------
data endometrial;
input nv pi eh hg ;
nv2 = nv - 0.5; pi2 = (pi-17.3797)/9.9978; eh2 = (eh-1.6616)/0.6621;
datalines;
    0 13 1.64 0
    0 16 2.26 0
    0  8 3.14 0
    0 34 2.68 0
    0 20 1.28 0
    0  5 2.31 0
    0 17 1.80 0
    0 10 1.68 0
...
    1 11 1.01 1
    1 21 0.98 1
    0  5 0.35 1
    1 19 1.02 1
    0 33 0.85 1
    ;
proc genmod descending;
 model hg = nv2 pi2 eh2 /  dist=bin link=logit;
bayes coeffprior=normal (var=1.0) diagnostics=mcerror nmc=1000000;
run;
proc genmod descending;
 model hg = nv2 pi2 eh2 /  dist=bin link=logit;
bayes coeffprior=normal (var=100) diagnostics=mcerror nmc=1000000;
run;
proc logistic descending;
 model hg = nv2 pi2 eh2 / firth clparm=pl ;
run;
-------------------------------------------------------------------------
```

Table 14: **SAS Code for Baseline-Category Logit Models with Alligator Data in Table 8.1**

```
-------------------------------------------------------------------------
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;
run;
-------------------------------------------------------------------------
```

Chapter 8: Multinomial Response Models

PROC LOGISTIC fits baseline-category logit models 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 14 shows for the text example on alligator food choice (Table 8.1). 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 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 15 fits it to the data shown in Table 8.5 on happiness, number of traumatic events, and race. 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. Both procedures now have graphic capabilities of displaying the cumulative probabilities as a function of a predictor (with PLOTS = EFFECT), at fixed values of other predictors. Cox (1995) used PROC NLIN for the more general model having a scale parameter.

With the UNEQUALSLOPES option in PROC LOGISTIC, one can fit the model without proportional odds structure. For example, here it is for the 4×5 table on 4 doses with an ordinal outcome for data on p. 207 of the 2nd edition of my book "Analysis of Ordinal Categorical Data."

```
-------------------------------------------------------------------------
proc logistic data=trauma; *non-proportional odds cumulative logit model;
model outcome = dose / unequalslopes aggregate scale=none;

                                  Standard     Wald
Parameter  outcome  DF  Estimate   Error   Chi-Square  Pr > ChiSq
Intercept   1        1  -0.8646   0.1969     19.2793    <.0001
Intercept   2        1  -0.0937   0.1803      0.2705    0.6030
Intercept   3        1   0.7063   0.1766     15.9919    <.0001
Intercept   4        1   1.9087   0.2388     63.8771    <.0001
dose        1        1  -0.1129   0.0741      2.3235    0.1274
dose        2        1  -0.2689   0.0690     15.2000    <.0001
dose        3        1  -0.1823   0.0643      8.0366    0.0046
dose        4        1  -0.1193   0.0850      1.9704    0.1604
-------------------------------------------------------------------------
```

Both PROC GENMOD and PROC LOGISTIC can use other links in cumulative link models. PROC GENMOD uses LINK = CPROBIT for the cumulative probit model and LINK = CCLL for the cumulative complementary log-log model. PROC LOGISTIC fits a cumulative probit model using LINK = PROBIT.

PROC SURVEYLOGISTIC described above for incorporating the sample design into the analysis can also fit multicategory regression models to survey data, with links such as the baseline-category logit and cumulative logit.

You can fit adjacent-categories logit models in CATMOD by fitting equivalent baseline-category logit models. Table 16 uses it for Table 8.5 from the textbook, on happiness, number of traumatic events, and race. Each line of code in the model statement specifies the predictor values (for the two intercepts, trauma, and race) for the two logits. The trauma and race predictor values are multiplied by 2 for the first logit and 1 for the second logit, 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.

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 (8.18) in the text], the statistic for the "row mean scores differ" alternative treats X as nominal and Y as ordinal, and the statistic for

Table 15: **SAS Code for Cumulative Logit and Probit Models with Happiness Data in Table 8.5**

