SOLUTIONS MANUAL FOR

Discrete Data Analysis with R: Visualization and Modeling Techniques for Categorical and Count Data

by

Michael Friendly
and David Meyer

CRC Press
Taylor & Francis Group

SOLUTIONS MANUAL FOR

Discrete Data Analysis with R: Visualization and Modeling Techniques for Categorical and Count Data

by

Michael Friendly
and David Meyer

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20160303

International Standard Book Number-13: 978-1-4987-2590-3 (Ancillary)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Discrete Data Analysis with R: Solutions and Hints to Exercises

Contents

This document

This document is intended as an aid to instructors who wish to use *Discrete Data Analysis with R* in a course. It contains the text of the **Exercises** sections from all chapters, together with some solutions or hints for the various problems. Answers and commentary are indicated with the ★ symbol, and with text in this font.

Instructors should recognize that many questions are open-ended or admit to different approaches. In many cases we simply present one reasonable approach.

All R code for the book, and other materials are available on the web site, `http://ddar.datavis.ca`.

For the most part, R code in answers indicates required packages with `library()` or `require()`. This document assumes, however, that the following packages are loaded: AER, car, effects, MASS vcd, vcdExtra.

Chapter 1 Introduction

★ These questions are all conceptual, or based on judgment. No individual solutions are provided. In general, students should come up with some interesting examples related to the questions and explain why they consider them to be good or bad graphic or tabular displays.

Some other sources that students might consult are:

- The Gallery of Data Visualization, `http://datavis.ca/gallery/`. A categorized collection of some of the best and worst of statistical graphics.
- Junk Charts, `http://junkcharts.typepad.com/`, a blog by Kaiser Fung. There is also a list of related blogs on graphics and data visualization at `http://junkcharts.typepad.com/junk_charts/other-graphics-blogs.html`.
- Flowing Data, `http://flowingdata.com/`, by Nathan Yau. An eclectic collection of examples and blog posts encomposing a wide range from information visualization to statistical graphics.

Exercise 1.1 A web page, "The top ten worst graphs," `http://www.biostat.wisc.edu/~kbroman/topten_worstgraphs/` by Karl Broman lists his picks for the worst graphs (and a table) that have appeared in the statistical and scientific literature. Each entry links to graph(s) and a brief discussion of what is wrong and how it could be improved.

(a) Examine a number of recent issues of a scientific or statistical journal in which you have some interest. Find one or more examples of a graph or table that is a particularly bad use of display material to summarize and communicate research findings. Write a few sentences indicating how or why the display fails and how it could be improved.
(b) Do the same task for some popular magazine or newspaper that uses data displays to supplement the text for some story. Again, write a few sentences describing why the display is bad and how it could be improved.

Exercise 1.2 As in the previous exercise, examine the literature in recent issues of some journal of interest to you. Find one or more examples of a graph or table that you feel does a *good* job of summarizing and communicating research findings.

(a) Write a few sentences describing why you chose these displays.
(b) Now take the role of a tough journal reviewer. Are there any features of the display that could be modified to make them more effective?

Exercise 1.3 Infographics are another form of visual displays, quite different from the data graphics featured in this book, but often based on some data or analysis. Do a Google image search for the topic "Global warming" to see a rich collection.

(a) Find and study one or two images that attempt some visual explanation of causes and/or effects of global warming. Describe the main message in a sentence or two.
(b) What visual and graphic features are used in these to convey the message?

Exercise 1.4 The Wikipedia web page `en.wikipedia.org/wiki/Portal:Global_warming` gives a few data-based graphics on the topic of global warming. Read the text and study the graphs.

(a) Write a short figure title for each that would announce the conclusion to be drawn in a presentation graphic.
(b) Write a figure caption for each that would explain what is shown and the important graphical details for a reader to understand.

Exercise 1.5 The R Graph Gallery, `http://rgraphgallery.blogspot.com/`, contains a large collection of examples of graphs in R, tagged by type or content, together with the R code to produce them. Explore this collection for the terms (a) association plot (b) bar chart (c) categorical data (d) fluctuation diagram (e) mosaic plot Find one or two you particularly like and write a few sentences saying why you do.

Chapter 2 Working with Categorical Data

Exercise 2.1 The packages vcd (Meyer et al., 2015) and vcdExtra (Friendly, 2015) contain many data sets with some examples of analysis and graphical display. The goal of this exercise is to familiarize yourself with these resources.

You can get a brief summary of these using the function `datasets()` from vcdExtra. Use the following to get a list of these with some characteristics and titles.

```
> ds <- datasets(package = c("vcd", "vcdExtra"))
> str(ds, vec.len = 2)

'data.frame': 75 obs. of  5 variables:
 $ Package: chr  "vcd" "vcd" ...
 $ Item   : chr  "Arthritis" "Baseball" ...
 $ class  : chr  "data.frame" "data.frame" ...
 $ dim    : chr  "84x5" "322x25" ...
 $ Title  : chr  "Arthritis Treatment Data" "Baseball Data" ...
```

(a) How many data sets are there altogether? How many are there in each package?
★ `nrow()` gives the number of rows in a data frame. `table()` for a single variable gives the frequencies for each level.

```
> ds <- datasets(package=c("vcd", "vcdExtra"))
> nrow(ds)

[1] 75

> table(ds$Package)

     vcd vcdExtra
      33       42
```

(b) Make a tabular display of the frequencies by `Package` and `class`.
★ Use `table()`, but now for `Package` and `class`.

```
> table(ds$Package, ds$class)

           array data.frame matrix table
  vcd          1         17      0    15
  vcdExtra     3         23      1    15
```

(c) Choose one or two data sets from this list, and examine their help files (e.g., `help(Arthritis)` or `?Arthritis`). You can use, e.g., `example(Arthritis)` to run the R code for a given example.
★ Run the following types of commands:

```
> ?Arthritis          # Help Files
> ?Baseball           # Help Files
> example(Arthritis)  # Example Syntax/Analysis
> example(Baseball)   # Example Syntax/Analysis
```

Exercise 2.2 For each of the following data sets in the vcdExtra package, identify which are response variable(s) and which are explanatory. For factor variables, which are unordered (nominal) and which should be treated as ordered? Write a sentence or two describing substantitive questions of interest for analysis of the data. (*Hint*: use `data(foo, package="vcdExtra")` to load, and `str(foo)`, `help(foo)` to examine data set `foo`.)

(a) Abortion opinion data: *Abortion*
★ `Support_Abortion` is the response, `Sex` and `Status` are binary, nominal explanatory variables. From `help(Abortion)`, How does support for abortion depend on sex and status?

```
> data(Abortion, package="vcdExtra")
> str(Abortion)

 table [1:2, 1:2, 1:2] 171 152 138 167 79 148 112 133
 - attr(*, "dimnames")=List of 3
  ..$ Sex             : chr [1:2] "Female" "Male"
  ..$ Status          : chr [1:2] "Lo" "Hi"
  ..$ Support_Abortion: chr [1:2] "Yes" "No"
```

(b) Caesarian Births: *Caesar*
★ `Infection` is the response, `Risk`, `Antibiotics` and `Planned` are binary, nominal explanatory variables.

```
> data(Caesar, package="vcdExtra")
> str(Caesar)

 table [1:3, 1:2, 1:2, 1:2] 0 1 17 0 1 1 11 17 30 4 ...
 - attr(*, "dimnames")=List of 4
  ..$ Infection  : chr [1:3] "Type 1" "Type 2" "None"
  ..$ Risk       : chr [1:2] "Yes" "No"
  ..$ Antibiotics: chr [1:2] "Yes" "No"
  ..$ Planned    : chr [1:2] "Yes" "No"
```

(c) Dayton Survey: *DaytonSurvey*
★ In `DaytonSurvey`, the variables `cigarette`, `alcohol`, and `marijuana` can all be treated as response variables. `sex` and `race` are potential explanatory variables. Potentially interesting questions are how each of the responses depend on `sex` and `race`, and how they vary jointly.

```
> data(DaytonSurvey, package="vcdExtra")
> str(DaytonSurvey)
```

(d) Minnesota High School Graduates: *Hoyt*
★ `Status` is the response, `Rank`, `Occupation`, and `Sex` are explanatory variables. Both `Rank` and `Occupation` are ordinal. How does `Status` vary with `Rank`, `Occupation`, and `Sex`?

```
> data(Hoyt, package="vcdExtra")
> str(Hoyt)
```

Exercise 2.3 The data set *UCBAdmissions* is a 3-way table of frequencies classified by `Admit`, `Gender`, and `Dept`.

(a) Find the total number of cases contained in this table.
★ For a `table` object, just use `sum()`

```
> data(UCBAdmissions)
> sum(UCBAdmissions)

[1] 4526
```

(b) For each department, find the total number of applicants.
★ Use `margin.table(UCBAdmissions, 3)` to find the marginal total for the third dimension (`dept`).

```
> margin.table(UCBAdmissions, 3)

Dept
  A   B   C   D   E   F
933 585 918 792 584 714
```

(c) For each department, find the overall proportion of applicants who were admitted.
★

```
> ucb.df <- as.data.frame(UCBAdmissions)
> abd <- xtabs(Freq ~ Dept + Admit, data=ucb.df)
> prop.table(abd, 1)

    Admit
Dept Admitted Rejected
   A 0.644159 0.355841
   B 0.632479 0.367521
   C 0.350763 0.649237
   D 0.339646 0.660354
   E 0.251712 0.748288
   F 0.064426 0.935574
```

(d) Construct a tabular display of department (rows) and gender (columns), showing the proportion of applicants in each cell who were admitted relative to the total applicants in that cell.
★

Exercise 2.4 The data set *DanishWelfare* in vcd gives a 4-way, $3 \times 4 \times 3 \times 5$ table as a data frame in frequency form, containing the variable `Freq` and four factors, `Alcohol`, `Income`, `Status`, and `Urban`. The variable `Alcohol` can be considered as the response variable, and the others as possible predictors.

(a) Find the total number of cases represented in this table.
★ This is a data set in the form of a frequency data.frame, so sum the `Freq` variable

```
> data("DanishWelfare", package="vcd")
> sum(DanishWelfare$Freq)

[1] 5144
```

(b) In this form, the variables `Alcohol` and `Income` should arguably be considered *ordered* factors. Change them to make them ordered.
★ Use `ordered()` or `as.ordered()` on the factor variable. `str()` will then show them as `Ord.factor`.

```
> levels(DanishWelfare$Alcohol)

[1] "<1"  "1-2" ">2"

> DanishWelfare$Alcohol <- as.ordered(DanishWelfare$Alcohol)
> DanishWelfare$Income <- as.ordered(DanishWelfare$Income)
> str(DanishWelfare)

'data.frame': 180 obs. of  5 variables:
 $ Freq   : num  1 4 1 8 6 14 8 41 100 175 ...
 $ Alcohol: Ord.factor w/ 3 levels "<1"<"1-2"<">2": 1 1 1 1 1 1 1 1 1 1 ...
 $ Income : Ord.factor w/ 4 levels "0-50"<"50-100"<..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Status : Factor w/ 3 levels "Widow","Married",..: 1 1 1 1 1 2 2 2 2 2 ...
 $ Urban  : Factor w/ 5 levels "Copenhagen","SubCopenhagen",..: 1 2 3 4 5 1 2 3 4 5 ...
```

