Visualizing Categorical Data with SAS and R: Exercises

Michael Friendly Ernest Kwan Cathy LaBrish

March 1, 2016

Getting started

The goal of these workshops is to give you some hands-on experience using the methods described in the lectures. For each lecture, the plan is to have you try to reproduce, and possibly extend, the analysis and visualization of one or two examples used in that lecture, with SAS or R (or both), as you are most comfortable with. In addition, there may also be one or two new examples to try. Some materials for these workshops are available online at `http://datavis.ca/courses/VCD/`.

For reference, Table 1 lists the principal data sets that are used in the lectures. The `SAS` column gives the name of the `.sas` file containing the data and the name of the SAS data set created from this. The `R` column gives the name of the data set in R. An asterisk (*) marks the first use of each data set.

Table 1: Data sets used in the lectures

Part	Data set	`SAS`	`R`
1	Federalist *	`madison`	`Federalist`
	Cholesterol-HD *	`fat`	
	Arthritis *	`arthrit`	`Arthritis`
2	Berkeley *	`berkeley`	`UCBAdmissions`
	Hair-Eye-Color *	`haireye`	`HairEyeColor`
	Distant-Vision *	`vision`	`VisualAcuity`
	Sex-Fun *	`sexfun`	`SexualFun`
	MS diagnosis *	`msdiag`	`MSPatients`
	Suicide-Rates *	`suicide`	`Suicide`
3	Berkeley	`berkeley`	`UCBAdmissions`
	Hair-Eye-Color *	`haireye`	`HairEyeColor`
	Mental-Health-SES *	`mental`	`Mental [vcdExtra]`
	Distant-Vision *	`vision`	`VisualAcuity`
4	Berkeley	`berkeley`	`UCBAdmissions`
	Arthritis	`arthrit`	`Arthritis`
	Volunteering *	`cowles`	`Cowles [car]`
	Arrests *	`arrests`	`Arrests [effects]`
5	Arthritis	`arthrit`	`Arthritis`
	Womens-Labor-Force *	`wlfpart`	`Womenlf [car]`

SAS and R setups

Here are a few things you should know to get started using SAS and R.

SAS All of the SAS programs, data sets and macros are stored in `N:/Vcd`. Documentation for all of my macro programs is available at `http://datavis.ca/sasmac/`, mostly under the **Categorical data** menu item. SAS

To use a data set in one of the exercises, start with `%include catdata(dataset)` for example,

```
%include catdata(madison);      *-- Federalist papers, 'may' by Madison;
%include catdata(arthrit);      *-- Arthritis treatment data;
```

The SAS macro programs are installed in a `macros` directory that is automatically searched when you invoke it with `%macroname(...)`. For example:

```
%include catdata(madison);      *-- Federalist papers, 'may' by Madison;
%goodfit(data=madison, var=count, freq=blocks, dist=poisson);
```

Within SAS, you can access the web help page for any macro with `%webhelp(macroname);`. For example,

```
%webhelp(goodfit);
```

For help on SAS procedures, you can use the Help menu in SAS, or access the complete SAS Online documentation (v. 9.1.3) via

```
%webhelp(online);
```

R All of the R programs and data sets should be pre-installed in R on the server, but some of them may be outdated. The *first* time you start R, you should update the available packages and load the `vcdExtra` package to make available some extra workshop materials. I recommend using R Studio, `https://www.rstudio.com/`, rather than the standard R GUI. R

```
update.packages(ask=FALSE)
library(vcdExtra)
```

Thereafter, when you start R, you should begin with

```
library(vcdExtra)
```

to load the `vcdExtra` package. This includes `vcd` plus some extra data sets.

To load a data set from an R package, use the `data()` function. To get a description of any R data set or function, use `help()`, or the short-hand `?`.

```
data(Federalist)        # load the data
?Federalist             # read the help()
?mosaic                 # help for mosaic()
```

Home-work

In the limited time available in the lab sessions, it is unlikely that you will be able to complete all the exercises or to master all of these techniques. To profit more fully, and apply them to your own data, you should try to practice more at home or at work.

To this end, most of the materials for this course and the workshops are available at `http://datavis.ca/courses/VCD/`. From there you can download the SAS data sets and macros (in the archive `vcdprog.zip`) and the necessary R packages. vcdprog.zip

1 Introduction & overview

1.1 Discrete distributions

In this exercise, we will examine the use of diagnostic and graphic methods for fitting discrete distributions to the one-way frequency tables shown in Table 2 and Table 3.

Table 2: Number of occurrences (k) and number of blocks of text (n_k) of the word *may* in essays written by Madison. SAS: `madison`; R: `Federalist`

Occurrences (k)	0	1	2	3	4	5	6
Blocks (n_k)	156	63	29	8	4	1	1

Table 3: Number of males in N=6115 Saxony families with 12 children. SAS: `saxony`; R: `Saxony`

k	0	1	2	3	4	5	6	7	8	9	10	11	12
n_k	3	24	104	286	670	1033	1343	1112	829	478	181	45	7

Tasks & questions

For the Madison data, it might be supposed that number of occurrences of the word *may* follow a Poisson distribution.

1. Fit a Poisson distribution to the data. Do you conclude that this distribution fits well or badly?
2. Examine the departure between the observed and fitted frequencies with a hanging rootogram. Is there a pattern to the departure?
3. Use an Ord plot to diagnose the form of the distribution.[1]
4. If necessary, fit a different distribution and examine visually with a hanging rootogram.

SAS

With SAS, you can begin these steps with statements like the following:

```
%include catdata(madison);
%goodfit(data=madison, var=count, freq=blocks, dist=poisson, out=fit);
%rootgram(data=fit, var=count, obs=blocks);

%ordplot(data=madison, count=count, freq=blocks);
```

R

With R the following statements will get you started:

```
data(Federalist)
gf <- goodfit(Federalist, type='poisson')
summary(gf)
plot(gf)

Ord_plot(Federalist, main='may in Federalist papers')
```

For the Saxony data, it might be supposed that number of male children follow a Binomial distribution, perhaps also with $p(Male) = 0.5$. Repeat the steps above using the Saxony data, with a Binomial distribution.

[1] To judge whether a coefficient is positive or negative a tolerance is used. Be careful with the conclusions from the Ord plot, as it implements just some simple heuristics!

1.2 Testing association

The Berkeley data set (Bickel et al., 1975) involves graduate school applicants to the six largest departments at University of California, Berkeley in 1973. The applicants are classified by `Admit` (admitted vs. rejected), `Gender` (male vs. female), and `Dept` (A to F).
This data set forms a 3-way table (i.e., a $2 \times 2 \times 6$ table), but in this exercise, we will first look at the 2×2 table of `Admit` by `Gender`, collapsed over `Dept`. In other words, we will ignore (temporarily) the department to which each applicant applied. In later lectures and other exercises we will examine the full 3-way table.
This analysis is of interest because an association between `Admit` and `Gender`— different rates of admissions for male and female applicants could be considered as evidence of gender bias.

Tasks & questions

1. Construct the two-way frequency table by collapsing over `Dept`, and test whether `Admit` and `Gender` are independent.
2. Find the proportion admitted for males and females and relate these to the marginal frequencies and expected frequencies under independence. An independent relationship between Admit and Gender in the sample implies that cell frequencies would be proportionate to the product of marginal frequencies. For example, 59.5% (2691/4526) of applicants are male, 61.2% (2771/4526) of applicants are rejected.
3. Do these frequencies suggest approximately equal admission rates in males and females? If not, try to describe the possible gender difference in admission. What is a good statistic to summarize the relation between gender and admission?

SAS

In SAS, with a data set in frequency form, you can collapse over any variables simply by omitting them from the `tables` statement in `PROC FREQ`. See `berkeley-freq.sas`

berkeley-freq.sas

```
%include catdata(berkeley);
proc freq data=berkeley;
   weight freq;
   tables gender*admit / chisq plots=all;
         format admit admit.;
   run;
```

R

For R, the data set `UCBAdmissions` is represented as a 3-way table in table form. Use `margin.table()` to collapse any such table over omitted dimensions. `CrossTable()` in the `gmodels` package has many options for displaying quantities of interest in contingency tables.

```
data(UCBAdmissions)
UCB.GA <-margin.table(UCBAdmissions, c(1,2))
chisq.test(UCB.GA)

library(gmodels)
CrossTable(UCB.GA, prop.t=FALSE, prop.r=FALSE, expected=TRUE)
```

1.3 Stratified analysis

The marginal analysis of the Berkeley data in Section 1.2 is misleading, because (as it turns out) it falsely assumes that relation between `Admit` and `Gender` is the *same* for each department. Only if this were true, would the collapsed table represent the relation in *all* departments.