```
-------------------------------------------------------------------------
data gss;
    input race $ trauma happy;  * race is 0=white, 1=black;
datalines;
    0 0 1
    0 0 1
    0 0 1
    0 0 1
    0 0 1
    0 0 1
    0 0 1
    0 0 2
...
    1 3 3
        ;
ods graphics on;
ods html;
proc genmod; class race;
model happy = trauma race / dist=multinomial link=clogit lrci type3;
effectplot slicefit;
run;
proc logistic data=gss plots=effect(at(race=all)); class race;
model happy = trauma race ;
 output out=predict p=hi_hat lower=LCL upper=UCL;
 proc print data=predict;
run;
proc logistic data=gss plots=all; class race;
model happy = trauma race;
run;
ods html close;
ods graphics off;
run;
-------------------------------------------------------------------------
```

Table 16: **SAS Code for Adjacent-Categories Logit Model Fitted to Table 8.5 on Happiness, Traumatic Events, and Race**

```
-----------------------------------------------------------------
data gss;
input race trauma happy count;
count2 = count + 0.00000001;
datalines;
0 0 1  7
0 0 2 15
0 1 1  8
0 1 2 12
0 1 3  1
0 2 1  5
0 2 2 16
0 2 3  1
0 3 1  1
0 3 2  9
0 3 3  1
0 4 1  1
0 4 2  4
0 4 3  0
0 5 1  0
0 5 2  0
0 5 3  2
1 0 1  0
1 0 2  1
1 0 3  1
1 1 1  0
1 1 2  3
1 1 3  1
1 2 1  0
1 2 2  3
1 2 3  1
1 3 1  0
1 3 2  2
1 3 3  1
1 4 1  0
1 4 2  0
1 4 3  0
1 5 1  0
1 5 2  0
1 5 3  0
    ;
proc catmod order=data; weight count2;
population race trauma;
model happy = (1 0 0 0,  0 1 0 0,
               1 0 2 0,  0 1 1 0,
               1 0 4 0,  0 1 2 0,
               1 0 6 0,  0 1 3 0,
               1 0 8 0,  0 1 4 0,
               1 0 10 0, 0 1 5 0,
               1 0 0 2,  0 1 0 1,
               1 0 2 2,  0 1 1 1,
               1 0 4 2,  0 1 2 1,
```

Table 17: **SAS Code for Fitting Loglinear Models to High School Drug Survey Data in Table 9.3**

```
-------------------------------------------------------------------------
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;
run;
-------------------------------------------------------------------------
```

the "nonzero correlation" alternative treats X and Y as ordinal [statistic (8.19)]. See Stokes et al. (2012, Sec. 6.2) for a detailed discussion of various generalized CMH analyses.

PROC MDC fits multinomial discrete choice models, with logit and probit links (including multinomial probit models). See

`support.sas.com/documentation/cdl/en/etsug/60372/HTML/default/viewer.htm#etsug_mdc_sect021.htm`

For such models, 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).

Chapters 9–10: Loglinear Models

For details on the use of SAS (mainly with PROC GENMOD) for loglinear modeling of contingency tables and discrete response variables, see *Advanced Log-Linear Models Using SAS* by D. Zelterman (published by SAS, 2002).

Table 17 uses PROC GENMOD to fit loglinear model (AC, AM, CM) to Table 9.3 from the survey of high school students about using alcohol, cigarettes, and marijuana.

Table 18 uses PROC GENMOD for table raking of Table 9.15 from the textbook. Note the artificial pseudo counts used for the response, to ensure the smoothed margins.

Table 19 uses PROC GENMOD to fit the linear-by-linear association model (10.5) and the row effects model (10.7) to Table 10.3 in the textbook (with column scores 1, 2, 4, 5). The defined variable "assoc" represents the cross-product of row and column scores, which has β parameter as coefficient in model (10.5).

Correspondence analysis is available in SAS with PROC CORRESP.

Table 18: **SAS Code for Raking Table 9.15**

```
-------------------------------------------------------------------------
data rake;
input partyid polviews count;
log_ct = log(count); pseudo = 100/3;
datalines;
1 1 306
1 2 279
1 3 116
2 1 185
2 2 312
2 3 194
3 1  26
3 2 134
3 3 338
;
proc genmod; class partyid polviews;
  model pseudo = partyid polviews / dist=poi link=log offset=log_ct
     obstats;
run;
-------------------------------------------------------------------------
```

Table 19: **SAS Code for Fitting Linear-by-Linear Association and Row Effects Models to GSS Data in Table 10.3**