(c) Convert this data frame to table form, `DanishWelfare.tab`, a 4-way array containing the frequencies with appropriate variable names and level names.
★ Use `xtabs()` with `Freq` as the response.

```
> DanishWelfare.tab <-xtabs(Freq ~ ., data = DanishWelfare)
> str(DanishWelfare.tab)

 xtabs [1:3, 1:4, 1:3, 1:5] 1 3 2 8 1 3 2 5 2 42 ...
 - attr(*, "dimnames")=List of 4
  ..$ Alcohol: chr [1:3] "<1" "1-2" ">2"
  ..$ Income : chr [1:4] "0-50" "50-100" "100-150" ">150"
  ..$ Status : chr [1:3] "Widow" "Married" "Unmarried"
  ..$ Urban  : chr [1:5] "Copenhagen" "SubCopenhagen" "LargeCity" "City" ...
 - attr(*, "class")= chr [1:2] "xtabs" "table"
 - attr(*, "call")= language xtabs(formula = Freq ~ ., data = DanishWelfare)
```

(d) The variable `Urban` has 5 categories. Find the total frequencies in each of these. How would you collapse the table to have only two categories, `City`, `Non-city`?
★ `margin.table()` handles the first part; `collapse.table()` is designed for the second part. It is arguable whether `SubCopenhagen` should be considered City or NonCity.

```
> margin.table(DanishWelfare.tab, 4)

Urban
   Copenhagen SubCopenhagen     LargeCity          City       Country
          552           614           594          1765          1619

> DW2 <- vcdExtra::collapse.table(DanishWelfare.tab,
+                                 Urban=c("City","NonCity","City","City","NonCity"))
> head(ftable(DW2))

                                "Urban" "City" "NonCity"
 "Alcohol" "Income"  "Status"
 "<1"      "0-50"    "Widow"                10        10
                     "Married"             155       183
                     "Unmarried"            14        10
           "50-100"  "Widow"                29         7
                     "Married"             338       306
                     "Unmarried"            36        32
```

(e) Use `structable()` or `ftable()` to produce a pleasing flattened display of the frequencies in the 4-way table. Choose the variables used as row and column variables to make it easier to compare levels of `Alcohol` across the other factors.
★

Exercise 2.5 The data set *`UKSoccer`* in vcd gives the distributions of number of goals scored by the 20 teams in the 1995/96 season of the Premier League of the UK Football Association.

```
> data("UKSoccer", package = "vcd")
> ftable(UKSoccer)

     Away  0  1  2  3  4
Home
0          27 29 10  8  2
1          59 53 14 12  4
2          28 32 14 12  4
3          19 14  7  4  1
4           7  8 10  2  0
```

This two-way table classifies all $20 \times 19 = 380$ games by the joint outcome (Home, Away), the number of goals scored by the `Home` and `Away` teams. The value `4` in this table actually represents 4 or more goals.

(a) Verify that the total number of games represented in this table is 380.
★

```
> data("UKSoccer", package="vcd")
> sum(UKSoccer)

[1] 380

> margin.table(UKSoccer)

[1] 380
```

(b) Find the marginal total of the number of goals scored by each of the home and away teams.
★ Use `margin.table()` for each dimension:

```
> margin.table(UKSoccer, 1)

Home
  0   1   2   3   4
 76 142  90  45  27

> margin.table(UKSoccer, 2)

Away
  0   1   2   3   4
140 136  55  38  11
```

(c) Express each of the marginal totals as proportions.
★ Use `prop.table()` on the result of `margin.table()` for each dimension:

```
> prop.table(margin.table(UKSoccer, 1))

Home
       0        1        2        3        4
0.200000 0.373684 0.236842 0.118421 0.071053

> prop.table(margin.table(UKSoccer, 2))

Away
       0        1        2        3        4
0.368421 0.357895 0.144737 0.100000 0.028947
```

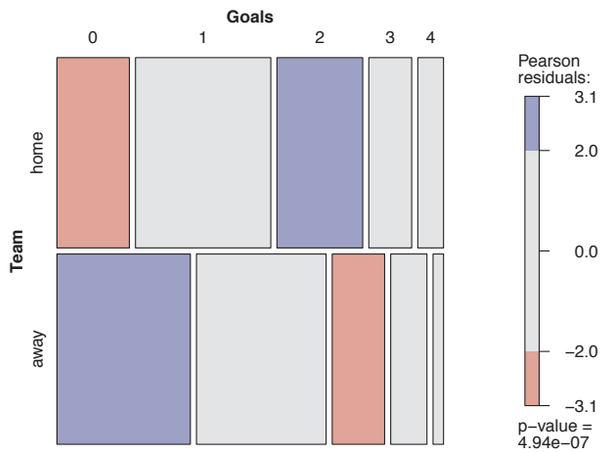
(d) Comment on the distribution of the numbers of home-team and away-team goals. Is there any evidence that home teams score more goals on average?
★ You could find the mean number of goals, weighted by their marginal frequencies. On average, home teams score about 0.4 more goals.

```
> weighted.mean(0:4, w=margin.table(UKSoccer,1))

[1] 1.4868

> weighted.mean(0:4, w=margin.table(UKSoccer,2))

[1] 1.0632
```

Graphically, you could also compare the marginal frequencies in a mosaic plot, or use `agreementplot()`.

```
> margins <- rbind(home=margin.table(UKSoccer,1),
+                  away=margin.table(UKSoccer,2))
> names(dimnames(margins)) <- c("Team", "Goals")
> margins
```

```
      Goals
Team     0   1  2  3  4
  home  76 142 90 45 27
  away 140 136 55 38 11

> mosaic(margins, shade=TRUE)
```



Exercise 2.6 The one-way frequency table *Saxony* in vcd records the frequencies of families with 0, 1, 2, ... 12 male children, among 6115 families with 12 children. This data set is used extensively in Chapter 3.

```
> data("Saxony", package = "vcd")
> Saxony

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

Another data set, *Geissler*, in the vcdExtra package, gives the complete tabulation of all combinations of `boys` and `girls` in families with a given total number of children (`size`). The task here is to create an equivalent table, `Saxony12` from the *Geissler* data.

```
> data("Geissler", package = "vcdExtra")
> str(Geissler)

'data.frame': 90 obs. of  4 variables:
 $ boys : int  0 0 0 0 0 0 0 0 0 0 ...
 $ girls: num  1 2 3 4 5 6 7 8 9 10 ...
 $ size : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Freq : int  108719 42860 17395 7004 2839 1096 436 161 66 30 ...
```

(a) Use `subset()` to create a data frame, `sax12` containing the *Geissler* observations in families with `size==12`. ★

```
> data("Saxony", package="vcd")
> data("Geissler", package="vcdExtra")
> sax12 <- subset(Geissler, size==12)
> sax12

   boys girls size Freq
12    0    12   12    3
24    1    11   12   24
35    2    10   12  104
45    3     9   12  286
54    4     8   12  670
62    5     7   12 1033
```

```
69    6     6   12 1343
75    7     5   12 1112
80    8     4   12  829
84    9     3   12  478
87   10     2   12  181
89   11     1   12   45
90   12     0   12    7
```

(b) Select the columns for `boys` and `Freq`.
★

```
> sax12 <- subset(sax12, select=c("boys","Freq"))
```

(c) Use `xtabs()` with a formula, `Freq ~ boys`, to create the one-way table.
★

```
> Saxony12<-xtabs(Freq~boys, data=sax12)
> Saxony12

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

(d) Do the same steps again to create a one-way table, `Saxony11`, containing similar frequencies for families of `size==11`.
★

```
> sax11 <- subset(Geissler, size==11, select = c("boys","Freq"))
> Saxony11 <- xtabs(Freq~boys, data=sax11)
> Saxony11

boys
   0    1    2    3    4    5    6    7    8    9   10   11
   8   72  275  837 1540 2161 2310 1801 1077  492   93   24
```

Exercise 2.7 * *Interactive coding of table factors*: Some statistical and graphical methods for contingency tables are implemented only for two-way tables, but can be extended to 3+-way tables by recoding the factors to interactive combinations along the rows and/or columns, in a way similar to what `ftable()` and `structable()` do for printed displays.

For the *`UCBAdmissions`* data, produce a two-way table object, `UCB.tab2`, that has the combinations of `Admit` and `Gender` as the rows, and `Dept` as its columns, to look like the result below:

```
                 Dept
Admit:Gender        A   B   C   D   E   F
  Admitted:Female  89  17 202 131  94  24
  Admitted:Male   512 353 120 138  53  22
  Rejected:Female  19   8 391 244 299 317
  Rejected:Male   313 207 205 279 138 351
```

(a) Try this the long way: convert *`UCBAdmissions`* to a data frame (`as.data.frame()`), manipulate the factors (e.g., `interaction()`), then convert back to a table (`as.data.frame()`).
★

```
> ucb.df$AG <- with(ucb.df, interaction(Admit, Gender, sep=":"))
> ucb <- subset(ucb.df, select = c("Dept", "AG", "Freq"))
> ucb.tab2 <- xtabs(Freq ~ AG + Dept, data=ucb)
> ucb.tab2

                 Dept
AG                  A   B   C   D   E   F
  Admitted:Male   512 353 120 138  53  22
  Rejected:Male   313 207 205 279 138 351
  Admitted:Female  89  17 202 131  94  24
  Rejected:Female  19   8 391 244 299 317
```

(b) Try this the short way: both `ftable()` and `structable()` have `as.matrix()` methods that convert their result to a matrix.
★

```
> ucb.tab2 <- as.matrix(structable(Dept ~ Admit + Gender, data = UCBAdmissions))
> ucb.tab2

                 Dept
Admit_Gender        A   B   C   D   E   F
  Admitted_Male   512 353 120 138  53  22
  Admitted_Female  89  17 202 131  94  24
  Rejected_Male   313 207 205 279 138 351
  Rejected_Female  19   8 391 244 299 317
```

Exercise 2.8 The data set *VisualAcuity* in vcd gives a $4 \times 4 \times 2$ table as a frequency data frame.

```
> data("VisualAcuity", package = "vcd")
> str(VisualAcuity)

'data.frame': 32 obs. of  4 variables:
 $ Freq  : num  1520 234 117 36 266 ...
 $ right : Factor w/ 4 levels "1","2","3","4": 1 2 3 4 1 2 3 4 1 2 ...
 $ left  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 2 2 2 2 3 3 ...
 $ gender: Factor w/ 2 levels "male","female": 2 2 2 2 2 2 2 2 2 2 ...
```

(a) From this, use `xtabs()` to create two 4×4 frequency tables, one for each gender. ★

```
> data("VisualAcuity", package="vcd")
> va.tabm <- xtabs(Freq ~ right+left, data = VisualAcuity, subset=gender=="male")
> va.tabm

     left
right   1   2   3   4
    1 821 112  85  35
    2 116 494 145  27
    3  72 151 583  87
    4  43  34 106 331

> va.tabf <- xtabs(Freq ~ right+left, data = VisualAcuity, subset=gender=="female")
> # or, subset after
> va.tab <- xtabs(Freq ~ ., data = VisualAcuity)
> va.tabm <- va.tab[,,"male"]
> va.tabf <- va.tab[,,"female"]
```

(b) Use `structable()` to create a nicely organized tabular display. ★

```
> structable(right ~ left + gender, data = va.tab)

            right    1    2    3    4
left gender
1    male          821  116   72   43
     female       1520  234  117   36
2    male          112  494  151   34
     female        266 1512  362   82
3    male           85  145  583  106
     female        124  432 1772  179
4    male           35   27   87  331
     female         66   78  205  492
```

(c) Use `xtable()` to create a LaTeX or HTML table. ★

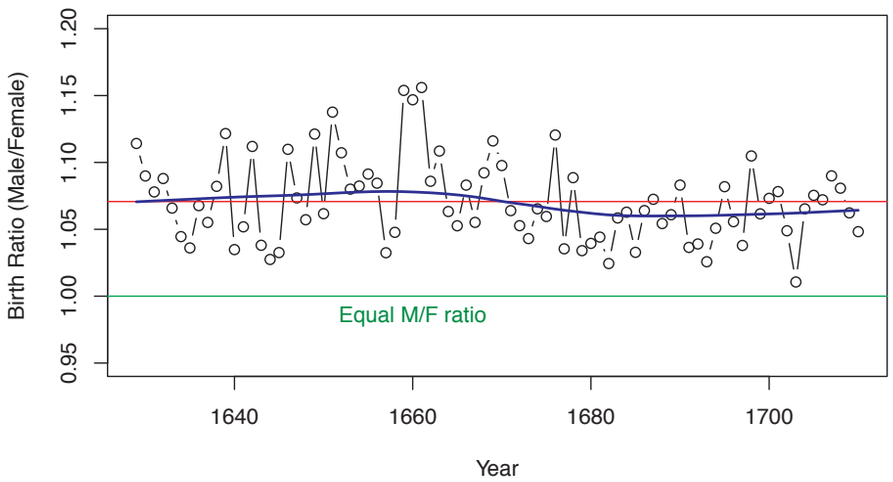
```
> library(xtable)
> va.xtab <- xtable(va.tabm)
> print(va.xtab, type="html")
```

Chapter 3 Fitting and Graphing Discrete Distributions

Exercise 3.1 The *Arbuthnot* data in HistData (Friendly, 2014a) (Example 3.1) also contains the variable `Ratio`, giving the ratio of male to female births.