Here, you will use a stratified analysis to examine the relation between `Admit` and `Gender` *controlling for* department (equivalently: a test of *conditional* independence `Admit` and `Gender` *given* `Gender`), as opposed to *ignoring* department in the marginal analysis. In Lectures II and III we will see some visualizations of these data that help to explain the contradictory results.

Tasks & questions

1. Carry out stratified tests of the association between `Admit` and `Gender` controlling for `Dept`.
2. What do you conclude about whether there is a difference in rate of admission in *each* of the departments?
3. Test for conditional independence of `Admit` and `Gender` given `Dept`.
4. For one 2×2 table, the odds ratio is a reasonable summary measure of association. For a collection of k such tables, the Breslow-Day χ^2 tests whether these are equal across strata. What do you conclude?

Using SAS, `PROC FREQ` will carry out a stratified analysis when there are *more than two* factors given in the `tables` statement You get a separate page of output for each stratum, followed by overall summary tests of the relation between the two primary table variables, *controlling for* the strata. These are the tests and statistics of most interest here. SAS

```
%include catdata(berkeley);
proc freq data=berkeley;
  weight freq;
  tables dept * gender * admit /chisq cmh;
  format dept dept. admit admit.;
  run;
```

In R, for $2 \times 2 \times K$ tables, the `mantelhaen.test()` function provides a test of overall association pooled over strata, `woolf_test()` tests for equal odds ratios across strata (similar to the Breslow-Day test), and `CMHtest()` in `vcdExtra` package gives generalized CMH tests like those provided by the `cmh` option in SAS. R

Tests of similar hypotheses can be carried out using loglinear models (`loglm()`) and generalized linear models (`glm()`), described in later lectures. Here is one simple way to test for association of `Admit` and `Gender` *separately* for each department. `oddsratio()` calculates and tests the (log) odds ratio for each stratum.

```
for (dept in 1:6) {print(chisq.test(UCBAdmissions[,,dept]))}
summary(oddsratio(UCBAdmissions))
```

Some tests related to this exercise can be done as follows. You should try to understand what hypothesis is tested by each one.

```
mantelhaen.test(UCBAdmissions)
woolf_test(UCBAdmissions)
CMHtest(UCBAdmissions)
```

1.4 Ordinal factors

The Sex-Fun data set (Hout et al., 1987) involves the responses of 91 couples to the questionnaire item: *Sex is fun for me and my partner*: (a) never or occasionally, (b) fairly often, (c) very often, (d) almost always.

The data can be regarded as a 4×4 frequency table, where both factors can be considered to be ordered. For today, we will just consider testing whether there is an association between the husband's and wife's rating, and how to take ordinality into account, as well as the relatively small sample size.
In this problem, we could arguably be more interested in assessing to what extent the ratings actually *agree*– there could actually be a strong *negative* association between the Husband's and Wife's rating of sexual fun! Methods for assessing agreement will be discussed in the lecture for Part II.

Tasks & questions

1. Carry out a simple Pearson χ^2 test for association of `Husband` and `Wife`. Are there any reasons to question the validity of this test?
2. Carry out the analogous Cochran-Mantel-Haenszel tests. SAS reports three different tests. Which one is most appropriate here, where both variables are ordinal?
3. For relatively small total sample size, where the asymptotic distribution of the Pearson χ^2 statistic is in doubt, Fisher's exact test provides an alternative. Do it.
4. Examine the association statistics between the ratings. What would be a reasonable value to take as a non-parametric measure of "correlation" between Husband and Wife ratings?[2]

For SAS, you can carry out all these steps as follows shown below. See the file `sexfun-cmh.sas`.

SAS

sexfun-cmh.sas

```
%include catdata(sexfun);
proc freq data=sexfun order=data;
  weight count;
  tables husband * wife / cmh chisq exact nocol norow;
  run;
```

For comparison with the association statistics, note that you can also calculate the Pearson correlation between the ratings using `PROC CORR`:

```
proc corr data=sexfun ;
  weight count;
  var husband wife;
  run;
```

In R, the equivalent of the `cmh` option in `PROC FREQ` is `vcdExtra::CMHtest()`. You can do the various steps as follows. See `sexfun-chisq.R`.

sexfun-chisq.R

R

```
library(vcdExtra)
data(SexualFun)
SexualFun                # show the table
CMHtest(SexualFun)       # CMH tests
chisq.test(SexualFun)    # general association
fisher.test(SexualFun)   # exact test
assocstats(SexualFun)    # association statistics
```

[2] Hint: the Cochran-Mantel-Haenszel $\chi^2_{CMH} = (n-1)r^2$ when both variables are assigned integer scores. The Phi coefficient, $\phi = \sqrt{\chi^2_P / n}$, but for larger than 2×2 tables it has a range $0 \leq \phi \leq \min(\sqrt{r-1}, \sqrt{c-1})$. Cramer's V is a corrected scaling of ϕ so that $0 \leq V \leq 1$, similar to r^2.

2 Two-way and n-way tables

In these exercises you will examine several of the methods for visualizing association in two-way and n-way tables. The goal is not simply to see *if* an association exists, but to understand the nature or pattern of association— *why* or *how* the variables are associated.

2.1 2×2 tables

In 2×2 tables, there are several measures of association between the row and column variables, but one of the simplest is the *odds ratio*, θ, $0 \leq \theta \leq \infty$, estimated in the sample as

$$\hat{\theta} = \frac{n_{11}/n_{12}}{n_{21}/n_{22}} = \frac{n_{11}n_{22}}{n_{12}n_{21}}$$

When there is no association, $\theta = 1$ (or, equivalently, $\log(\theta) = 0$). For a stratified $2 \times 2 \times k$ table, another question of interest is whether the odds ratios are the same for all strata, $\theta_1 = \theta_2, \ldots, \theta_k$. In this exercise, you will use fourfold displays to examine these questions.

2.1.1 Cholesterol and heart disease

Tasks & questions

1. Test whether there is an association between cholesterol and heart disease using a fourfold display.
2. What evidence is shown in the fourfold display regarding whether the odds ratio differs significantly from 1?
3. Describe *how* cholesterol and heart disease are related.

In SAS, the data are in `fat.sas` in the `catdata` directory. This file also runs a `proc freq` to give answers to the first question. SAS

```
%include catdata(fat);
%ffold(data=fat, var=diet disease);
```

In R, you can just create the `fat` data using `matrix()`. `chisq.test(fat)` will give an approximate answer to the first question. R

```
fat <- matrix( c(6, 4, 2, 11), 2, 2)
dimnames(fat) <- list(diet=c("LoChol", "HiChol"), disease=c("No", "Yes"))
fourfold(t(fat))
```

2.1.2 Berkeley data

Here, you will re-examine the Berkeley admissions data considered in Section 1.2 and Section 1.3 using fourfold plots.

Tasks & questions

1. Assess visually whether there is an association between gender and admission, in the marginal table that collapses over department.
2. Assess visually the association between gender and admission separately in each department. What do you conclude?
3. Examine the question of whether the odds ratio of admission for males and females is the *same* for all departments. [Hint: See the printed output from the `ffold` macro, or use the `woolf_test()` in R.]

In SAS, the `table` macro can be used to collapse a frequency table, or to change numeric variables into character labels, suitable for the `ffold` macro. The required SAS statements for this exercise are contained in `berk-4fold.sas`. The first step looks like this:

SAS

berk-4fold.sas

```
%include catdata(berkeley);
%table(data=berkeley, out=berk2,
   var=Admit Gender,
   weight=freq,
   order=data);
%ffold(data=berk2, var=Admit Gender);
```

In R, `margin.table()` collapses a table over dimensions not listed in the second argument. The R statements below are contained in `berk-4fold.R`. See the `vignette("vcd-tutorial")` tutorial, Section 3.4, for other tests and plots of these data.

berk-4fold.R

R

```
UCB <- aperm(UCBAdmissions, c(2,1,3))
fourfold(margin.table(UCB, c(1, 2)))     # two-way plot

fourfold(UCB, mfrow=c(2,3))              # three-way plot
woolf_test(UCB)
```

2.2 Sieve diagrams

The Distant-vision data records the classification of visual acuity in the left and right eyes for a large sample of employees of the Royal Ordnance factor in the UK during world War II. There are two samples— one of men and one of women. Here we will just look at the female sample.

We expect to find a strong degree of association (and agreement) between the grade recorded for the two eyes. The more interesting question is the *nature* of the association. A sieve diagram is useful for this purpose.

Tasks & questions

1. Construct a sieve diagram for these data.
2. Make sure you understand the principal by which this diagram is constructed, e.g., why are the boxes for the cells of the table aligned in rows and columns.
3. Describe the patterns you see in the density of shading.