```
-----------------------------------------------------------------
data sex;
input premar birth u v count @@;  assoc = u*v ;
datalines;
1 1 1 1  81   1 2 1 2  68   1 3 1 4 60   1 4 1 5 38
...
;
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;
run;
-----------------------------------------------------------------
```

Table 20: **SAS Code for McNemar's Test and Comparing Proportions for Matched Samples in Table 11.1**

```
-----------------------------------------------------------------
data matched;
input first second count @@;
datalines;
 1 1 175   1 2 16   2 1  54   2 2 188
;
proc freq; weight count;
    tables first*second / agree;  exact mcnem;
proc catmod; weight count;
    response marginals;
    model first*second =  (1 0 ,
                           1 1 ) ;
run;
-----------------------------------------------------------------
```

Chapter 11: Models for Matched Pairs

Table 20 analyzes Table 11.1 on presidential voting in two elections. 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*), 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 Wald 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 21 shows ways of testing marginal homogeneity for the migration data in Table 11.5 of the textbook. The PROC 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 (11.15) of marginal homogeneity, as shown.

Table 22 shows various square-table analyses of Table 11.6 of the textbook on premarital and extramarital sex. 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

Table 21: **SAS Code for Testing Marginal Homogeneity with Migration Data in Table 11.5**

```
-------------------------------------------------------------------------------
data migrate;
input then $ now $ count m11 m12 m13 m21 m22 m23 m31 m32 m33 m44 m1 m2 m3;
datalines;
  ne ne 266  1  0  0  0  0  0  0  0  0  0  0  0 1  1
  ne mw  15  0  1  0  0  0  0  0  0  0  0  0  0 2  5
  ne  s  61  0  0  1  0  0  0  0  0  0  0  0  0 3  5
  ne  w  28 -1 -1 -1  0  0  0  0  0  0  1  0  0 4  5
  mw ne  10  0  0  0  1  0  0  0  0  0  0  0  0 2  5
  mw mw 414  0  0  0  0  1  0  0  0  0  0  0  0 5  2
  mw  s  50  0  0  0  0  0  1  0  0  0  0  0  0 6  5
  mw  w  40  0  0  0 -1 -1 -1  0  0  0  0  1  0 7  5
   s ne   8  0  0  0  0  0  0  1  0  0  0  0  0 3  5
   s mw  22  0  0  0  0  0  0  0  1  0  0  0  0 6  5
   s  s 578  0  0  0  0  0  0  0  0  1  0  0  0 8  3
   s  w  22  0  0  0  0  0  0 -1 -1 -1  0  0  0 1 9  5
   w ne   7 -1  0  0 -1  0  0 -1  0  0  0  1  0  0 4  5
   w mw   6  0 -1  0  0 -1  0  0 -1  0  0  0  1  0 7  5
   w  s  27  0  0 -1  0  0 -1  0  0 -1  0  0  0  1 9  5
   w  w 301  0  0  0  0  0  0  0  0  0  1  0  0  0 10 4
;
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;
run;
-------------------------------------------------------------------------------
```

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 a quasi-uniform association model that takes the uniform association version of the linear-by-linear association model and imposes a perfect fit on the main diagonal.

The bottom of Table 22 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 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 mentioned in Note 11.2.

Table 23 uses PROC GENMOD for logit fitting of the Bradley–Terry model (11.30) to the baseball data of Table 11.10, 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.

Table 24 uses PROC GENMOD for fitting the complete symmetry and quasi-symmetry models to Table 11.13 on attitudes toward legalized abortion.

Table 25 shows the likelihood-ratio test of marginal homogeneity for the attitudes toward abortion data of Table 11.13, where for instance *m11p* denotes μ_{11+}. The marginal homogeneity model expresses the eight cell expected frequencies in terms of $\mu_{111}, \mu_{11+}, \mu_{1+1}, \mu_{+11}, \mu_{1++}$, 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 (11.37) of marginal homogeneity.

Chapter 12: Clustered Categorical Responses: Marginal Models

Table 26 uses PROC GENMOD to analyze Table 12.1 from the textbook on depression, 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-type tests about effects. See Stokes et al. (2012) for the use of GEE with missing data. PROC GENMOD also provides deletion and diagnostics statistics for its GEE analyses and provides graphics for these statistics.

Table 22: **SAS Code Showing Square-Table Analysis of Table 11.6 on Premarital and Extramarital Sex**