(a) Make a plot of `Ratio` over `Year`, similar to Figure 3.1. What features stand out? Which plot do you prefer to display the tendency for more male births?
★

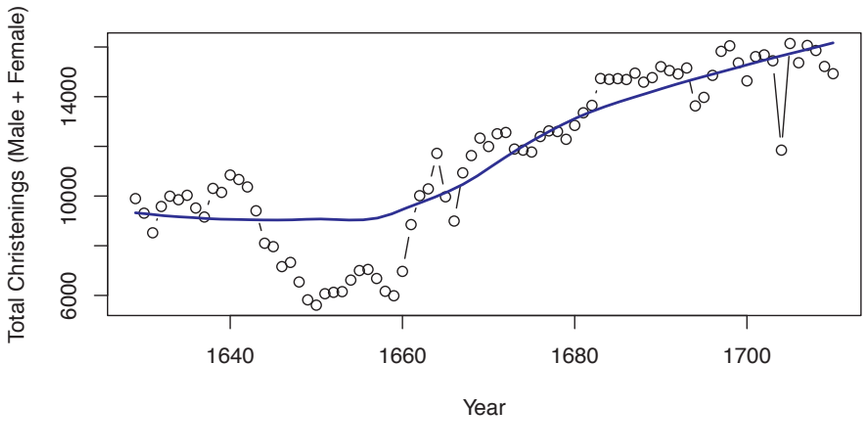
```
> library(HistData)
> data(Arbuthnot, package ="HistData")
>
>    # plot of Ratio by Year
> par(mar=c(5,4,1,1)+.1)
> with(Arbuthnot, {
+   plot(Year, Ratio, type='b', ylim=c(.95, 1.2),
+        ylab="Birth Ratio (Male/Female)")
+   abline(h=1, col="green", lwd=1)
+   abline(h=mean(Ratio), col="red")
+   text(x=1660, y=1, "Equal M/F ratio", pos=1, col="green3")
+   Arb.smooth <- loess.smooth(Year,Ratio)
+   lines(Arb.smooth$x, Arb.smooth$y, col="blue", lwd=2)
+ })
```



The plot is similar to Figure 3.1 in the text. If it is easier to think in terms of probability of a male birth, plotting that directly may be preferable.

(b) Plot the total number of christenings, `Males + Females` or `Total` (in 000s) over time. What unusual features do you see?
★

```
>    # total number of Christenings
> with(Arbuthnot, {
+   Total= Males + Females
+   plot(Year, Total, type='b', ylab="Total Christenings (Male + Female)")
+   Arb.smooth <- loess.smooth(Year,Total)
+   lines(Arb.smooth$x, Arb.smooth$y, col="blue", lwd=2)
+ })
```

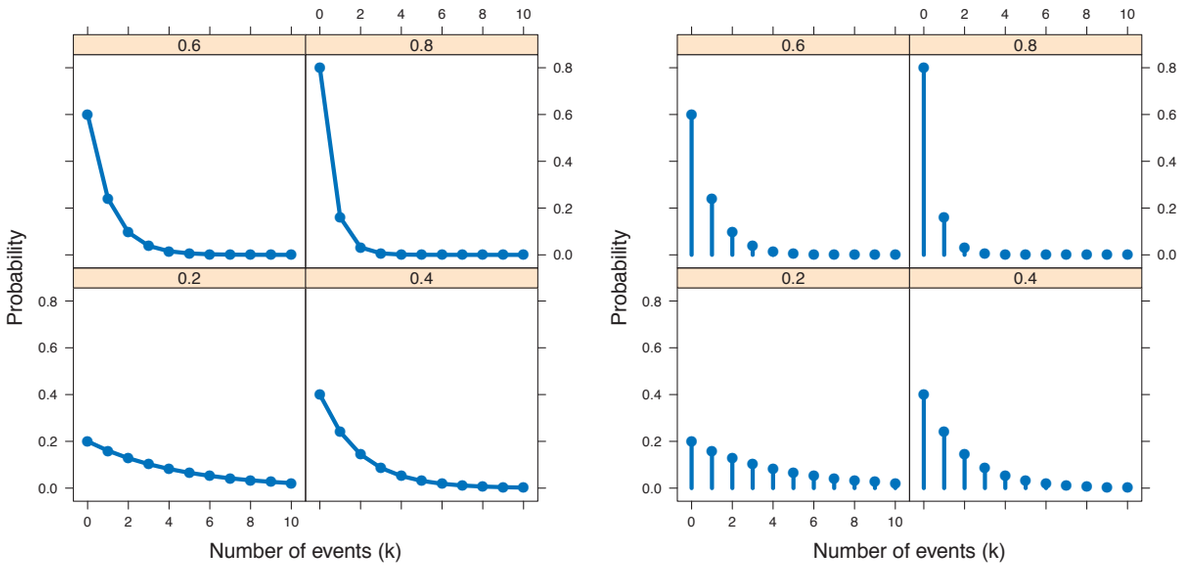


There was a large decline in births between 1640–1660, corresponding to years of plague in England.

Exercise 3.2 Use the graphical methods illustrated in Section 3.2 to plot a collection of geometric distributions for $p = 0.2, 0.4, 0.6, 0.8$, over a range of values of $k = 0, 1, \ldots 10$.

(a) With `xyplot()`, try the different plot formats using points connected with lines, as in Figure 3.9, or using points and lines down to the origin, as in the panels of Figure 3.10.
★

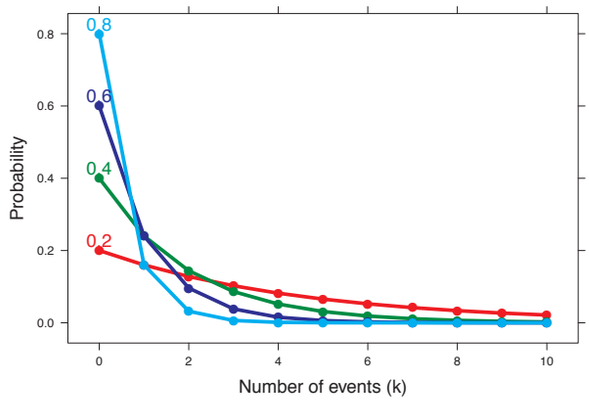
```
> KL <- expand.grid(k = 0 : 10, p = c(0.2, 0.4, 0.6, 0.8))
> geom_df <- data.frame(KL, prob = dgeom(KL$k, KL$p))
> geom_df$p = factor(geom_df$p)
>
> library(lattice)
> mycol<-palette()[2:5]
> xyplot(prob ~ k | p , data = geom_df, type = c("b"),
+        pch = 16, lwd = 4, cex = 1.25,
+        xlab = list("Number of events (k)", cex = 1.25), layout = c(2,2),
+        ylab = list("Probability", cex = 1.25))
> xyplot(prob ~ k | p , data = geom_df, type = c("h", "p"),
+        pch = 16, lwd = 4, cex = 1.25,
+        xlab = list("Number of events (k)", cex = 1.25), layout = c(2,2),
+        ylab = list("Probability", cex = 1.25))
```



(b) Also with `xyplot()`, produce one version of a multi-line plot in a single panel that you think shows well how these distributions change with the probability p of success.

★

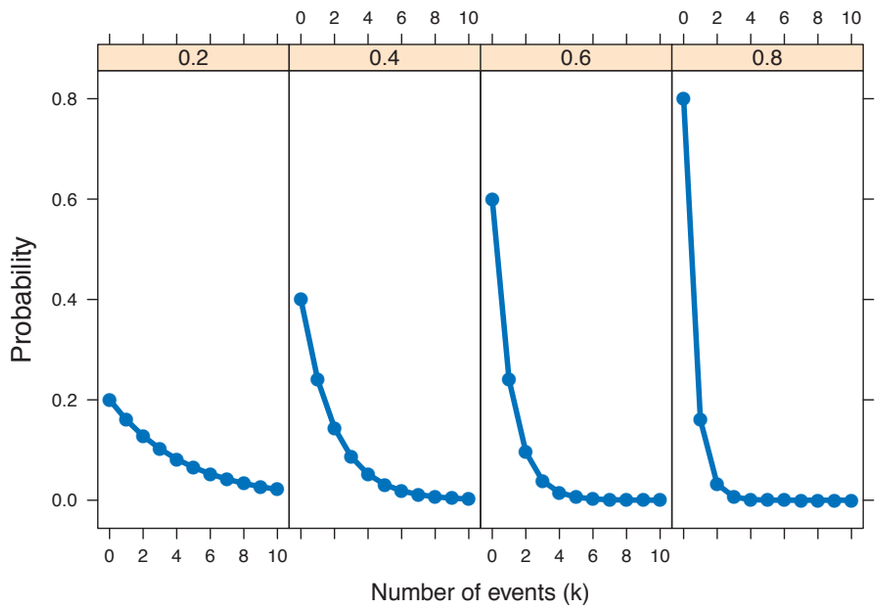
```
> geomplt<-xyplot(prob ~ k , data = geom_df, groups = p,
+                 type = c("b"), pch = 16, lwd = 4, cex = 1.25, col = mycol,
+                 xlab = list("Number of events (k)", cex = 1.25),
+                 ylab = list("Probability", cex = 1.25))
> library(directlabels)
> direct.label(geomplt, list("top.points", cex = 1.25, dl.trans(y = y + 0.1)))
```



(c) Do the same in a multi-panel version, conditional on p.

★

```
> xyplot(prob ~ k | p , data = geom_df, type = c("b"),
+        pch = 16, lwd = 4, cex = 1.25,
+        xlab = list("Number of events (k)", cex = 1.1), layout = c(4,1),
+        ylab = list("Probability", cex = 1.25))
```

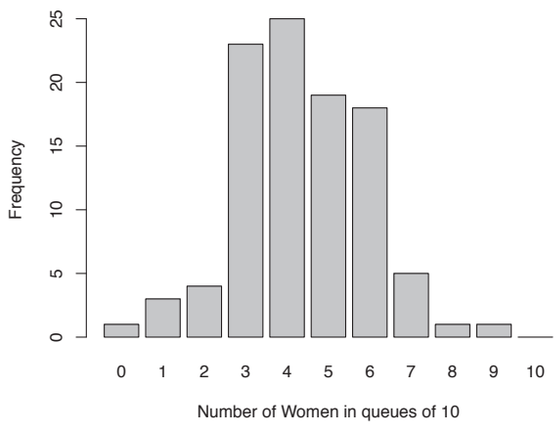


Exercise 3.3 Use the data set *WomenQueue* to:

(a) Produce plots analogous to those shown in Section 3.1 (some sort of bar graph of frequencies).