In SAS, the data are found in the file `vision.sas`, and contain the data set `women`, for the female sample. See the file `vision-sieve.sas` for this and more.

SAS

vision-sieve.sas

```
%include catdata(vision);
proc print data=women;
   run;
title 'Vision data: Women';
%sieveplot(data=women, var=right left, filltype=obsp);
```

For R, you will find the `VisualAcuity` data in the `vcd` package. In R, `subset()` allows easy selection from a data.frame, which you can use to select the female sample. See the file `vision-sieve.R` for this and more.

R

vision-sieve.R

```
library(vcd)
women <- subset(VisualAcuity, gender=="female", select=-gender)
structable(~right + left, data=women)
sieve(Freq ~ right + left,  data = women,
      gp=shading_Friendly, labeling=labeling_values)
```

2.3 Observer agreement

Westlund & Kurland (1953) reported data on the diagnosis of multiple sclerosis (MS): two samples of patients, one from Winnipeg and one from New Orleans, were each rated by two neurologists (one from each city) in four diagnostic categories: Certain, Probable, Possible, and Doubtful. The questions of interest here concern how well the neurologists agree on diagnosis for these two samples of patients.

Tasks & questions

1. Obtain and test Cohen's κ for each of the samples, using both equal spacing (Cicchetti-Allison) weights and inverse-square (Fleiss-Cohen) weights. What do you conclude about the strength of agreement in the two samples?
2. Obtain agreement charts for each of the samples. Visually, how does the strength of agreement shown in these charts compare to the κ measures?
3. Is there any evidence of difference between the two neurologists in their frequency of use of the diagnostic categories (marginal heterogeneity)?
4. Test whether the strength of agreement differs between the two samples (only in SAS).

SAS

For SAS, the data are contained in the file `msdiag.sas`, as a single data set (`msdiag`) with variables `Patients`, `N_rating` and `W_rating`. With `PROC FREQ`, a stratified analysis by `patients` gives you most of the information to answer the questions concerning agreement. For agreement plots, with the `agreeplot` macro you need to subset the data to select each sample. The statements below are contained in the file `msdiag-agree.sas`.

msdiag-agree.sas

```
%include catdata(msdiag);
*-- Agreement, separately, and controlling for Patients;
proc freq data=msdiag;
  tables patients * N_rating * W_rating / norow nocol nopct agree plots=all;
  test kappa;
  weight count;
run;

data Winnipeg;
  set msdiag;
  where (patients="Winnipeg");
%agreeplot(data=Winnipeg, var=N_rating W_rating, title=Winnipeg patients);

data NewOrleans;
  set msdiag;
  where (patients="New Orleans");
%agreeplot(data=NewOrleans, var=N_rating W_rating, title=New Orleans patients);
```

In R, the data are contained in `MSPatients`, a $4 \times 4 \times 2$ table. Patient is the third dimension, so you can use subscripting on the table to select the two samples separately. The statements below are contained in the file `msdiag-agree.R`.

msdiag-agree.R

R

```
library(vcd)
Kappa(MSPatients[,,1])  # use ,weights="Fleiss" for F-C weights
Kappa(MSPatients[,,2])

agreementplot(t(MSPatients[,,1]), main = "Winnipeg Patients")
agreementplot(t(MSPatients[,,2]), main = "New Orleans Patients")
```

2.4 Correspondence analysis

Correspondence analysis provides one way to visualize association between variables, both for nominal and ordinal categories. Here, we examine the relation between mental health and partents' SES. For a 4×6 table, the total Pearson χ^2 can be accounted for in $\min(r-1, c-1) = 3$ dimensions, but we hope for a simpler explanation.

Tasks & questions

1. Carry out a correspondence analysis of these data. From the printed output, how many dimensions seem necessary here?
2. Interpret the results of the graph in terms of the question of whether both mental and ses can be considered to be linearly spaced.

For SAS, the statements that carry out these steps are contained in the file `mental-ca.sas`.

mental-ca.sas

SAS

```
%include catdata(mental);
data mental;
    set mental;
    format mental mental. ses ses.;
run;
```

Use the `corresp` macro to carry out the analysis and produce the graph.

```
*-- Using the corresp macro;
%corresp(data=mental,
    tables = mental / ses,
    weight = count, htext=1.3);
```

In SAS 9.3, an equivalent plot can be obtained directly with `PROC CORRESP` (via SAS ODS Graphics).

```
*-- Using SAS 9.3 ODS Graphics;
proc corresp short data=mental;
tables mental, ses;
    weight count;
    run;
```

For R, the statements that carry out these steps are contained in the file `mental-ca.R`. Begin by creating a two-way table, then use `ca()` for numerical results and `plot()` to graph the result.

mental-ca.R

R

```
library(vcdExtra)
library(ca)
Mental.tab <- xtabs(Freq ~ mental+ses, data=Mental)
ca(Mental.tab)
plot(ca(Mental.tab))
```

3 Mosaic displays and loglinear models

3.1 Mosaic displays web applet

A web application to fit loglinear models and produce mosaic displays is available at `http://datavis.ca/online/mosaics/`. It is not as flexible as `PROC GENMOD` in SAS or the combination of `loglm()` and `mosaic()` in R, but is easy to use and contains several sample data sets, including the Berkelely data. Moreover, you can use it from home with your own data, uploaded from a file, or entered into a form.

Tasks & questions

1. Navigate to the Mosaic Displays web page, select the Berkeley Admission Data from the Sample datasets list, and press **SelectData**.

2. To illustrate, we will fit the model $[AD][GD]$ of conditional independence (Exercise 3 in Section 3.2), as follows:

   (a) In the Variable order box enter: `gender admit dept`.
   (b) Select `CONDIT` for FitType
   (c) (You may wish to explore the Analysis Options and Display Options help page.)
   (d) Press **GetData**
   (e) Interpret each display in relation to the χ^2 tests printed above.
   (f) What happens if you click on a tile in a mosaic?

3.2 Three-way+ tables

In this exercise you will exmine fitting several different loglinear models for the data on admissions to graduate programes in Berkeley. We will use the GLM approach here.[3] The table variables are `dept`, `gender` and `admit`, abbreviated as D, G and A. Note that admission is considered the response variable here, while gender and department are explanatory, so any reasonable model must include the $[GD]$ association between gender and department.

Tasks & questions

1. Fit the homogeneous association model $[AD][GD][AG]$ as generalized linear model for log frequency, and obtain tests for each term as well as the test for residual association. Does it appear that there is an association between admission and gender?
2. Interpret the model– what does each term mean? What does the test for residual association mean?
3. Now fit the reduced model $[AD][GD]$ of conditional independence of admission and gender given department, $A \perp G \,|\, D$. Obtain the residuals from this model and display these in a mosaic plot. What do you conclude?
4. It appears that conditional independence fits well, except in Department A. Fit a model that allows an association of admission and gender only in Department A.

The following steps will be helpful with SAS. If you need more help, see the statements in the file `berkeley-glm.sas`.

berkeley-glm.sas

SAS

- For display purposes it is useful to apply format statements to use as value labels for the table variables.

[3] As explained in the lecture, such models can be fit directly as loglinear models for the data in contingency table form, or as generalized linear models (GLM) with a log link and a Poisson distribution for the frequencies. In SAS, this amounts to the difference between `PROC CATMOD` vs. `PROC GENMOD`. In R, the corresponding functions are `loglm()` and `glm()`.

```
%include catdata(berkeley);
*-- Apply formats to use value labels for numeric variables;
data berkeley;
   set berkeley;
   format dept dept. admit admit. gender $sex.;
```

- The model of homogeneous association model $[AD][GD][AG]$ (no 3-way association) can be specified using `"|"` notation as shown below.

```
*-- Fit model of homogeneous associations:  [AD][GD][AG];
proc genmod data=berkeley;
   class dept gender admit;
   model freq = dept|gender|admit@2 / dist=poisson type3 wald;
run;
```

- The step below shows the pattern to (a) fit a model, (b) obtain residuals in an `obstats` data set and (c) produce a mosaic plot for the model.

```
*-- Fit model of conditional independence:  [AD][GD];
proc genmod data=berkeley;
   class dept gender admit;
   model freq = dept|gender dept|admit / dist=poisson obstats;
   ods output obstats=obstats;
run;

%mosaic(data=obstats, vorder=admit gender dept, count=freq,
        resid=streschi, cellfill=dev, split=H V,
        title=Model: [AdmitDept] [GenderDept]);
```

- You can fit a model allowing an association only in department A by defining a dummy variable that causes these cells to fit perfectly. Add this term to the `model` statementn `PROC GENMOD` and see what happens.

```
data berkeley;
   set berkeley;
   dept1AG = (gender='F') * admit * (dept=1);
```

In R, all of these models can be fit using `glm()`; the `loglm()` function cannot handle models with special terms (like `dept1AG`), but makes it quite easy to produce mosaic displays. The statements below are in the file `berkeley-glm.R`. In R model formulas, `()^2` means "all main effects and interactions up to order 2".