```
-------------------------------------------------------------------------
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 qi unif / dist=poi link=log;
                                          * quasi unif. assoc. ;
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 = / dist=bin link=logit noint; * symmetry;
proc genmod data=sex2;
  model above/trials = score / dist=bin link=logit noint; * ordinal QS;
proc genmod data=sex2;
  model above/trials = / dist=bin link=logit; * conditional symm;
run;
-------------------------------------------------------------------------
```

Table 23: **SAS Code for Fitting Bradley–Terry Model to Baseball Data of Table 11.10**

```
-------------------------------------------------------------------------
data baseball;
input wins games boston newyork tampabay toronto baltimore ;
datalines;
 12 18  1 -1  0  0  0
 6  18  1  0 -1  0  0
 10 18  1  0  0 -1  0
 10 18  1  0  0  0 -1
 9  18  0  1 -1  0  0
 11 18  0  1  0 -1  0
 13 18  0  1  0  0 -1
 12 18  0  0  1 -1  0
 9  18  0  0  1  0 -1
 12 18  0  0  0  1 -1
    ;
proc genmod;
   model wins/games = boston newyork tampabay toronto baltimore /
   dist=bin link=logit noint covb obstats;
run;
-------------------------------------------------------------------------
```

Table 24: **SAS Code for Fitting Complete Symmetry and Quasi-Symmetry Models to Attitudes toward Legalized Abortion Data of Table 11.13**

```
-------------------------------------------------------------------------
data gss;
    input absingle abhlth abpoor count;
    sum = absingle + abhlth + abpoor;
 datalines;
 1 1  1 466
 1 1  0  39
 1 0  1   3
 1 0  0   1
 0 1  1  71
 0 1  0 423
 0 0  1   3
 0 0  0 147
     ;
proc genmod data=gss; class sum;
    model count = sum / dist=poisson link=log; * symmetry;
run;
proc genmod data=gss; class sum;
    model count = sum absingle abhlth abpoor / dist=poisson link=log
        obstats; * quasi-symmetry;
run;
-------------------------------------------------------------------------
```

Table 25: **SAS Code for Testing Marginal Homogeneity with Attitudes toward Legalized Abortion Data of Table 11.13**

```
-------------------------------------------------------------------------
data abortion;
input a b c count m111 m11p m1p1 mp11 m1pp m222 @@;
datalines;
 1 1 1 466   1  0  0  0  0  0      1 1 2   3  -1  1  0  0  0  0
 1 2 1  71  -1  0  1  0  0  0      1 2 2   3   1 -1 -1  0  1  0
 2 1 1  39  -1  0  0  1  0  0      2 1 2   1   1 -1  0 -1  1  0
 2 2 1 423   1  0 -1 -1  1  0      2 2 2 147   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 item 3;
run;
-------------------------------------------------------------------------
```

Table 26: **SAS Code for Marginal (GEE) and Random Effects Modeling of Depression Data in Table 12.1**