★

```
> data("WomenQueue", package = "vcd")
> barplot(WomenQueue, xlab="Number of Women in queues of 10", ylab= "Frequency")
```

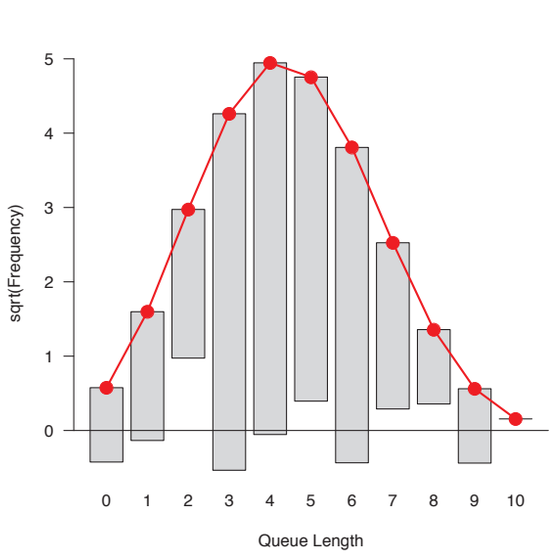


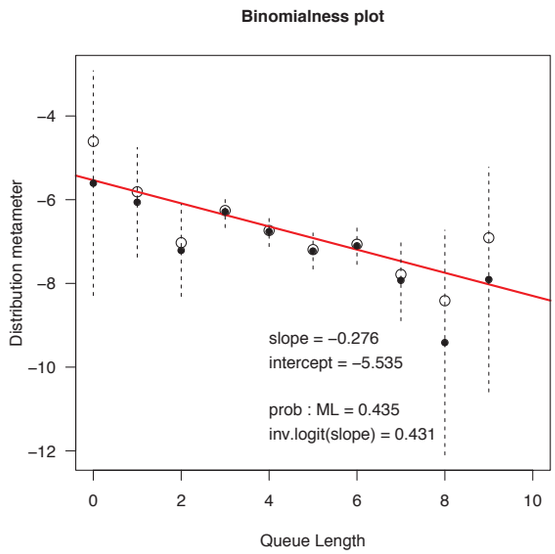
(b) Check for goodness-of-fit to the binomial distribution using the `goodfit()` methods described in Section 3.3.2.
★ Note that with `goodfit()`, you should specify $n = 10$ for the binomial distribution as the `size` parameter.

```
> library(vcd)
> gf.women <- goodfit(WomenQueue, type = "binomial", par=list(size=10))
> summary(gf.women)

 Goodness-of-fit test for binomial distribution

                     X^2 df P(> X^2)
Likelihood Ratio 8.651  8  0.37259
```

(c) Make a reasonable plot showing departure from the binomial distribution.
★ The simplest plot is the hanging rootogram. An alternative plot is a "binomialness" plot produced by `distplot()`.

```
> plot(gf.women, xlab = "Queue Length")
> distplot(WomenQueue, type = "binomial", size=10, xlab = "Queue Length")
```





(d) Suggest some reasons why the number of women in queues of length 10 might depart from a binomial distribution, $\text{Bin}(n = 10, p = 1/2)$.

★

- Perhaps women (or men) are more prevalent in these queues, so $p \neq 1/2$.
- People often join lines in groups, so the observations are unlikely to be independent.

Exercise 3.4 Continue Example 3.13 on the distribution of male children in families in Saxony by fitting a binomial distribution, $\text{Bin}(n = 12, p = \frac{1}{2})$, specifying equal probability for boys and girls. [*Hint*: you need to specify both `size` and `prob` values for `goodfit()`.]

(a) Carry out the GOF test for this fixed binomial distribution. What is the ratio of χ^2/df? What do you conclude?
★ Note that you need to specify both n and p as fixed parameters here.

```
> Saxony_gf <-goodfit(Saxony, type = "binomial", par=list(size=12, prob=.5))
> ss <-summary(Saxony_gf)

 Goodness-of-fit test for binomial distribution

                    X^2 df   P(> X^2)
Pearson          249.20 12 2.0133e-46
Likelihood Ratio 205.41 12 2.4936e-37

>   #The ratio of Chi-square/df
> ss[,"X^2"] / ss[,"df"]

         Pearson Likelihood Ratio
          20.766           17.117
```

The binomial model fits very badly.

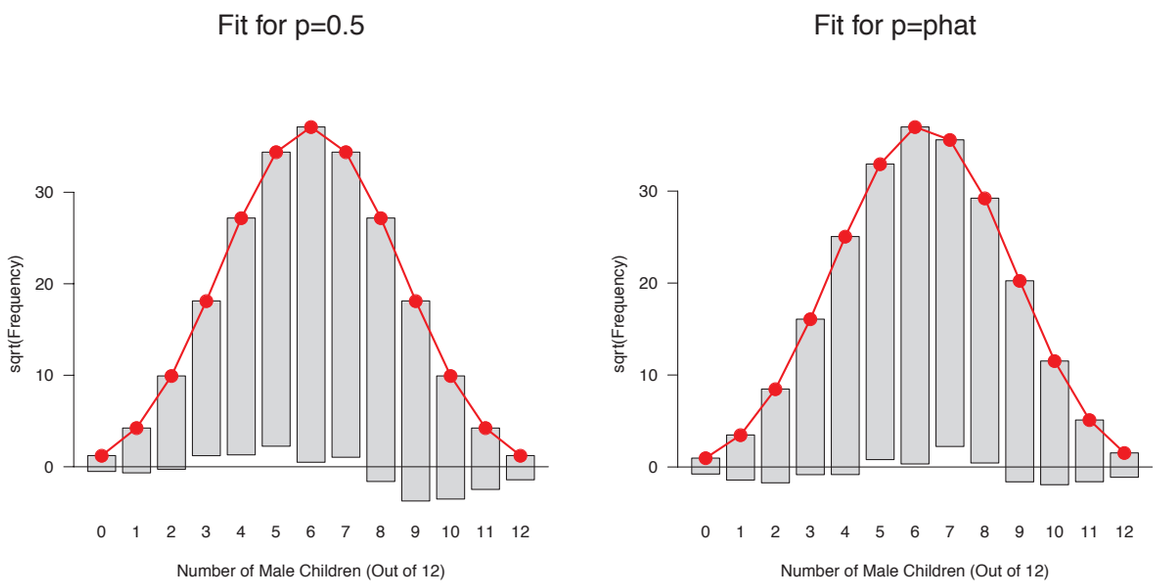
(b) Test the additional lack of fit for the model $\text{Bin}(n = 12, p = \frac{1}{2})$ compared to the model $\text{Bin}(n = 12, p = \hat{p})$ where $\hat{p}$ is estimated from the data.
★

```
> Saxony_gf2 <- goodfit(Saxony, type = "binomial", par=list(size=12))
> summary(Saxony_gf2)

 Goodness-of-fit test for binomial distribution

                    X^2 df   P(> X^2)
Likelihood Ratio 97.007 11 6.9782e-16
```

This fits much better, but still not a good fit.

(c) Use the `plot.gootfit()` method to visualize these two models.
★

```
> plot(Saxony_gf, main = "Fit for p=0.5", xlab = "Number of Male Children (Out of 12)")
> plot(Saxony_gf2, main = "Fit for p=phat", xlab = "Number of Male Children (Out of 12)")
```



Exercise 3.5 For the *Federalist* data, the examples in Section 3.3.1 and Section 3.3.2 showed the negative binomial to provide an acceptable fit. Compare this with the simpler special case of geometric distribution, corresponding to $n = 1$.

(a) Use `goodfit()` to fit the geometric distribution. [Hint: use `type="nbinomial"`, but specify `size=1` as a parameter.]
★

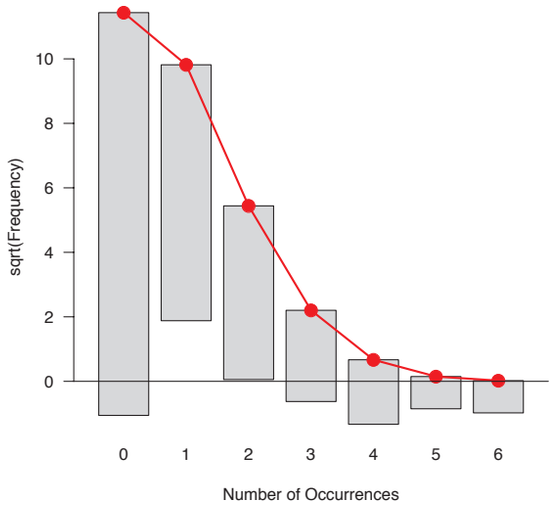
```
> fdfit1 <- goodfit(Federalist, type = "binomial", par = list(size=6))
> fdfit1

Observed and fitted values for binomial distribution
with parameters estimated by `ML'

 count observed     fitted pearson residual
     0      156 1.3072e+02          2.21074
     1       63 9.6362e+01         -3.39860
     2       29 2.9597e+01         -0.10972
     3        8 4.8483e+00          1.43139
     4        4 4.4673e-01          5.31624
     5        1 2.1954e-02          6.60094
     6        1 4.4953e-04         47.14399

> fdfit2 <- goodfit(Federalist, type = "nbinomial", par = list(size=1))
> fdfit2

Observed and fitted values for nbinomial distribution
with parameters estimated by `ML with size fixed'

 count observed    fitted pearson residual
     0      156 158.16590        -0.172219
     1       63  62.68326         0.040006
     2       29  24.84221         0.834194
     3        8   9.84530        -0.588102
     4        4   3.90182         0.049702
     5        1   1.54635        -0.439353
     6        1   0.61284        -0.015044
```

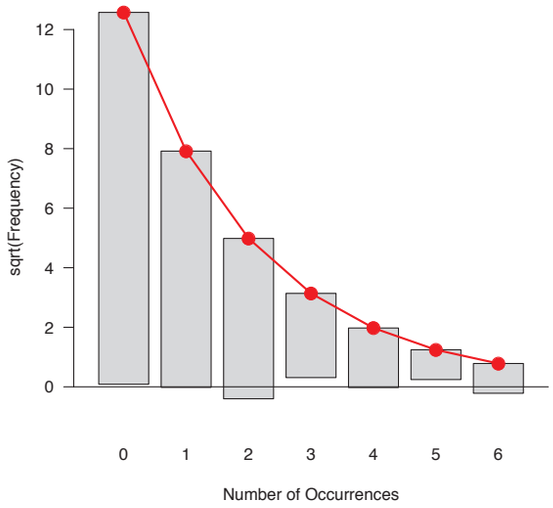
(b) Compare the negative binomial and the geometric models statistically, by a likelihood-ratio test of the difference between these two models.
★

```
> summary(fdfit1)

 Goodness-of-fit test for binomial distribution

                     X^2 df   P(> X^2)
Likelihood Ratio 49.026  5 2.1927e-09

> summary(fdfit2)

 Goodness-of-fit test for nbinomial distribution

                     X^2 df P(> X^2)
Likelihood Ratio 2.2941  5  0.80713
```

(c) Compare the negative binomial and the geometric models visually by hanging rootograms or other methods.
★ Hanging rootograms:

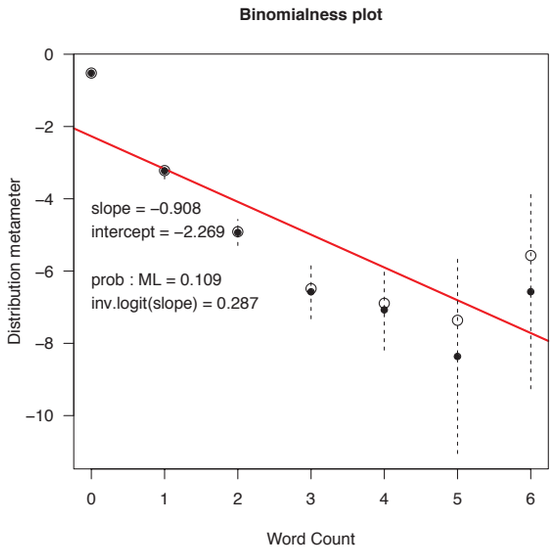
```
> plot(fdfit1)
> plot(fdfit2)
```

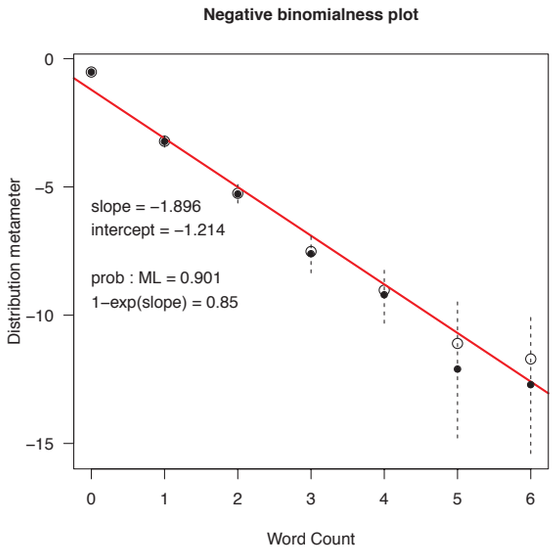




Distribution-ness plots:

```
> distplot(Federalist, type = "binomial", size=6, xlab = "Word Count")
> distplot(Federalist, type = "nbinomial", size=6, xlab = "Word Count")
```





Exercise 3.6 Mosteller and Wallace (1963, Table 2.4) give the frequencies, n_k, of counts $k = 0, 1, \ldots$ of other selected marker words in 247 blocks of text known to have been written by Alexander Hamilton. The data below show the occurrences of the word *upon*, that Hamilton used much more than did James Madison.

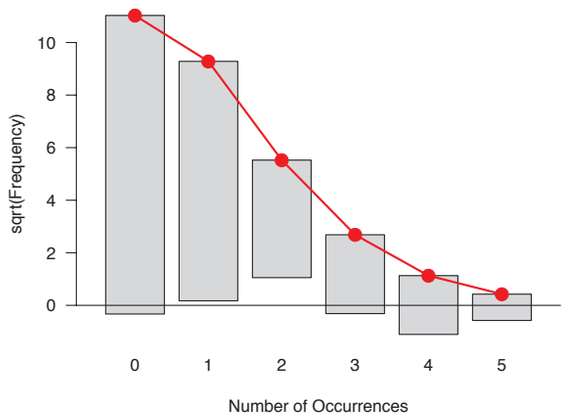
```
> count <- 0 : 5
> Freq <- c(129, 83, 20, 9, 5, 1)
```

(a) Read these data into R and construct a one-way table of frequencies of counts or a matrix or data frame with frequencies in the first column and the corresponding counts in the second column, suitable for use with `goodfit()`.
★ `goodfit()` requires its first argument to be either a one-way table (from `xtabs()`), or a data.frame with frequencies in the *first* column and the corresponding counts in the second column. Both of the following forms will work.

```
> count <- 0:5
> Freq <- c(129, 83, 20, 9, 5, 1)
> sum(Freq)  # check N

[1] 247

> (Upon <- data.frame(Freq, count))              # as a data.frame

  Freq count
1  129     0
2   83     1
3   20     2
4    9     3
5    5     4
6    1     5

> (Upon.tab <- xtabs(Freq ~ count, data=Upon))  # one-way table

count
  0   1   2   3   4   5
129  83  20   9   5   1
```

(b) Fit and plot the Poisson model for these frequencies.
★

```
> (up0 <- goodfit(Upon, type="poisson"))

Observed and fitted values for poisson distribution
with parameters estimated by `ML'

 count observed    fitted pearson residual
     0      129 121.61816          0.66937
     1       83  86.16671         -0.34115
     2       20  30.52465         -1.90494
     3        9   7.20892          0.66708
     4        5   1.27688          3.29481
     5        1   0.18094          1.75800

> summary(up0)

 Goodness-of-fit test for poisson distribution

                    X^2 df P(> X^2)
Likelihood Ratio 13.139  4 0.010617

> plot(up0)
```



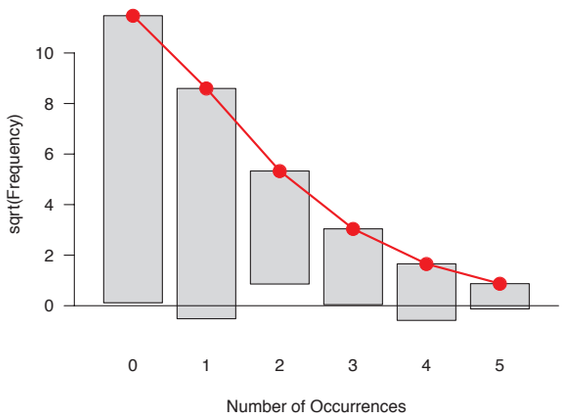
(c) Fit and plot the negative binomial model for these frequencies.
★

```
> (up1 <- goodfit(Upon, type="nbinomial"))

Observed and fitted values for nbinomial distribution
with parameters estimated by `ML'

 count observed    fitted pearson residual
     0      129 131.65936        -0.231767
```

```
     1       83  73.89421         1.059285
     2       20  28.41547        -1.578705
     3        9   9.25319        -0.083233
     4        5   2.74068         1.364738
     5        1   0.76332        -0.036432
```

```
> summary(up1)

 Goodness-of-fit test for nbinomial distribution

                    X^2 df P(> X^2)
Likelihood Ratio 6.0306  3  0.11013
```

```
> plot(up1)
```



(d) What do you conclude?

★ The negative binomial model fits better than the Poisson.

Exercise 3.7 The data frame `Geissler` in the vcdExtra package contains the complete data from Geissler's (1889) tabulation of family sex composition in Saxony. The table below gives the number of boys in families of size 11.

boys	0	1	2	3	4	5	6	7	8	9	10	11
Freq	8	72	275	837	1,540	2,161	2,310	1,801	1,077	492	93	24

(a) Read these data into R.

★ See Exercise 2.6, which calculates `sax11` in the form of a data frame.

(b) Following Example 3.13, use `goodfit()` to fit the binomial model and plot the results. Is there an indication that the binomial does not fit these data?

★ The binomial distribution fits badly, where the extremes are under-fitted, and the middle values are over-fitted.

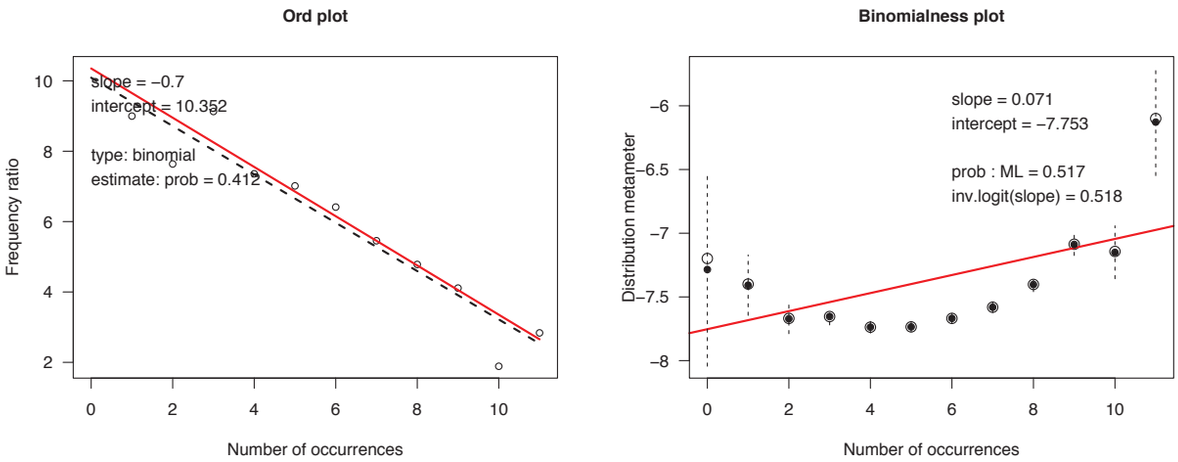
```
> sax11.tab <- xtabs(Freq ~ boys, data=sax11)
> goodfit(sax11.tab, type="binomial", par=list(size=11))

Observed and fitted values for binomial distribution
with parameters estimated by `ML'

 count observed    fitted pearson residual
     0        8    3.5616           2.3518
     1       72   41.9479           4.6400
     2      275  224.5724           3.3650
     3      837  721.3629           4.3055
     4     1540 1544.7559          -0.1210
     5     2161 2315.6023          -3.2128
     6     2310 2479.3627          -3.4013
     7     1801 1896.2173          -2.1866
     8     1077 1015.1593           1.9409
     9      492  362.3173           6.8130
    10       93   77.5881           1.7497
    11       24    7.5523           5.9850
```

```
> summary(goodfit(sax11.tab, type="binomial", par=list(size=11)))
```

```
 Goodness-of-fit test for binomial distribution

                     X^2 df   P(> X^2)
Likelihood Ratio 148.09 10 9.2126e-27
```

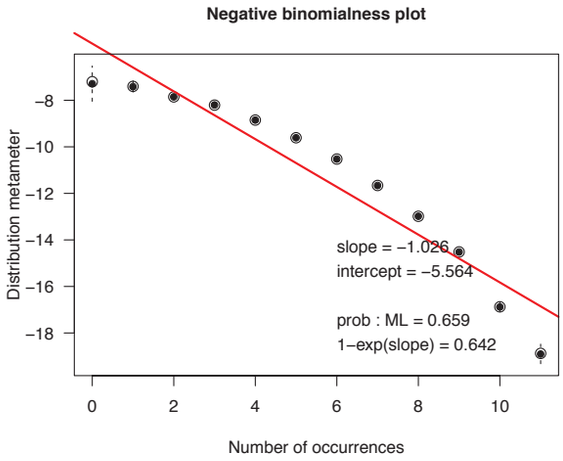
(c) Diagnose the form of the distribution using the methods described in Section 3.4.
★ The Ord plot indicates that the closest distribution according to its heuristics is the binomial; the binomialness distribution plot, however, shows this is not an acceptable model, as was also seen in the text for families of size 12 (Figure 3.22).

```
> Ord_plot(sax11.tab)
> distplot(sax11.tab, type="binomial", size=11)
```



(d) Try fitting the negative binomial distribution, and use `distplot()` to diagnose whether the negative binomial is a reasonable fit.
★

```
> distplot(sax11.tab, type = "nbinomial", size = 11)
> goodfit(sax11.tab, type = "nbinomial", par = list(size = 11))

Observed and fitted values for nbinomial distribution
with parameters estimated by `ML with size fixed'

 count observed  fitted pearson residual
     0        8  109.12          -9.6801
     1       72  409.11         -16.6667
     2      275  836.63         -19.4171
     3      837 1235.67         -11.3414
     4     1540 1474.07           1.7171
     5     2161 1507.26          16.8389
     6     2310 1369.95          25.3981
     7     1801 1133.97          19.8082
     8     1077  869.62           7.0323
     9      492  625.73          -5.3462
    10       93  426.55         -16.1500
    11       24  277.55         -25.3999
```



The negative binomial is not a good choice either, as can be seen by the size of the residuals. In fact, the ***double binomial distribution*** will fit well, as illustrated in Example 3.23.