R

berkeley-glm.R

```
library(vcd)
data("UCBAdmissions")
structable(Dept ~ Admit+Gender,UCBAdmissions)

## conditional independence in UCB admissions data
berk.mod1 <- loglm(~ Dept * (Gender + Admit), data=UCBAdmissions)
berk.mod1
mosaic(berk.mod1, gp=shading_Friendly)

## all two-way model
berk.mod2 <-loglm(~(Admit+Dept+Gender)^2, data=UCBAdmissions)
berk.mod2
mosaic(berk.mod2, gp=shading_Friendly)
```

```
# compare models
anova(berk.mod1, berk.mod2)
```

3.3 Survival on the Titanic

These exercises examine the fitting of various loglinear models to data about survival on the *Titanic*, a 4-way table giving the cross-classification of 2201 passengers and crew, according to

- Gender (G): M vs. F
- Age (A): Adult vs. Child
- Class (C): 1st, 2nd, 3rd, Crew
- Survival (S): Died vs. Survived

1. One slight complication here is that there are 8 cells with zero frequencies. Four of these (male and female children in 1st and 2nd class who died) should be considered *sampling zeros*, but 4 (children among the crew) should probably be considered *structural zeros*– cells where data could not occur. In these analyses, you can treat these all as sampling zeros by adding a small number to each cell.
2. It is natural to consider Survival as the natural response variable, and the remaining variables as explanatory. Therefored, all models should include the high-order term among Age Gender and Class. Therefore, the minimal "null model" is [AGC][S], which asserts that survival is jointly independent of Age, Gender and Class. Fit this model, and obtain a mosaic plot. Interpret the pattern of the residuals in this mosaic plot.
3. Fit a "main effects" model for survival, [AGC][AS][GS][CS], that includes an association of survival with each of age gender and class. Is this an adequate fit? What does the pattern of residuals tell you about remaining associations?
4. What model would you use to allow an interaction of Age and Gender in their effect on Survival? Fit this model as above, and obtain the mosaic plot.

For SAS, the statements for this exercise are in the file `titanic-loglin.sas`, but you should try to figure them out for yourself first. The table variables are named `sex`, `age`, `class`, `survive`.

titanic-loglin.sas

SAS

- Adjusting for 0 cells.

```
%include catdata(titanic);
data titanic;
   set titanic;
   count = count + .5;
```

- Fitting and graphing the null model. Note how the high-order term [AGC] is specified using | notation.

```
title '[AGC][S]: Baseline model';
proc genmod data=titanic;
   class age sex class survive;
   model count=age|sex|class  survive
           / dist=poisson type3 obstats;
   ods output obstats=obstats;

%mosaic(data=obstats, vorder=class sex age survive,
        resid=reschi, title=[AGC][S]: Baseline model);
```

In R, it is easiest here to use the `loglm()` function to fit a model, and the `plot()` method for loglinear objects to obtain the mosaic plot. The statements below are contained in the file `titanic-loglin.R`.

R

titanic-loglin.R

```
library(vcd)
data(Titanic)

Titanic <- Titanic + 0.5   # adjust for 0 cells
titanic.mod1 <- loglm(~ (Class * Age * Sex) + Survived, data=Titanic)
titanic.mod1
plot(titanic.mod1, main="Model [AGC][S]")

titanic.mod2 <- loglm(~ (Class * Age * Sex) + Survived*(Class + Age + Sex),
                      data=Titanic)
titanic.mod2
plot(titanic.mod2,  main="Model [AGC][AS][GS][CS]")

titanic.mod3 <- loglm(~ (Class * Age * Sex) + Survived*(Class + Age * Sex),
                      data=Titanic)
titanic.mod3
plot(titanic.mod3,  main="Model [AGC][AS][GS][CS][AGS]")

# compare models
anova(titanic.mod1, titanic.mod2, titanic.mod3, test="chisq")
```

3.4 Ordinal factors and structured associations

When some of the variables in a contingency table represent *ordered* categories, we can usually gain both *power* and *parsimony* by fitting models that take ordinality into account. We gain power by testing a more specific hypothesis than just a general association; we gain parsimony by fitting fewer parameters.

3.4.1 Mental health and SES

In this exercise, you will examine several models for Mental-Health-SES data described in the lecture. In this two-way table, parents SES and child mental health status are both ordered factors. Simple ways of handling ordinal variable involve assigning *scores* to the table categories, and the simplest cases are to use *integer* scores, either for the row variable ("column effects" model), the column variable ("row effects" model), or both ("uniform association" model).

Tasks & questions

1. Fit the independence model to these data. How well does it fit? Make a mosaic plot showing departures (residuals) from independence. What do you see here that suggests a simpler description of the association based on ordinality?
2. Assign integer scores to both parents SES and child mental health status. Fit the model of uniform association. How well does it fit, compared to the independence model? Make a mosaic plot showing residuals from this model, and compare with that for the independence model.
3. If you have time, try also fitting the column effects model, using scores for just mental health status, and the row effects model, using scores for just parents SES.

For SAS, most of the program statements for this exercise are contained in the file `mentgen2.sas`. Here are a few explanatory notes:

SAS

mentgen2.sas

- In `PROC GENMOD`, a given variable can be *either* categorical (a factor) — when declared in a `CLASS` statement or numeric (a quantitative covariate), but not both. To allow a factor to be treated as numeric in an association term, simply make a copy of that variable under a different name:

```
%include catdata(mental);
*-- Create numeric variables for row & column effects;
data mental;
   set mental;
   m_lin = mental;
   s_lin = ses;
   format mental mental. ses ses.;
```

- Fit the independence model as follows, obtaining an output `obstats` data set of fitted values, residuals, etc. that can be passed to the `mosaic` macro.

```
proc genmod data=mental;
   class mental ses;
   model count = mental ses / dist=poisson obstats residuals;
   title 'Independence';
   ods output obstats=obstats;
```

- Visualize the remaining associations (residuals) with the `mosaic` macro.

```
%mosaic(data=obstats, vorder=Mental SES, resid=stresdev,
        title=Mental Impairment and SES: Independence, split=H V,
        cellfill=dev 0.5, htext=2.0);
```

- Include one or both of the numeric versions of the table variables in an interaction term. For example, the model of uniform association, with a linear $\times$ linear association is fit and visualized as follows:

```
proc genmod data=mental;
   class mental ses;
   model count = mental ses m_lin*s_lin / dist=poisson obstats residuals;
   title 'Lin x Lin';
   ods output obstats=obstats;
run;
%mosaic(data=obstats, vorder=Mental SES, resid=stresdev,
        title=Linear x Linear, split=H V, cellfill=dev 0.5, htext=2.0);
```

For R the steps are similar. We use the data set `Mental` in the `vcdExtra` package. These models are fit using `glm()`, and visualized using `mosaic.glm()` applied to the `glm` object. See the file `mental-glm.R` for these and other steps.

R

mental-glm.R

- Independence model:

```
library(vcdExtra)
data(Mental)
indep <- glm(Freq ~ mental+ses,
                family = poisson, data = Mental)
mosaic(indep,residuals_type="rstandard", labeling=labeling_residuals)
```

- Other models: Row effects, col effects and uniform association– Create numeric equivalents of the table variables, then fit models with interactions using the numeric scores.

```
Cscore <- as.numeric(Mental$ses)
Rscore <- as.numeric(Mental$mental)

# row effects model (mental)
roweff <- coleff <- glm(Freq ~ mental + ses + mental:Cscore,
                family = poisson, data = Mental)

linlin <- glm(Freq ~ mental + ses + Rscore:Cscore,
                family = poisson, data = Mental)
```

- Visualize a given model with `mosaic()` as shown above for the independence model.

3.4.2 Distant vision: quasi-independence, quasi-symmetry

In Section 2.2 we examined the Distant-Vision data for women with a sieve diagram. Here, we will consider several specialized models that range between the independence model and the saturated model. The goal is to find a relatively simple model, but one that accounts for the association.

The model of quasi-independence ignores the diagonal cells. The symmetry model tests whether the pattern of association is symmetric around the diagonal cells. Quasi-symmetry tests for symmetry, while allowing for differences in the marginal frequencies.

Tasks & questions

1. Fit the models of independence, and quasi-independence ignoring the diagonal cells. Compare the goodness of fit of the models numerically and with mosaic displays.
2. Fit the models of symmetry and quasi-symmetry (symmetric association, but allowing for differences in marginal frequencies). Compare the goodness of fit of the models numerically and with mosaic displays.
3. What do you conclude? Which model provides the *simplest, adequate* explanation of the relation of the relation between visual acuity in the two eyes?