```
-----------------------------------------------------------------------------
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 nlmixed  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;
run;
-----------------------------------------------------------------------------
```

Table 27 uses PROC GENMOD to implement GEE for a cumulative logit model for the insomnia data of Table 12.3. For multinomial responses, independence is currently the only working correlation structure.

Chapter 13: Clustered Categorical Responses: Random Effects Models

PROC NLMIXED extends GLMs to GLMMs by including random effects. Table 28 analyzes the matched pairs model (13.3) for the change in presidential voting data in Table 13.1.

Table 29 analyzes the Presidential voting data in Table 13.2 of the text, using a one-way random effects model.

Table 30 fits model (13.11) to the attitudes on legalized abortion data of Table 13.3. 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 26 above uses PROC NLMIXED for fitting the random effects model to Table 12.1 on depression, with initial estimates specified. Table 27 uses PROC NLMIXED for ordinal random effects modeling of Table 12.3 on insomnia, defining a general multinomial log likelihood.

Agresti et al. (2000) showed PROC NLMIXED examples for clustered data. Table 31 shows code for multicenter trials such as Table 13.7. Table 32 shows code for multicenter trials with an ordinal response, for data analyzed in Hartzel et al. (2001a) on comparing two drugs for a three-category response at eight centers.

Table 33 shows a correlated bivariate random effect analysis of Table 13.8 on attitudes toward the leading crowd.

Table 34 shows PROC NLMIXED code for an adjacent-categories logit model and a nominal model for the movie rater data analyzed in Hartzel et al. (2001b).

Chen and Kuo (2001) discussed fitting multinomial logit models, including discrete-choice models, with random effects.

PROC NLMIXED allows only one RANDOM statement, which makes it difficult to incorporate random effects at different levels. PROC GLIMMIX has more flexibility. It also fits random effects models and provides built-in distributions and associated variance functions as well as link functions for categorical responses. It can provide a variety of fitting methods, including pseudo likelihood methods, but not ML. See

`support.sas.com/rnd/app/da/stat/procedures/glimmix.html`

Chapter 14: 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 (14.10), and SCALE = PEARSON uses the scaled binomial variance (14.11). Table 35 illustrates for Table 4.7 from a teratology study. That table also uses PROC NLMIXED for adding litter random intercepts.

Table 27: **SAS Code for Marginal (GEE) and Random Intercept Cumulative Logit Analysis of Insomnia Data in Table 12.3**

```
------------------------------------------------------------------------
data francom;
 input case treat time outcome @@;
 y1=0;y2=0;y3=0;y4=0;
 if outcome=1 then y1=1;
 if outcome=2 then y2=1;
 if outcome=3 then y3=1;
 if outcome=4 then y4=1;
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 nlmixed  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(eta1)/(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));
  ll = y1*log(p1) + y2*log(p2) + y3*log(p3) + y4*log(p4);
  model y1 ~ general(ll);
  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;
run;
------------------------------------------------------------------------
```

Table 28: **SAS Code for Fitting Model (13.3) for Matched Pairs to Table 13.1**

```
-------------------------------------------------------------------------
data kerryobama;
 input case occasion response count;
datalines;
 1  0 1 175
 1  1 1 175
 2  0 1 16
 2  1 0 16
 3  0 0 54
 3  1 1 54
 4  0 0 188
 4  1 0 188
;
proc nlmixed data=kerryobama qpoints=1000;
  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;
run;
-------------------------------------------------------------------------
```

Table 29: **SAS Code for GLMM Analysis of Election Data in Table 13.2**

```
-------------------------------------------------------------------
data vote;
input y n;
case = _n_;
datalines;
1   5
16  32
...
1   4
;
proc nlmixed;
   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;
run;
-------------------------------------------------------------------
```

For Table 14.6 on homicides, Table 36 uses PROC GENMOD to fit a negative binomial model and a quasi-likelihood model with scaled Poisson variance using the Pearson statistic, and PROC NLMIXED to fit a Poisson GLMM. PROC NLMIXED can also fit negative binomial models.

The PROC GENMOD procedure fits zero-inflated Poisson regression models.

Chapter 15: Non-Model-Based Classification and Clustering

PROC DISCRIM in SAS can perform discriminant analysis. For example, for the ungrouped horseshoe crab as analyzed above in Table 9, you can add code such as shown in Table 37. The statement "priors prop" sets the prior probabilities equal to the sample size. Alternatively "priors equal" would have equal prior proportions in the two categories.

PROC DISCRIM can also be used to perform smoothing using kernel methods and using k-nearest neighbor methods. See

`support.sas.com/documentation/cdl/en/statug/63347/HTML/default/viewer.htm#statug_discrim_sect017.htm`

Classification trees can be construted using the special Enterprise Miner add-on.

PROC DISTANCE can form distances such as the Jaccard index between pairs of variables. Then, PROC CLUSTER can perform a cluster analysis. Table 38 illustrates for Table 15.6 of the textbook on statewide grounds for divorce, using the average

Table 30: **SAS Code for GLMM Modeling of Opinions in Table 13.3**