Exercise 3.8 The data frame *Bundesliga* gives a similar data set to that for UK soccer scores (*UKSoccer*) examined in Example 3.9, but over a wide range of years. The following lines calculate a two-way table, BL1995, of home-team and away-team goals for the 306 games in the year 1995.

```
> data("Bundesliga", package = "vcd")
> BL1995 <- xtabs(~ HomeGoals + AwayGoals, data = Bundesliga,
+                 subset = (Year == 1995))
> BL1995

         AwayGoals
HomeGoals  0  1  2  3  4  5  6
        0 26 16 13  5  0  1  0
        1 19 58 20  5  4  0  1
        2 27 23 20  5  1  1  1
        3 14 11 10  4  2  0  0
        4  3  5  3  0  0  0  0
        5  4  1  0  1  0  0  0
        6  1  0  0  1  0  0  0
```

(a) As in , find the one-way distributions of HomeGoals, AwayGoals, and TotalGoals = HomeGoals + AwayGoals.
★ There are several ways to do this, but as illustrated in the text for Example 3.9, create the one-way variables in a data.frame, and then use xtabs() to get their marginal distributions.

```
> BL.df <- as.data.frame(BL1995, stringsASFactors=FALSE)
> BL.df <- within(BL.df, {
+   HomeGoals <- as.numeric(HomeGoals)
+   AwayGoals <- as.numeric(AwayGoals)
+   TotalGoals <- HomeGoals + AwayGoals
+   })
>    # marginal distributions
> (BL.home <- xtabs(Freq ~ HomeGoals, data=BL.df))

HomeGoals
  1   2   3   4   5   6   7
 61 107  78  41  11   6   2

> (BL.away <- xtabs(Freq ~ AwayGoals, data=BL.df))

AwayGoals
  1   2   3   4   5   6   7
 94 114  66  21   7   2   2

> (BL.total <- xtabs(Freq ~ TotalGoals, data=BL.df))

TotalGoals
 2  3  4  5  6  7  8  9 10 11 12 13 14
26 35 98 62 39 29 10  4  2  1  0  0  0
```

(b) Use `goodfit()` to fit and plot the Poisson distribution to each of these. Does the Poisson seem to provide a reasonable fit?

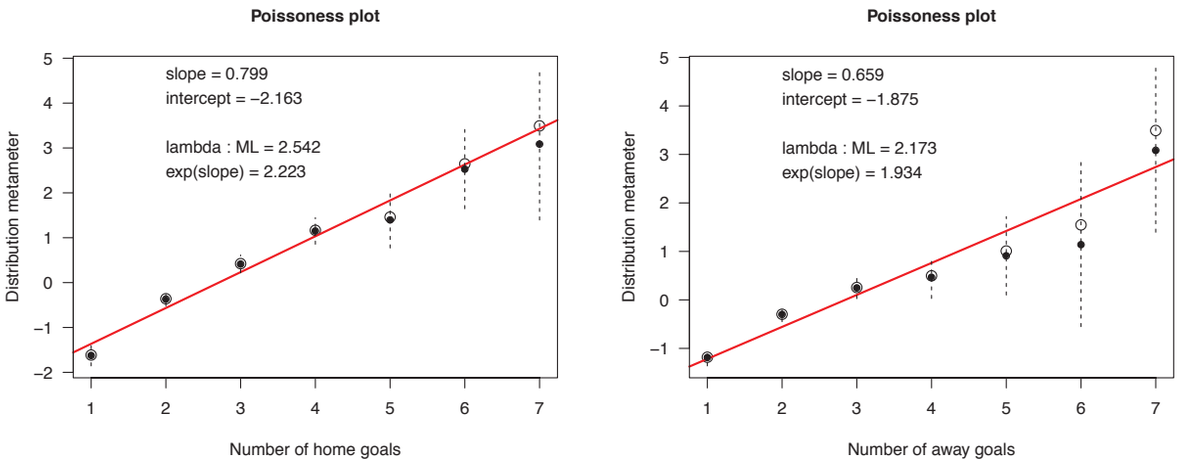
★ The Poisson distribution has a bad fit for all of these.

```
> summary(goodfit(BL.home))

 Goodness-of-fit test for poisson distribution

                     X^2 df   P(> X^2)
Likelihood Ratio 70.722  5 7.2516e-14

> summary(goodfit(BL.away))

 Goodness-of-fit test for poisson distribution

                     X^2 df   P(> X^2)
Likelihood Ratio 97.973  5 1.4131e-19

> summary(goodfit(BL.total))

 Goodness-of-fit test for poisson distribution

                     X^2 df   P(> X^2)
Likelihood Ratio 72.558  8 1.5185e-12
```

(c) Use `distplot()` to assess fit of the Poisson distribution.

★ The distribution plots for home goals looks better than that for away goals, which shows a systematic departure from the red line.

```
> distplot(BL.home, xlab="Number of home goals")
> distplot(BL.away, xlab="Number of away goals")
```



(d) What circumstances of scoring goals in soccer might cause these distributions to deviate from Poisson distributions?

★ The Poisson distribution relies on (a) independent events with (b) constant probabilities. The probability of scoring a goal is almost certainly not constant over all pairs of teams.

Exercise 3.9 * Repeat the exercise above, this time using the data for all years in which there was the standard number (306) of games, that is for `Year>1965`, tabulated as shown below.

```
> BL <- xtabs(~ HomeGoals + AwayGoals, data = Bundesliga,
+             subset = (Year > 1965))
> BL

         AwayGoals
HomeGoals    0    1    2    3    4    5    6    7    8    9
       0   868  590  458  206   88   22   12    2    0    0
       1  1049 1550  589  360  121   34    8    6    1    1
       2  1039 1144  810  228   95   26   10    2    1    0
       3   712  793  392  187   43    8    5    2    0    0
       4   346  388  245   73   26    2    3    0    0    0
       5   128  164  106   34    2    2    1    0    0    0
```

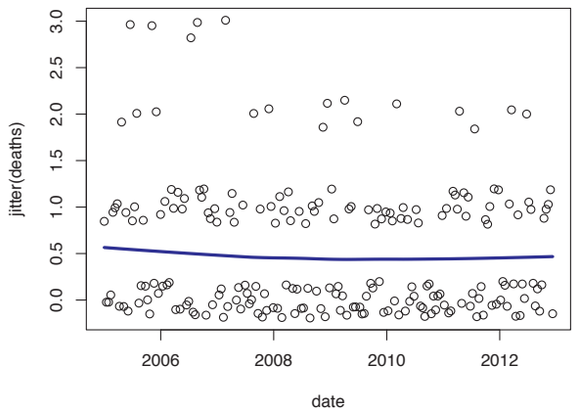
```
       6    61    63    38    10     0     2     0     0     0     0
       7    20    16    12     4     3     0     0     0     0     0
       8     2     4     3     0     1     0     0     0     0     0
       9     2     2     1     0     0     0     0     0     0     0
       10    2     0     0     0     0     0     0     0     0     0
       11    1     2     0     0     0     0     0     0     0     0
       12    1     0     0     0     0     0     0     0     0     0
```

★ The steps are similar to those in the previous problem, but now there are more years, and the range of both home goals and away goals is larger.

Exercise 3.10 Using the data *CyclingDeaths* introduced in Example 3.6 and the one-way frequency table `CyclingDeaths.tab = table(CyclingDeaths$deaths)`,

(a) Make a sensible plot of the number of deaths over time. For extra credit, add a smoothed curve (e.g., using `lines(lowess(...))`).
★ The number of deaths in a given fortnight are discrete, taking values `0:3`. A barplot, using `plot(deaths ~ date, type="h", data=CyclingDeaths)` is one option. Perhaps slightly better is to jitter the number of deaths.

```
> data("CyclingDeaths", package="vcdExtra")
> CyclingDeaths.tab <- table(CyclingDeaths$deaths)
> plot(jitter(deaths) ~ date, data=CyclingDeaths)
> with(CyclingDeaths, {lines(lowess(date, deaths), lwd=3, col="blue")})
```

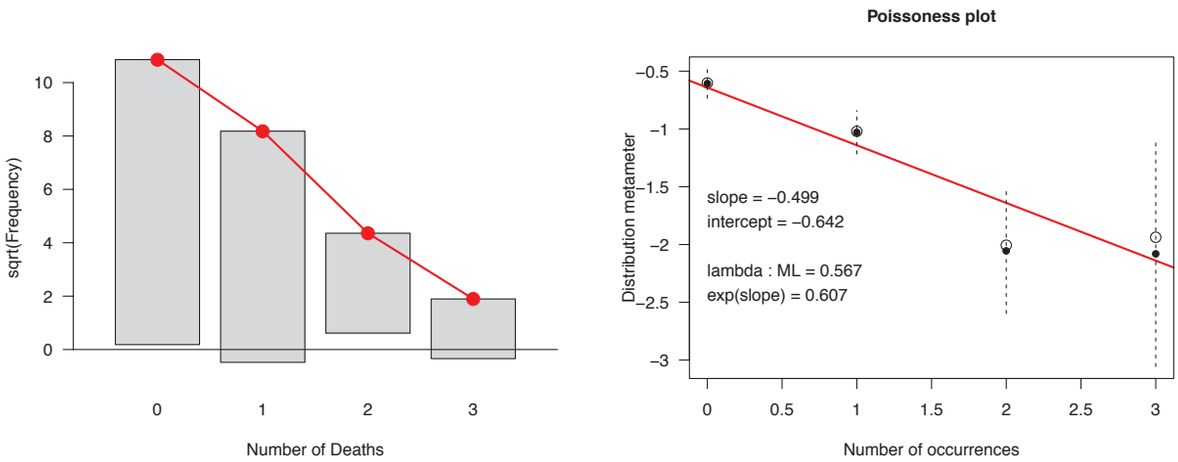


(b) Test the goodness of fit of the table `CyclingDeaths.tab` to a Poisson distribution statistically using `goodfit()`.
★

```
> gf <- goodfit(CyclingDeaths.tab)
> gf

Observed and fitted values for poisson distribution
with parameters estimated by `ML'

 count observed   fitted pearson residual
     0      114 117.9464         -0.36338
     1       75  66.9119          0.98877
     2       14  18.9798         -1.14306
     3        5   3.5891          0.41084

> summary(gf)

 Goodness-of-fit test for poisson distribution

                    X^2 df P(> X^2)
Likelihood Ratio 4.1517  2  0.12545
```

(c) Continue this analysis using a `rootogram()` and `distplot()`.
★

```
> plot(gf, xlab="Number of Deaths")
> distplot(CyclingDeaths.tab)
```



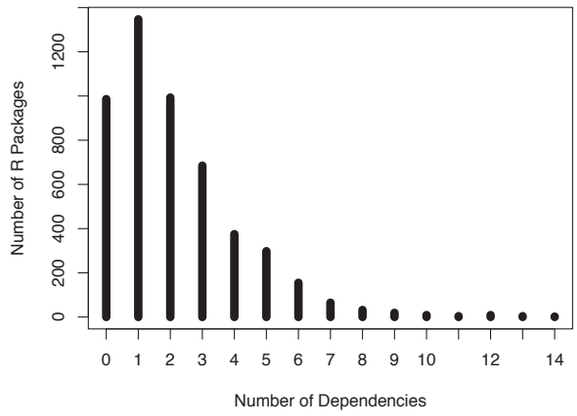
(d) Write a one-paragraph summary of the results of these analyses and your conclusions.
★

Exercise 3.11 * The one-way table, *Depends*, in vcdExtra and shown below gives the frequency distribution of the number of dependencies declared in $4,983$ R packages maintained on the CRAN distribution network on January 17, 2014. That is, there were 986 packages that had no dependencies, $1,347$ packages that depended on one other package, . . . up to 2 packages that depended on 14 other packages.