In SAS, recall that the data set `women` has numeric factors, `left` and `right` for the eye grades of the two eyes. The key to fitting these models is to assign new variables corresponding to the diagonal cells `diag`, and to the diagonally symmetric ones `diag`. The statements below are included in the file `vision-quasi.sas`, along with others not shown here.

SAS

vision-quasi.sas

```
%include catdata(vision);

data women;
  set women;
  if left=right then do;
    diag=left;
    symmetry=-left;
    end;
  else do;
    diag=0;
    symmetry = abs(left-right);
    end;
```

- Independence and quasi-independence: The pattern is (a) fit the model with `PROC GENMOD`, (b) obtain the `obstats` data set with residuals, (c) plot the data and residuals from the model with the `mosaic` macro.

```
title 'Independence model (women)';
proc genmod data=women;
  class Right Left;
  model Count = Left Right /
    dist=poisson link=log obstats residuals type3;
  ods exclude obstats;
  ods output obstats=obstats;
run;
%mosaic(data=obstats, vorder=Right Left, split=H V, htext=1.9,
        resid=reschi, title=%str(Independence, G2(9)=6671.5));

title 'Quasi-independence model (women)';
proc genmod data=women;
  class Right Left diag;
  model Count = Left Right diag /
    dist=poisson link=log obstats residuals type3;
  ods exclude obstats;
  ods output obstats=obstats;
run;
%mosaic(data=obstats, vorder=Right Left, split=H V, htext=1.9,
        resid=reschi, title=%str(Quasi-Independence, G2(5)=199.1));
```

- Symmetry and quasi-symmetry: Use `model Count = symmetry` for strict symmetry and `model Left Right symmetry` quasi symmetry in the above.

In R, you can fit these models with `glm()` or `gnm()` in the `gnm` package. The `gnm` package provides special functions, `Diag()`, `Symm()` and others that considerably extend the range of loglinear models for structured tables. See the vignette, "Generalized nonlinear models in R," accessible in R as `vignette("gnmOverview" package="gnm")` for more details. At present `mosaic()` doesn't quite work with *all* of these more general models. The statements below are included in the file `vision-quasi.R`. vision-quasi.R

R

```
library(vcdExtra)
library(gnm)
women <- subset(VisualAcuity, gender=="female", select=-gender)

indep <- glm(Freq ~ right + left,  data = women, family=poisson)
mosaic(indep, main="Vision data: Independence (women)"  )

quasi.indep <- glm(Freq ~ right + left + Diag(right, left),
                   data = women, family = poisson)
mosaic(quasi.indep, residuals_type="rstandard", gp=shading_Friendly,
       main="Quasi-Independence (women)"  )

symmetry <- glm(Freq ~ Symm(right, left),
                   data = women, family = poisson)
mosaic(symmetry, residuals_type="rstandard", gp=shading_Friendly,
       main="Symmetry model (women)"  )
```

```
quasi.symm <- glm(Freq ~ right + left + Symm(right, left),
                  data = women, family = poisson)
mosaic(quasi.symm, residuals_type="rstandard", gp=shading_Friendly,
       main="Quasi-Symmetry model (women)")
```

Which model is "best"? You can use `anova()` to compare a collection of *nested* models:

```
# model comparisons: for *nested* models
 anova(indep, quasi.indep, quasi.symm, test="Chisq")
 anova(symmetry, quasi.symm, test="Chisq")
```

The `LRstats()` in the `vcdExtra` package also gives a compact summary of goodness-of-fit statistics for a set of related models (a `glmlist`). AIC and BIC statistics penalize larger models (more *df*), and smaller is better for both.

```
# model summaries, with AIC and BIC
models <- glmlist(indep, quasi.indep, symmetry, quasi.symm)
LRstats(models)
```

4 Logit models and logistic regression

Logit models are just a special case of logistic regression models in which *all* predictors are discrete. The only reasons for treating them specially are pedagogical: (a) every logit model for a binary response is equivalent to a loglinear model; (b) it is then a simple step to logistic regression models that include continuous predictors.

4.1 Logit models

It's time to have another look at the Berkeley data, from the perspective of logit models. In these models, with Admit as a response, we just specify the factors that Admit depends upon (Dept?, Gender?).

Tasks & questions

1. Fit a model in which Admit depends only on Dept. (The equivalent loglinear model is [AD][DG], which includes the association of Department and Gender.)
2. Fit the main effects model, Admit ∼ Dept + Gender. Plot the observed and fitted logits under this model.

In SAS, logit models are most easily fit with `PROC CATMOD`, and graphical displays of the fitted model most easily obtained with the `CATPLOT` macro. For this purpose, the option `out=predict` on the `response` statementroduces an output data set continaining fitted values and standard errors. The statements below are contained in the file `berkeley-logit.sas`, which includes some additional plotting statements to make plots prettier.

SAS

berkeley-logit.sas

- Effect of Dept only:

```
%include catdata(berkeley);
*-- logit (Admit) ~ Dept;
proc catmod order=data data=berkeley;
   format dept dept. ;
   population dept gender;
   weight freq;
   response / out=predict;
   model admit = dept  / ml noiter noprofile title="Model (AD, DG)";
run;
```

- Main effects of both Dept and Gender:

```
proc catmod order=data data=berkeley;
   weight freq;
   response / out=predict;
   model admit = dept gender / ml noiter noprofile title="Model (AD, AG, DG)" ;
  run;
```

- Plotting the observed and fitted values (on the logit scale) with the `CATPLOT` macro:

```
%catplot(data=predict, class=gender, x=dept, type=FUNCTION, z=1.96);
```

In R, the parallel analysis as a logit model is easiest if the data `UCBAdmissions` is first be converted from a $2 \times 2 \times 6$ table to a `data.frame` in frequency form. Models can then be expressed for the variable `Admit`, using the `Freq` variable as a weight. The statements below are contained in the file `berkeley-logit.R`.

R

berkeley-logit.R

- Effect of Dept only:

```
library(car)    # for Anova()
data(UCBAdmissions)
UCB.df <- as.data.frame(UCBAdmissions)
berk.mod1 <- glm(Admit=="Admitted" ~ Dept, data=UCB.df,
                 weights=UCB.df$Freq, family="binomial")
Anova(berk.mod1, test="Wald")
summary(berk.mod1)
```

- Adding a main effect of gender:

```
berk.mod2 <- glm(Admit=="Admitted" ~ Dept+Gender, data=UCB.df,
                 weights=UCB.df$Freq, family="binomial")
Anova(berk.mod2, test="Wald")
summary(berk.mod2)
```

- Plotting fitted values. The `effects` package is explored in more detail later in this section.

```
library(effects)   ## load the effects package
berk.eff2 <- allEffects(berk.mod2)
plot(berk.eff2)
plot(effect('Dept:Gender', berk.mod2), multiline=TRUE)
```

Note: The `effects` package issues Warnings when you ask to plot a higher-order term not contained in the model. This is OK; the correct plots should be produced.

4.2 Logistic regression

4.2.1 Arthritis treatment data

In previous lectures, I used the Arthritis data, but always ignoring the covariate Age, so I could treat it as a contingency table, Sex × Treatment × Improve. Here, we will fit logistic regression models, but only for a binary response,

$$\text{better} = \begin{cases} 0, & \text{improve} = \text{'None'} \\ 1, & \text{improve} = \text{'Some','Marked'} \end{cases}$$

Tasks & questions

1. Fit a logistic model predicting $\Pr(better = 1)$ from Age alone. Make a graph to visualize how better changes with Age.
2. Add the effects of Sex and and Treatment to this model. Make sure you understand how these effects are parameterized in your model. Interpret the coefficients for Age, Sex and Treatment on the probability of improvement.
3. Make a reasonable graph to visualize the effects of Age, Sex and Treatment on the probability of improvement.
4. Consider the possibilities there may be two-way interactions among Age, Sex and Treatment and the effect of Age may be non-linear (e.g., Age, Age^2). Fit a model that tests for these additional terms over and above the main effects.

For SAS, the variable `better` has been defined in the data step that creates the `arthrit` data set. Use the option `descending` to model the probability `better=1`. In SAS 9.3, the `effectplot` statementproduces reasonable plots either on the probability scale or logit scale. The statements below are contained in the file `arthritis-logistic.sas`.