```
-------------------------------------------------------------------------
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 nlmixed qpoints = 1000;
   parms alpha=-0.7   beta1=.8  beta2=.3  gamma=0 sigma=8.7;
   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 = subject;
   replicate count;
run;
-------------------------------------------------------------------------
```

Table 31: **SAS Code for GLMM for Multicenter Trial Data in Table 13.7**

```
-------------------------------------------------------------------------------
data binomial;
input center treat y n @@ ; * y successes out of n trials;
if treat = 1 then treat = .5; else treat = -.5;
cards;
1 1 11 36    1 0 10 37     2 1 16 20   2 0 22 32
3 1 14 19    3 0  7 19     4 1  2 16   4 0  1 17
5 1  6 17    5 0  0 12     6 1  1 11   6 0  0 10
7 1  1  5    7 0  1  9     8 1  4  6   8 0  6  7
;
run;

proc genmod data=binomial;  * fixed effects, no interaction model;
  class center;
  model y/n = treat center / dist=bin link=logit noint;
run;

proc nlmixed data=binomial qpoints=15;  * random effects, no interaction;
   parms alpha=-1 beta=1 sig=1;  * initial values for parameter estimates;
   pi = exp(a + beta*treat)/(1+exp(a + beta*treat)); * logistic formula for prob;
   model y ~ binomial(n, pi);
   random a ~ normal(alpha, sig*sig) subject=center;
   predict a + beta*treat   out=OUT1;
run;

proc nlmixed data=binomial qpoints=15;  * random effects, interaction;
   parms alpha=-1 beta=1 sig_a=1 sig_b=1;  * initial values;
   pi = exp(a + b*treat)/(1+exp(a + b*treat));
   model y ~ binomial(n, pi);
   random a b ~ normal([alpha,beta], [sig_a*sig_a,0,sig_b*sig_b]) subject=center;
run;
-------------------------------------------------------------------------------
```

Table 32: **SAS Code for GLMM for cumulative logit random effects heterogeneity model fitted to data in Hartzel et al. (2001a)**

```
-------------------------------------------------------------------------------
data ordinal;
   do center = 1 to 8;
      do trt = 1 to 0 by -1;
         do resp = 3 to 1 by -1;
            input count @@;
            output;
         end;
      end;
   end;
   datalines;
   13   7   6   1   1  10   2   5  10   2   2   1
   11  23   7   2   8   2   7  11   8   0   3   2
   15   3   5   1   1   5  13   5   5   4   0   1
    7   4  13   1   1  11  15   9   2   3   2   2
run;

proc nlmixed data=ordinal qpoints=15;
  ** To maintain the threshold ordering define thresholds such that  **;
  ** threshold 1 = 0 and threshold 2 = i2, where i2 > 0.             **;
  ** Use starting value of 0 for sig_cb.                            **;
  bounds i2>0;   parms sig_cb=0;
  eta1 =      c - b*trt;
  eta2 = i2 - c - b*trt;
  if (resp=1) then z = 1/(1+exp(-eta1));
     else if (resp=2) then z = 1/(1+exp(-eta2)) - 1/(1+exp(-eta1));
     else z = 1 - 1/(1+exp(-eta2));
  if (z > 1e-8) then ll = count*log(z);  ** Check for small values of z **;
     else ll=-1e100;
  model resp ~ general(ll);            ** Define general log-likelihood. **;
  random c b ~ normal([gamma,beta],[sig_c*sig_c, sig_cb, sig_b*sig_b])
         subject = center out = out1; ** OUT1 contains predicted center-     **;
                                      ** specific cumulative log odds ratios **;
run;
-------------------------------------------------------------------------------
```

Table 33: **SAS Code for GLMM for Leading Crowd Data in Table 13.8**

```
-------------------------------------------------------------------------------
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;
run;
-------------------------------------------------------------------------------
```

Table 34: **SAS Code for GLMM for Movie Rater Data in Hartzel et al. (2001b)**