Depends	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
# Pkgs	986	1,347	993	685	375	298	155	65	32	19	9	4	9	4	2

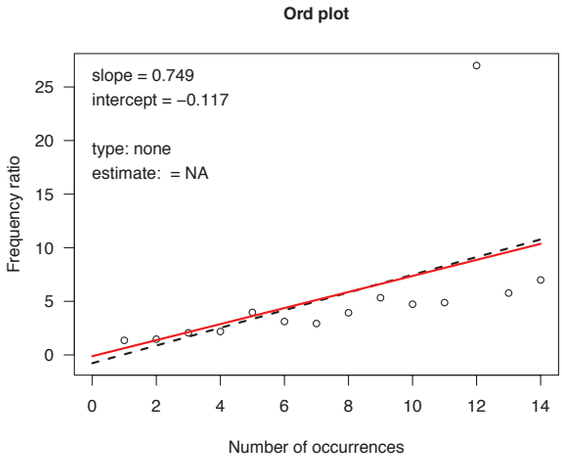
(a) Make a bar plot of this distribution.
★

```
> data("Depends", package="vcdExtra")
> plot(Depends, xlab="Number of Dependencies", ylab="Number of R Packages", lwd=8)
```



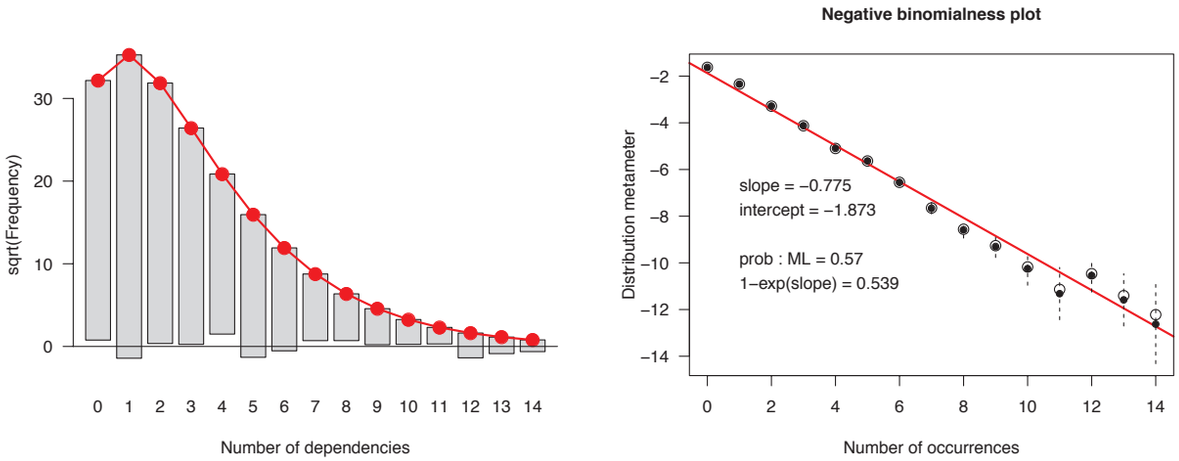
(b) Use `Ord_plot()` to see if this method can diagnose the form of the distribution.
★ This turns out to be a case where the Ord plot method, as implemented in `Ord_plot()` does not determine the form of the distribution. According to Table 3.11, the log series distribution is the only one with positive slope b and negative intercept a, but this requires $a = -b$.

```
> Ord_plot(Depends)
```



(c) Try to fit a reasonable distribution to describe dependencies among R packages.
★ Among the distributions described in Chapter 3 and implemented in the vcd package the negative binomial may be the best to try.

```
> dep.gf <- goodfit(Depends, type="nbinomial")
> summary(dep.gf)

 Goodness-of-fit test for nbinomial distribution

                     X^2 df   P(> X^2)
Likelihood Ratio 49.081 12 2.0243e-06

> plot(dep.gf, xlab="Number of dependencies")
> distplot(Depends, type="nbinomial")
```



So, this does not fit well, particularly for the packages with many dependencies, but it is not altogether terrible. The remaining differences come from further heterogeneity for which we haven't got any covariates in the data set. For example, the age of the package would seem like a natural candidate: older packages probably have fewer dependencies.

Exercise 3.12 * How many years does it take to get into the baseball Hall of Fame? The Lahman (Friendly, 2014b) package provides a complete record of historical baseball statistics from 1871 to the present. One table, *HallOfFame*, records the history of players nominated to the Baseball Hall of Fame, and those eventually inducted. The table below, calculated in `help(HallOfFame, package="Lahman")`, records the distribution of the number of years taken

(from first nomination) for the 109 players in the Hall of Fame to be inducted (1936–present). Note that `years==0` does not, and cannot, occur in this table, so the distribution is restricted to positive counts. Such distributions are called ***zero-truncated distribution***s. Such distributions are like the ordinary ones, but with the probability of zero being zero. Thus the other probabilities are scaled up (i.e., divided by $1 - \Pr(Y = 0)$) so they sum to 1.

years	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
inducted	46	10	8	7	8	4	2	4	6	3	3	1	4	1	2

(a) For the Poisson distribution, show that the zero-truncated probability function can be expressed in the form

$$\Pr\{X = k \,|\, k > 0\} = \frac{1}{1 - e^{-\lambda}} \times \frac{e^{-\lambda}\,\lambda^k}{k!} \qquad k = 1, 2, \ldots$$

★ The standard Poisson distribution has the probability mass function

$$\Pr\{X = k\} = \frac{e^{-\lambda}\,\lambda^k}{k!} \quad k = 1, 2, \ldots .$$

For this, $\Pr\{X = 0\} = e^{-\lambda}$. The formula given in the problem scales the standard probability to account for zero-truncation, i.e., by the factor $\frac{1}{1-\Pr\{X=0\}}$.

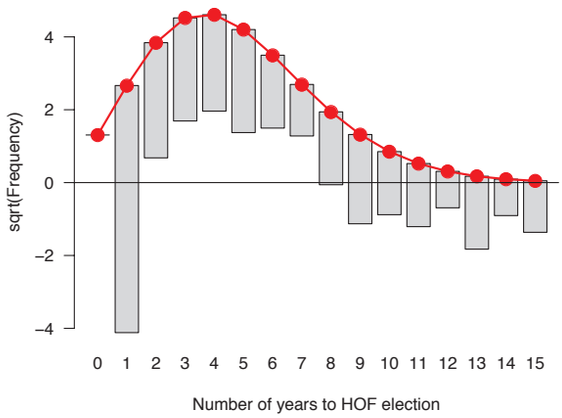
(b) Show that the mean is $\lambda/(1 - \exp(-\lambda))$.

★

(c) Enter these data into R as a one-way table, and use `goodfit()` to fit the standard Poisson distribution, as if you hadn't encountered the problem of zero truncation.

★ The rootogram below shows why zero truncation needs to be taken into account.

```
> years <- 1:15
> inducted <- c(46, 10,  8,  7,  8,  4,  2,  4,  6,  3,  3,  1,  4,  1,  2)
> HOF.df <- data.frame(years, inducted)
> HOF.tab <- xtabs(inducted ~ years, data=HOF.df)
> goodfit(HOF.tab)

Observed and fitted values for poisson distribution
with parameters estimated by `ML'

 count observed     fitted pearson residual
     0        0  1.7081050         -1.30694
     1       46  7.0988218         14.60056
     2       10 14.7512214         -1.23706
     3        8 20.4351783         -2.75082
     4        7 21.2319627         -3.08866
     5        8 17.6478516         -2.29660
     6        4 12.2239706         -2.35221
     7        2  7.2574819         -1.95157
     8        4  3.7702285          0.11833
     9        6  1.7409924          3.22782
    10        3  0.7235500          2.67623
    11        3  0.2733679          5.21498
    12        1  0.0946756          2.94229
    13        4  0.0302668         22.81803
    14        1  0.0089848         10.45503
    15        2  0.0024894         34.54549

> summary(HOF.tab)

Number of cases in table: 109
Number of factors: 1

> plot(goodfit(HOF.tab), xlab='Number of years to HOF election')
```

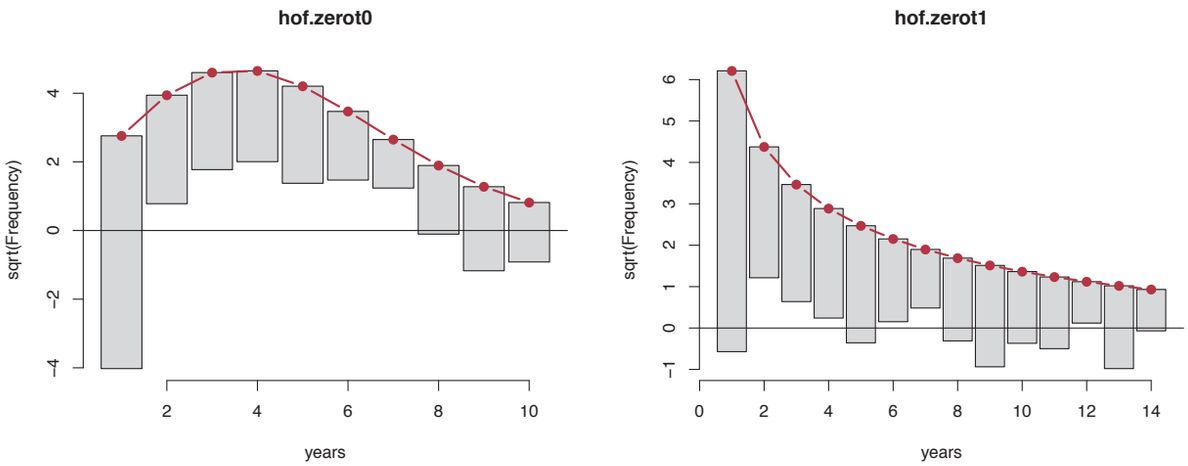


Though not asked in the problem, the zero-truncated Poisson distribution can be fit using `vglm()` in the VGAM (Yee, 2015) package.

```
> library(VGAM)
> hof.tpois0 <- vglm(years ~ 1, family=pospoisson, data=HOF.df, weights=inducted)
> hof.tpois0

Call:
vglm(formula = years ~ 1, family = pospoisson, data = HOF.df,
    weights = inducted)

Coefficients:
(Intercept)
     1.4076

Degrees of Freedom: 15 Total; 14 Residual
Log-likelihood: -339.31
```

Models for count data, taking account of excess zeros or zero truncation are examined in more detail in Chapter 11. There, we use the countreg (Zeileis and Kleiber, 2014) package, that provides a `zerotrunc()` function for these problems. The zero truncated negative binomial provides a better fit than the zero truncated Poisson, but that is not great either.

```
> library(countreg)
> hof.zerot0 <- zerotrunc(years ~ 1, weights = inducted, data=HOF.df)
> hof.zerot1 <- zerotrunc(years ~ 1, weights = inducted, dist="negbin")
> countreg::rootogram(hof.zerot0)
> countreg::rootogram(hof.zerot1)
```



Chapter 4 Two-Way Contingency Tables

Exercise 4.1 The data set `fat`, created below, gives a 2×2 table recording the level of cholesterol in diet and the presence of symptoms of heart disease for a sample of 23 people.