SAS

arthritis-logistic.sas

- Logistic regression on age alone:

```
%include catdata(arthrit);
title 'Simple logistic regression on age';
proc logistic data=arthrit descending;
   model  better = age ;
   effectplot fit / obs(jitter(y=0.02));
   effectplot fit / obs(jitter(y=0.02)) link;  * display on logit scale;
run;
```

- Fit the main effects model. Note the use of the `ref=` option to assign reference categories for Sex and Treat, and the `output` statemento obtain a data set for plotting with the `meanplot` macro.

```
title 'main effects model';
proc logistic data=arthrit descending;
   class sex (ref=last) treat (ref=first) / param=ref;
   model  better = sex  treat  age ;
   output out=results p=prob l=lower u=upper
          xbeta=logit stdxbeta=selogit / alpha=.33;
   effectplot slicefit (sliceby=treat) / at(sex=all) clm alpha=0.33;
   effectplot interaction (x=treat sliceby=sex) / at(age=30 60) clm alpha=0.33 noobs;
        run;
*-- or use meanplot macro;
%meanplot(data=results, response=prob, class=age treat sex, pmean=no);
```

- To test potential high-order terms, you can use forward selection, starting with the main effects model.

```
title 'screening for higher-order effects';
proc logistic data=arthrit descending;
   class sex(ref=last) treat(ref=first) / param=ref;
   model  better = age sex treat
                   age | sex | treat @2  age*age
                   / selection=forward
                     slentry=0.1   /* be a bit liberal */
                     start=3 ;     /* Start after all main effects */
   effectplot slicefit (sliceby=treat) / at(sex=all)clm alpha=0.33;
   title2 'Testing all interactions via forward selection';
        run;
```

In R, use the `Arthritis` data in the `vcd` package. The variable `Better` must be calculated as shown below from `Improved`. Logistic regression models are fit with `glm()`, and fitted values can be obtained with `predict()` for the `glm` object. The statements below are contained in the file `arthritis-logistic.R`.

R

arthritis-logistic.R

- Logistic regression on age alone:

```
library(vcd)
library(car)
data(Arthritis)

# define Better
Arthritis$Better <- Arthritis$Improved > 'None'

# simple linear regression
```

```
arth.mod0 <- glm(Better ~ Age, data=Arthritis, family='binomial')
anova(arth.mod0)

# plot, with +-1 SE
plot(Better ~ Age, data=Arthritis, ylab="Prob (Better)")
pred <- predict(arth.mod0, type="response", se.fit=TRUE)
ord <-order(Arthritis$Age)

lines(Arthritis$Age[ord], pred$fit[ord], lwd=3)
upper <- pred$fit + pred$se.fit
lower <- pred$fit - pred$se.fit
lines(Arthritis$Age[ord], upper[ord], lty=2, col="blue")
lines(Arthritis$Age[ord], lower[ord], lty=2, col="blue")
```

- Fitting the main effects model:

```
# main effects model
arth.mod1 <- glm(Better ~ Age + Sex + Treatment , data=Arthritis,
                 family='binomial')
Anova(arth.mod1)

# same, using update()
arth.mod1 <- update(arth.mod0, . ~ . + Sex + Treatment)
Anova(arth.mod1)
```

- Forward selection:

```
# forward selection from the main effects model
step(arth.mod1, direction="forward", scope=.~ (Age+Sex+Treatment)^2 + Age^2)
```

4.2.2 Volunteering for psychology experiments

Cowles and Davis (1987) examined personality factors that might relate to whether an individual volunteered to participate in a psychology experiment. The predictors used here are scales of Neuroticism and Extraversion from the Eysenck personality inventory, plus Sex of the participant. There are 1421 observations.

Tasks & questions

1. Fit a main effects model predicting Pr (Volunteer = "Yes") from Sex, Neuroticism and Extraversion. How well does this model fit?
2. Fit a model that includes all two-way interactions among Sex, Neuroticism and Extraversion. How well does this model fit? Which interaction terms appear not to contribute to prediction?
3. Fit a reduced model including only the important interaction(s) from the all two-way model. What do you conclude?

In SAS, the data are available in `cowles.sas` and some statements for this exercise in the file `cowles-logistic.sas`. The statements below will get you started with the main effects model. Note the use of the `descending` option to model the event `Volunteer=1` ("yes"). For logistic models, the `lackfit` option on the `model` statementgives the Hosmer-Lemeshow lack of fit test

SAS

cowles-logistic.sas

```
%include catdata(cowles);
*-- apply formats to Sex and Volunteer;
data cowles;
```

```
  set cowles;
  format Sex sex. Volunteer volun.;
proc print data=cowles(obs=20);
  run;

*-- main effects model;
proc logistic data=cowles descending ;
  class Sex;
  model Volunteer = Sex  Extraver  Neurot /  lackfit ;
  run;
```

For other models, you can use `|` notation in the `model` statement here, `@` forms all high-order terms up to a given degree. Thus, the all two-way model can be specified as

```
*-- all two-way model;
proc logistic data=cowles descending ;
  class Sex;
  model Volunteer = Sex | Extraver | Neurot @2 /  lackfit ;
  run;
```

For R, the data frame `Cowles` is contained in the `car` package, which also has an `Anova()` function providing Type II and Type III tests, more useful than the sequential Type I tests provided by `anova()`. The R statements for this exercise are provided in the file `cowles-logistic.R`. For the main effects model, the commands are shown below.

R

cowles-logistic.R

```
library(car)       ## for Anova: type II tests
data(Cowles)
mod.cowles0 <- glm(volunteer ~ sex + neuroticism + extraversion,
    data=Cowles, family=binomial)
summary(mod.cowles0)
Anova(mod.cowles0)
```

4.2.3 Marijuana arrests data

In this exercise you will fit and examine graphically a more realistic (and complex) model. The data concern police treatment of individuals arrested in Toronto for simple possession of small quantities of marijuana.[4] Under these circumstances police have the option of releasing an arrestee with a summons to appear in court similar to a traffic ticket; alternatively, the individual may be brought to the police station for questioning and possible indictment.
The principal question of interest is whether and how the probability of release is influenced by the subjects race, age, and other characteristics. Beyond this, there is also interest in whether race interacts with other variables.
There are 5226 observations. The variables are:

`released` Whether the arrestee was released with a summons: "No" or "Yes"
`colour` The arrestee's race: "Black", "White".
`year` 1997 through 2002. Although numeric, we will treat `year` as a factor here.
`age` in years
`sex` "Female", "Male".

[4] This was part of a larger data set on racial profiling, featured in a series of articles in the *Toronto Star* newspaper, December, 2002.

`employed` "No", "Yes"
`citizen` "No", "Yes"
`checks` Number of police data bases (of previous arrests, previous convictions, parole status, etc. – 6 in all) on which the arrestee's name appeared

Tasks & questions

1. Fit a main effects logistic regression model predicting `released` from all the other variables.
2. Fit a model that includes all possible two-way interactions as well as all main effects. Which two-way interactions appear to be important?
3. Fit a reduced model that includes only the important two-way interactions as well as all main effects. Methods for model selection is a complex topic— here, we will only use backward selection from the all two-way model to illustrate one simple approach.

For SAS, the data are read from the file `arrests.sas`, and some code for this exercise is in the file `Arrests-logistic.sas`.

SAS

Arrests-logistic.sas

```
%include catdata(arrests);
*-- main effects model;
proc logistic data=arrests;
        class colour year sex citizen employed;
        model released = colour year age sex employed citizen checks;
        run;

*-- all two-way model, with backward elimination;
proc logistic data=arrests;
        class colour year sex citizen employed;
        run;
        model released = colour | year | age | sex | employed | citizen | checks@2 /
              selection=backward;
        run;
```

For R, the `Arrests` data are contained in the `effects` package, and some statements for this exercise are contained in the file `Arrests-logistic.R`. Note that, in model formulas, `~.` stands for "is modeled by all predictors"; `~.^2` is a short-hand for "all main effects plus two-way terms."

R

Arrests-logistic.R

```
library(effects)    # for Arrests data
library(car)        # for Anova()
data(Arrests)
Arrests$year <- as.factor(Arrests$year)

# all main effects
arrests.mod1 <- glm(released ~ ., family=binomial, data=Arrests)
Anova(arrests.mod1)

# all two-way effects
arrests.mod2 <- glm(released ~ .^2, family=binomial, data=Arrests)
Anova(arrests.mod2)
```

To eliminate unnecessary two-way terms, try `stepAIC()`, that tries to minimize the AIC statistic.

```
# backward selection, using AIC
arrests.step <- stepAIC(arrests.mod2, direction="backward")

anova(arrests.mod1, arrests.step, arrests.mod2, test="Chisq")
```

4.3 Effect plots

Effect plots are among the most useful graphical displays for understanding the pattern of effects in complex models. As explained in the lecture, the idea is relatively simple: to visualize a high-order term in a model, plot the predicted values ("adjusted means") for that term, averaging over all other factors *not included* in that term.