```
--------------------------------------------------------------------------------
/*Input Data*/
data movie;
do lyons = 1 to 3;
 do siskel = 1 to 3;
  do medved = 1 to 3;
   do ebert = 1 to 3;
      input count @@;
      if (count ne 0) then output;
   end;
  end;
 end;
end;
datalines;
15 2 0   2 0 0   8 0 2
 0 3 0   2 3 0   3 2 1
 1 0 1   1 2 1   1 1 2

 2 0 0   2 1 0   0 1 0
 1 1 0   0 1 0   0 1 0
 1 0 0   0 0 0   0 1 1

 3 1 1   0 0 0   4 1 0
 0 0 1   0 0 0   1 3 0
 0 0 1   2 1 0   2 2 4
;


/*Create subject variable "movie" and code dummy variables*/
data movie;
set movie;
movie = _n_;
x1 = 1; x2 = 0; x3 = 0; resp = siskel; output;
x1 = 0; x2 = 1; x3 = 0; resp = ebert;  output;
x1 = 0; x2 = 0; x3 = 1; resp = lyons;  output;
x1 = 0; x2 = 0; x3 = 0; resp = medved; output;
drop lyons siskel medved ebert;

/*Recode responses as multivariate Bernoulli*/
data movie;
set movie;
  if resp = 1 then do;
     y1=1; y2=0; y3=0;
  end;
  else if resp = 2 then do;
     y1=0; y2=1; y3=0;
  end;
  else if resp = 3 then do;
     y1=0; y2=0; y3=1;
  end;
  else delete;
  output;
```

Table 35: **SAS Code for Overdispersion Analysis of Teratology Study Data of Table 4.7**

```
-------------------------------------------------------------------------------
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 nlmixed  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;
run;
-------------------------------------------------------------------------------
```

Table 36: **SAS Code for Fitting Models to Murder Data in Table 14.6**

```
------------------------------------------------------------------------
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 nlmixed  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;
run;
------------------------------------------------------------------------
```

Table 37: **SAS Code for Discriminant Analysis for Horseshoe Crab Data Analyzed in Section 15.1.2**

```
--------------------------------------
proc discrim data=crab crossvalidate;
    priors prop;
    class y;
    var width color;
run;
--------------------------------------
```

linkage method for pairs of clusters with the Jaccard dissimilarity index.

Chapter 16: Large- and Small-Sample Theory for Multinomial Models

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. Or, you can pick the type of confidence interval you want by specifying CLTYPE=EXACT or CLTYPE=MIDP. In particular, this enables you to get the Cornfield exact interval for an odds ratio (also available with PROC FREQ as shown above in Table 4), or its mid-P adaptation. You can also conduct the exact conditional version of the Cochran–Armitage test using the TREND option in the EXACT statement with PROC FREQ. One can also conduct an asymptotic conditional logistic regression, using a STRATA statement to indicate the stratification parameters to be conditioned out. PROC PHREG can also do this (Stokes et al. 2012). For a $2\times2\times K$ table, using the EQOR option in an EXACT statement in PROC FREQ provides an exact test for equal odds ratios proposed by Zelen (1971), as shown above in Table 12.

Table 38: **SAS Code for Discriminant Analysis for Table 15.6 on Statewide Grounds for Divorce**

```
-------------------------------------------------------------------------
data divorce;
  input state $ incompat cruelty desertn non_supp alcohol
                felony impotenc insanity separate ;
   datalines;
   California     1 0 0 0 0 0 0 1 0
   Florida        1 0 0 0 0 0 0 1 0
   Illinois       0 1 1 0 1 1 1 0 0
   Massachusetts  1 1 1 1 1 1 1 0 1
   Michigan       1 0 0 0 0 0 0 0 0
   NewYork        0 1 1 0 0 1 0 0 1
   Texas          1 1 1 0 0 1 0 1 1
   Washington     1 0 0 0 0 0 0 0 1
 ;
 title Grounds for Divorce;
 proc distance  data=divorce  method=djaccard  absent=0  out=distjacc;
   var anominal(incompat--separate);
   id state;
run;
proc print data=distjacc(obs=8);
  id state;
  title2 Only 8 states;
run;
proc cluster  data=distjacc  method=average
  pseudo outtree=tree;
  id state;
run;
proc tree data=tree horizontal n=7 out=out ;
id state;
run;
-------------------------------------------------------------------------
```