4.3.1 Volunteering for psychology experiments

Tasks & questions

1. Construct an effects plot to visualize the Neuroticism * Extraversion interaction for the Cowles data.
2. Give a verbal description of *how* probability of volunteering changes with both neuroticism and extraversion.

In SAS 9.3, effect plots are easily provided by the `effectplot` statementThe program statements for this exercise are contained in the file `cowles-effect.sas`.

cowles-effect.sas

```
%include catdata(cowles);
proc logistic data=cowles outest=parm  descending ;
    class Sex;
        model Volunteer = Sex   Extraver | Neurot / lackfit ;
        effectplot slicefit(x=Extraver sliceby=Neurot) / at(sex=1.5) noobs;
        effectplot slicefit(x=Neurot sliceby=Extraver) / at(sex=1.5) noobs;
        effectplot contour(x=Neurot y=Extraver) / at(sex=1.5) noobs;
        run;
```

In R, effect plots for a wide class of linear models are provided in the `effects` package. The function `allEffects()` creates the predicted effect values for all high-order terms in the model. These can then be plotted with the `plot()` method. The statements below are contained in the file `cowles-effect.R`. They show two different forms of effect plots for the `neuroticism:extraversion` effect.

R

cowles-effect.R

```
library(effects)   ## load the effects package
data(Cowles)
mod.cowles <- glm(volunteer ~ sex + neuroticism*extraversion,
    data=Cowles, family=binomial)
summary(mod.cowles)

eff.cowles <- allEffects(mod.cowles,
        xlevels=list(neuroticism=seq(0, 24, 6),
               extraversion=seq(0, 24, 8)))
#-- separate panels
plot(eff.cowles, 'neuroticism:extraversion', ylab="Prob(Volunteer)",
    ticks=list(at=c(.1,.25,.5,.75,.9)), layout=c(4,1))
#-- multiline plot
plot(eff.cowles, 'neuroticism:extraversion', multiline=TRUE,
    ylab="Prob(Volunteer)",
    key.args=list(x = .8, y = .9))
```

4.3.2 Marijuana arrests data

Here we examine effect plots for high-order terms in one model for `released` in the `Arrests` data. This model allows for interactions of `color` with both `year` and `age`. As mentioned above, these plots are easy to do with the `effects` package in R, but more difficult in SAS.

Tasks & questions

1. Fit the model that includes terms for `colour*year` and `colour*age`, as well as other variables as main effects.
2. Obtain an effect plot for the 3-way term `colour:year:age` (not included in the model). Interpret this plot in relation to how the probability of release varies with both Year and Age.
3. Obtain separate effect plots for the 2-way terms `colour:age` and `colour:year`

In SAS 9.3, we can again use the `effectplot` statement Note how the various options control the details. The statements below are contained in the file `Arrests-effect.sas`.

SAS

Arrests-effect.sas

```
proc logistic data=arrests descending;
   class colour year sex citizen employed;
   model released =
         colour|year colour|age sex employed citizen checks / clodds=wald;
   effectplot interaction (x=year sliceby=colour) / clm alpha=0.33 noobs yrange=clip;
   effectplot slicefit (x=age sliceby=colour) /
         clm alpha=0.33 obs(fringe jitter) yrange=(.7, 1);
   run;
```

The statements below are contained in the file `Arrests-effects.R`. Fit the model:

R

Arrests-effects.R

```
library(effects)
data(Arrests)
Arrests$year <- as.factor(Arrests$year)

arrests.mod <- glm(released ~ employed + citizen + checks
       + colour*year + colour*age, family=binomial, data=Arrests)
```

Three-way effect plot:

```
plot(effect("colour:year:age", arrests.mod, xlevels=list(age=15:45)),
  multiline=TRUE, ylab="Probability(released)", rug=FALSE)
```

Two-way effect plots. Note the difference between the style of plot with `multiline=TRUE` and `multiline=FALSE`.

```
plot(effect("colour:age", arrests.mod),
      multiline=TRUE, ylab="Probability(released)")

plot(effect("colour:age", arrests.mod),
      multiline=FALSE, ylab="Probability(released)")
```

For complex models, a convenient feature of the `effects` package is the ability to obtain the necessary statistics for *all effects* simultaneously, and plot various terms by selection from a menu.

```
arrests.effects <- allEffects(arrests.mod, xlevels=list(age=seq(15,45,5)))
plot(arrests.effects, ylab="Probability(released)")
```

4.4 Diagnostic plots

In this exercise, the goal is to explore diagnostic plots for generalized linear models, which are simple extensions of similar plots for ordinary linear models (regression and ANOVA). We return to the Berkeley data one more time.

Tasks & questions

1. Fit the model $[AD][GD]$ of conditional independence, as before.
2. Construct plots to show the distribution of residuals from this model and to identify influential cells in terms of residual and leverage.
3. Interpret what you see in these plots in relation to other plots of these data in previous exercises.

In SAS, you can use the `inflglim` and `halfnorm` macros, as well as the ODS Graphics facility in SAS 9.2+ to obtain a wide variety of diagnostic plots. See `berkeley-diag.sas`.

SAS

berkeley-diag.sas

- Using the `inflglim` and `halfnorm` macros. For these plots, begin by constructing a character variable that joins the factor levels into a single cell variable for labels in the plots.

```
%include catdata(berkeley);
*-- make a cell ID variable, joining factors;
data berkeley;
   set berkeley;
   cell = trim(put(dept,dept.)) ||
        gender ||
          trim(put(admit,yn.));
run;

*-- Conditional independence;
proc genmod data=berkeley;
   class dept gender admit;
   model freq = dept|gender dept|admit / dist=poisson obstats residuals;
   ods output obstats=obstats;
```

- Use the `inflglim` macro to produce influence plots for this model.

```
%inflglim(data=berkeley, class=dept gender admit,
        resp=freq, model=dept|gender dept|admit, dist=poisson, id=cell,
        gx=hat, gy=stresdev);

%inflglim(data=berkeley, class=dept gender admit,
        resp=freq, model=dept|gender dept|admit, dist=poisson, id=cell,
        gx=hat, gy=difdev);
```

- Use the `halfnorm` macro to produce a half-normal residual plot for this model, with a simulated 95% envelope for the mean response in the model.

```
%halfnorm(data=berkeley, class=dept gender admit,
   resp=freq, model=dept|gender dept|admit, dist=poisson, id=cell);
```

- Using ODS graphics with `plot=all`. This only works in SAS 9.2+, and the plots are all index plots– variable plotted against observation (cell) number.

```
proc genmod data=berkeley plots=all;
   class dept gender admit;
   model freq = dept|gender dept|admit / dist=poisson;
run;
```

In R, some of these plots can be obtained from the `plot()` method for `glm` objects. An influence plot is provided by `influencePlot()` in `car` package. See `berkeley-diag.R`.

berkeley-diag.R

R

```
library(car)
berkeley <- as.data.frame(UCBAdmissions)
cellID <- paste(berkeley$Dept, substr(berkeley$Gender,1,1), '-',
                substr(berkeley$Admit,1,3), sep="")
rownames(berkeley) <- cellID

berk.mod <- glm(Freq ~ Dept * (Gender+Admit), data=berkeley, family="poisson")
influencePlot(berk.mod, labels=cellID, id.n=3)
plot(berk.mod)
```

5 Polytomous response models

5.1 Proportional odds model

5.1.1 Arthritis data

In this exercise you will try to fit a proportional odds model to the Arthritis data, modeling the 3-category response Improve ("None", "Some", or "Marked") and using Sex, Treatment and Age as predictors.

Tasks & questions

1. Fit a main effects model predicting `improve` from Sex, Treatment and Age. Note that both Sex and Treatment are binary factors (CLASS variables in SAS), so the interpretation of the coefficients depends on how they are coded.
2. How well does the proportional odds model fit for these data?[5]
3. Interpret the coefficients for Sex, Treatment and Age in this model. For example, how would you interpret the coefficient for Sex in relation to the odds of a better outcome?
4. Obtain and plot predicted (fitted) values on either the probability scale or the logit scale. Interpret this plot in relation to the numerical results (coefficients, odds ratios, etc.)

For SAS, the steps are illustrated in the lecture notes, "Proportional odds model: Fitting and plotting." Below, I show just the steps to fit the model, and obtain an output data set with the fitted probabilities and logits. Note that Sex is coded so that Male is the reference category, and Treat so that Placebo is the reference category. The additional steps to produce nicely formatted plots are contained in the file `arthritis-propodds.sas`.

SAS

arthritis-propodds.sas

```
title 'Logistic Regression: Proportional Odds Model';
%include catdata(arthrit);

proc logistic data=arthrit descending;
   class sex (ref=last) treat (ref=first) / param=ref;
   model  improve = sex  treat  age ;
   output out=results p=prob l=lower u=upper
          xbeta=logit stdxbeta=selogit / alpha=.33;

proc print data=results(obs=6);
   id id treat sex;
   var improve _level_ prob lower upper logit;
   format prob lower upper logit selogit 6.3;
run;
```

In SAS 9.2+ it is easier to plot results using ODS graphics and the `effectplot` statement The code below is contained in `arthritis-propodds-ods.sas`

arthritis-propodds-ods.sas

```
proc logistic data=arthrit descending  ;
   class sex (ref=last) treat (ref=first) / param=ref;
   model  improve = sex  treat  age / clodds=wald expb;
   effectplot slicefit(sliceby=improve plotby=Treat) / at(sex=all) clm alpha=0.33 ;
   effectplot interaction(x=Treat sliceby=improve) / at(sex=all) noobs clm alpha=0.33;
run;
```

In R, we can fit the proportional odds model with `polr()` in the `MASS` package. There is

R

[5] Caution: The score test for the proportional odds model is known to be anti-conservative, i.e., it yields p-values that are too small in many cases.

no simple equivalent for the score test of the proportional odds assumption. The statements below are contained in the file `arthritis-propodds.R`.

arthritis-propodds.R

```
library(vcd)
library(car)          # for Anova()

arth.polr <- polr(Improved ~ Sex + Treatment + Age, data=Arthritis)
summary(arth.polr)
Anova(arth.polr)      # Type II tests
```

In R it is easy to obtain effect plots showing the fitted probabilities in relation to any of the terms in the model (or even terms *not* in the model).

```
library(effects)
arth.effects <- allEffects(arth.polr, xlevels=list(age=seq(15,45,5)) )
plot(arth.effects)  # visualize all main effects

# plot terms not in model
plot(effect("Sex:Age", arth.polr))
plot(effect("Treatment:Age", arth.polr))
```

5.2 Nested dichotomies

This exercise concerns data on women's labour-force participation collected at York University in a 1977 survey of the Canadian population. The three-level response has categories Not working outside the home ($n = 155$), working Part-time ($n = 42$) and working Full-time ($n = 66$). The predictors are Husband's income ($1000s), Presence of children in the household, and Region of Canada. Region turns out not to be an important predictor, so we will ignore it here.

Tasks & questions

1. The first question is whether the proportional odds model provides a reasonable account of these data— if so, it would give a simple description. Fit the proportional odds model, and test whether the proportional odds assumption is tenable (only in SAS).
2. Fit separate logistic models for the two nested dichotomies:
   - Working (full- or part-time) vs. Not Working; and
   - Working Full-time vs. Part-time.
3. Interpret the coefficients in the two models. What do they say about the effect of Husband's income and Children on the probabilities for the two dichotomies?
4. Plot the fitted probabilities for the 3 categories of the response in relation to Husband's income and Children.

For SAS, the data are contained in the data set `wlfpart`. The 3-level response here is `labor`, with values 1 (Full-time), 2 (Part-time), 3 (Not working). When the data are read in, two dichotomous responses are created: `working = labor < 3`, and `fulltime = labor=1` (but just for working women. The statements below (and more) are contained in the file `wlf-nested.sas`.

SAS

wlf-nested.sas

Fitting the proportional odds model:

```
%include catdata(wlfpart);
proc logistic data=wlfpart nosimple descending;
   model labor  = husinc children ;
   title2 'Proportional Odds Model for Fulltime/Parttime/NotWorking';
run;
```

Fitting and plotting the models for the nested dichotomies. Note how an `ODS OUTPUT` statementan be used to capture the global test statistics. See the `wlf-nested.sas` file for how to combine these to give overall tests for the 3-level nested dichotomy model. wlf-nested.sas

```
proc logistic data=wlfpart nosimple descending;
   model working = husinc children ;
   output out=resultw p=predict;
   ods output GlobalTests=gtests1;
   title 'Nested Dichotomies';
proc plot;
   plot predict * husinc = children;

proc logistic data=wlfpart nosimple descending;
   model fulltime = husinc children ;
   output out=resultf p=predict;
   ods output GlobalTests=gtests2;
proc plot;
   plot predict * husinc = children;
run;
quit;
```

In R, the data are in the data frame `Womenlf` in the `car` package. The 3-level response is called `partic` here, an unordered factor with values `fulltime`, `not.work` and `parttime`. One easy way to create the two dichotomous responses is with the `recode()` function in `car` package. The file `wlf-nested.R` contains the statements below, and also shows how to produce a plot of the fitted values. R wlf-nested.R

```
library(car)
data(Womenlf)
attach(Womenlf)
working <- recode(partic, " 'not.work' = 'no'; else = 'yes' ")
fulltime <- recode (partic,
    " 'fulltime' = 'yes'; 'parttime' = 'no'; 'not.work' = NA")
```

Fitting nested dichotomies models: We use treatment contrasts for `children`, so the coefficient represents the effect for having children.

```
contrasts(children)<- 'contr.treatment'
mod.working <- glm(working ~ hincome + children, family=binomial)
summary(mod.working)
mod.fulltime <- glm(fulltime ~ hincome + children, family=binomial)
summary(mod.fulltime)
Anova(mod.working)
Anova(mod.fulltime)
```

Obtaining predicted values for plotting: The general idea is to create a data frame with the combinations of the predictor values you want to see, and use the `predict()` method to get

the fitted values. For binomial `glm()` models, `type='response'` gives fitted values on the probability scale.[6]

```
# get fitted values for both sub-models
predictors <- expand.grid(hincome=1:45, children=c('absent', 'present'))

p.work <- predict(mod.working, predictors, type='response')
p.fulltime <- predict(mod.fulltime, predictors, type='response')
```

One slight complication: Because the model for the `fulltime` dichotomy is *conditional* on working, you need to do some calculations to obtain the *unconditional* probabilities for each of the three response categories.

```
# calculate unconditional probs for the three response categories
p.full <- p.work * p.fulltime
p.part <- p.work * (1 - p.fulltime)
p.not <- 1 - p.work
```

5.3 Generalized logits

The generalized (or multinomial) logit model is an alternative to using nested dichotomies for a polytomous reponse. One advantage is that you can fit this as a single model, without resorting to the special tricks used for nested dichotomies.
In using the generalized logit model, you should take care to specify the baseline response category, so you understand how to interpret the fitted coefficients. As always, plots of the fitted model (probabilities or logits) facilitate interpretation.
In this exercise, we continue with the data on women's labor participation from Section 5.2, but now fit a generalized logit model.

Tasks & questions

1. Fit the generalized logit model, using Not working as the baseline response category.
2. Examine the coefficients for Husband's income and Children in the comparisons of Part-time and Full-time vs. Not working. On which comparison to the predictors show a greater impact?
3. Plot the fitted probabilities of all three response categories over the range of Husbabd's income for both levels of Children. Compare this plot to the analogous plot for the nested dichotomies model.

In SAS, you can fit the generalized logit model for the polytomous response `labor` simply by specifying the option `link=glogit` on the `model` statement Use the `output` statemento obtain fitted values, both on the probability scale `p=predict` and on the logit scale `xbeta=logit`.

SAS

```
*-- Fit generalized logit model;
proc logistic data=wlfpart;
   model labor = husinc children / link=glogit;
   output out=results p=predict xbeta=logit;
```

In the output data set `results`, the three response categories are identified by a variable `_level_`. The simplest plot of the fitted probabilities is produced with `PROC PLOT` as shown below. See the file `wlf-glogit.sas` for details on plotting the fitted results with `PROC GPLOT` and better labels for the curves.

wlf-glogit.sas

[6] The default, `type='link'`, gives fitted values on the logit scale.

```
proc sort data=results;
        by children husinc _level_;

*-- simple plot;
proc plot data=results;
        plot predict * husinc = _level_ ;
        by children;
```

For R, it is helpful to first turn `partic` into an ordered factor, so that Not working is the baseline category. The multinomial logit model cannot be fit directly with `glm()`; it *can* be fit using the `multinom()` function in the `nnet` package, which also requires the `MASS` library. See the file `wlf-glogit.R` for more details on plotting the fitted results. Fitting the multinomial logit model:

R

wlf-glogit.R

```
library(car)
data(Womenlf)
attach(Womenlf)
participation <- ordered(partic,
    levels=c('not.work', 'parttime', 'fulltime'))
library(nnet)
library(MASS)
mod.multinom <- multinom(participation ~ hincome + children)
summary(mod.multinom, cor=F, Wald=T)
Anova(mod.multinom)
```

Obtaining predicted values for plotting: Here, we get fitted values on the probability scale with `type='probs'`.

```
predictors <- expand.grid(hincome=1:45, children=c('absent', 'present'))
p.fit <- predict(mod.multinom, predictors, type='probs')
```