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

(a) Use `chisq.test(fat)` to test for association between diet and disease. Is there any indication that this test may not be appropriate here?
★

(b) Use a fourfold display to test this association visually. Experiment with the different options for standardizing the margins, using the `margin` argument to `fourfold()`. What evidence is shown in different displays regarding whether the odds ratio differs significantly from 1?
★

(c) `oddsratio(fat, log = FALSE)` will give you a numerical answer. How does this compare to your visual impression from fourfold displays?
★

(d) With such a small sample, Fisher's exact test may be more reliable for statistical inference. Use `fisher.test(fat)`, and compare these results to what you have observed before.
★

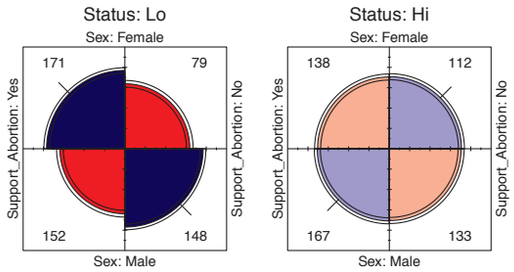
(e) Write a one-paragraph summary of your findings and conclusions for this data set.
★

Exercise 4.2 The data set *`Abortion`* in vcdExtra gives a $2 \times 2 \times 2$ table of opinions regarding abortion in relation to sex and status of the respondent. This table has the following structure:

```
> data("Abortion", package = "vcdExtra")
> str(Abortion)

 table [1:2, 1:2, 1:2] 171 152 138 167 79 148 112 133
 - attr(*, "dimnames")=List of 3
  ..$ Sex             : chr [1:2] "Female" "Male"
  ..$ Status          : chr [1:2] "Lo" "Hi"
  ..$ Support_Abortion: chr [1:2] "Yes" "No"
```

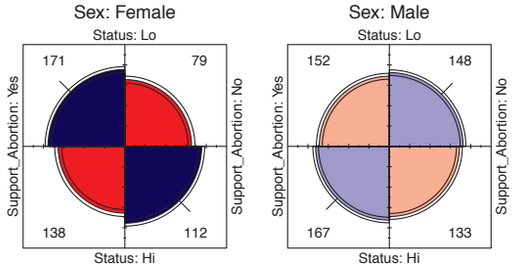
(a) Taking support for abortion as the outcome variable, produce fourfold displays showing the association with sex, stratified by status.
★

```
> data("Abortion", package="vcdExtra")
> structable(Abortion)

                         Status  Lo  Hi
Sex    Support_Abortion
Female Yes                     171 138
       No                       79 112
Male   Yes                     152 167
       No                      148 133

> Abortion2<-aperm(Abortion, c(1,3,2))
> fourfold(Abortion2)
```



(b) Do the same for the association of support for abortion with status, stratified by sex.
★

```
> Abortion3<-aperm(Abortion, c(2,3,1))
> fourfold(Abortion3)
```



(c) For each of the problems above, use `oddsratio()` to calculate the numerical values of the odds ratio, as stratified in the question.
★

```
>       # Sex by support for abortion, stratified by status
> summary(oddsratio(Abortion2))

z test of coefficients:

                     Estimate Std. Error z value Pr(>|z|)
Female:Male/Yes:No|Lo   0.7455     0.1784    4.18  2.9e-05 ***
Female:Male/Yes:No|Hi  -0.0189     0.1723   -0.11     0.91
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

>       # Status by support for abortion, stratified by sex
> summary(oddsratio(Abortion3))

z test of coefficients:

                    Estimate Std. Error z value Pr(>|z|)
Lo:Hi/Yes:No|Female    0.563      0.186    3.03   0.0025 **
Lo:Hi/Yes:No|Male     -0.201      0.164   -1.23   0.2199
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(d) Write a brief summary of how support for abortion depends on sex and status.
★ Regardless of status, more women say no to abortion and more men say yes. Regardless of gender, more low status say no and more high status say yes.

Exercise 4.3 The *JobSat* table on income and job satisfaction created in Example 2.5 is contained in the vcdExtra package.

(a) Carry out a standard χ^2 test for association between income and job satisfaction. Is there any indication that this test might not be appropriate? Repeat this test using `simulate.p.value = TRUE` to obtain a Monte Carlo test that does not depend on large sample size. Does this change your conclusion?
★

(b) Both variables are ordinal, so CMH tests may be more powerful here. Carry out that analysis. What do you conclude?
★

Exercise 4.4 The *Hospital* data in vcd gives a 3×3 table relating the length of stay (in years) of 132 long-term schizophrenic patients in two London mental hospitals with the frequency of visits by family and friends.

(a) Carry out a χ^2 test for association between the two variables.
★

```
> data("Hospital", package="vcd")
> chisq.test(Hospital)

Pearson's Chi-squared test

data:  Hospital
X-squared = 35.2, df = 4, p-value = 4.3e-07
```

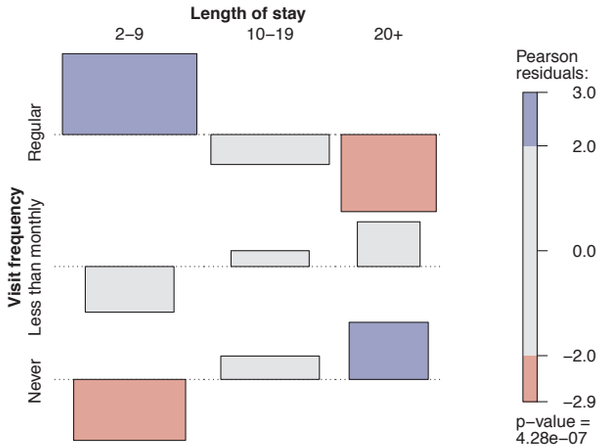
(b) Use `assocstats()` to compute association statistics. How would you describe the strength of association here?
★

```
> assocstats(Hospital)

                    X^2 df   P(> X^2)
Likelihood Ratio 38.353  4 9.4755e-08
Pearson          35.171  4 4.2842e-07

Phi-Coefficient   : NA
Contingency Coeff.: 0.459
Cramer's V        : 0.365
```

By the contingency coefficient, there is moderately strong association between the length of stay long-term schizophrenic patients and the frequency of visits by family and friends.

(c) Produce an association plot for these data, with visit frequency as the vertical variable. Describe the pattern of the relation you see here.
★

```
> assoc(Hospital, shade=TRUE)
```



(d) Both variables can be considered ordinal, so `CMHtest()` may be useful here. Carry out that analysis. Do any of the tests lead to different conclusions?
★

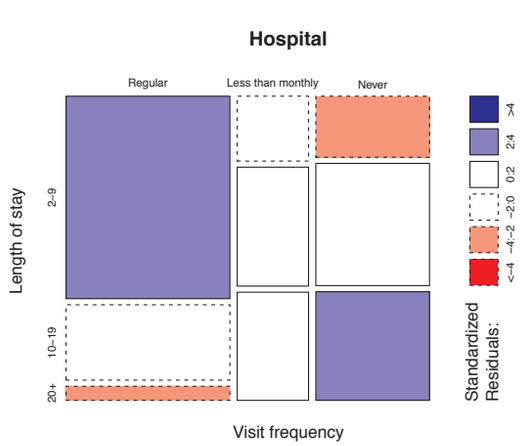
```
> CMHtest(Hospital)

Cochran-Mantel-Haenszel Statistics for Visit frequency by Length of stay

                 AltHypothesis Chisq Df     Prob
cor         Nonzero correlation  29.1  1 6.74e-08
rmeans   Row mean scores differ  34.4  2 3.40e-08
cmeans   Col mean scores differ  29.6  2 3.72e-07
general     General association  34.9  4 4.86e-07
```

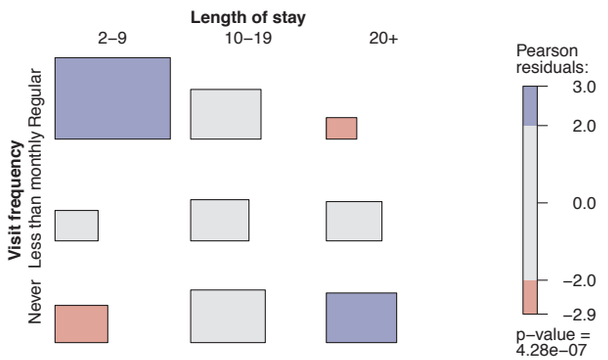
The results of all four tests lead to the same conclusion that there is a significant association between the visit frequency and the length of stay. The test for non-zero correlation, trteating both variables as ordinal, has the largest ratio of χ^2/df.

Exercise 4.5 Continuing with the *Hospital* data:

(a) Try one or more of the following other functions for visualizing two-way contingency tables with this data: `plot()`, `tile()`, `mosaic()`, and `spineplot()`. [For all except `spineplot()`, it is useful to include the argument `shade=TRUE`].
★

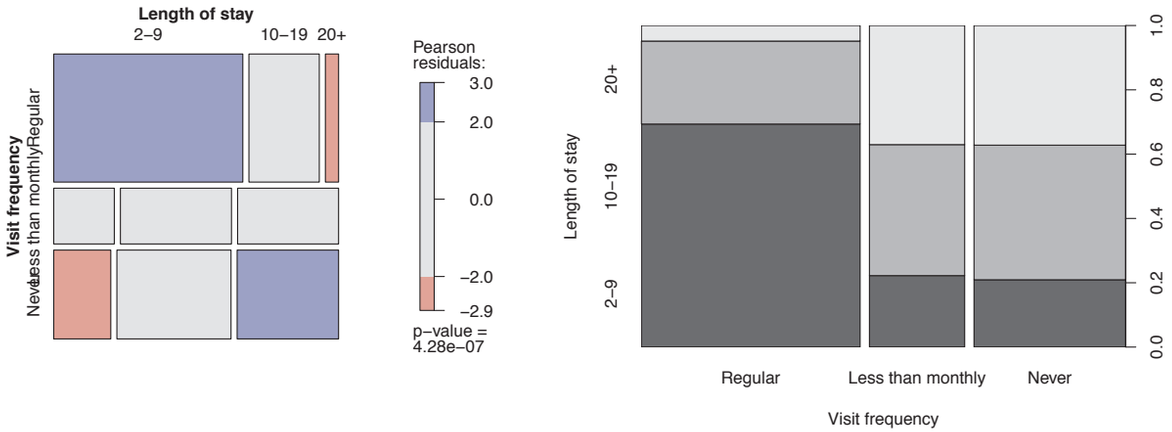
```
> plot(Hospital, shade=TRUE)
> tile(Hospital, shade=TRUE)
```





★

```
> mosaic(Hospital, shade=TRUE)
> spineplot(Hospital)
```



(b) Comment on the differences among these displays for understanding the relation between visits and length of stay.
★

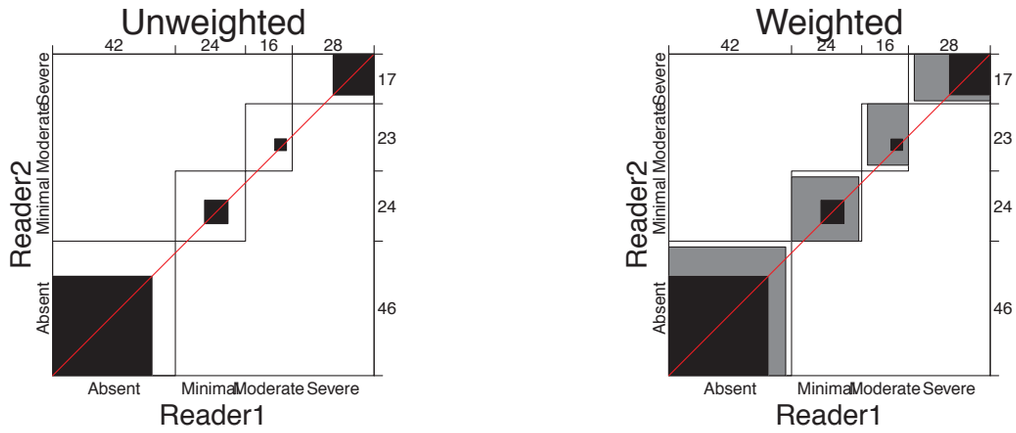
Exercise 4.6 The two-way table *Mammograms* in vcdExtra gives ratings on the severity of diagnosis of 110 mammograms by two raters.

(a) Assess the strength of agreement between the raters using Cohen's κ, both unweighted and weighted.
★ Both unweighted and weighted κ indicate substantial agreement. Fleiss-Cohen weights give greater weight to the "near-misses," so gives a larger value.

```
> Kappa(Mammograms)

           value    ASE     z Pr(>|z|)
Unweighted 0.371 0.0603  6.15 7.56e-10
Weighted   0.596 0.0492 12.11 8.90e-34

> Kappa(Mammograms, weights= "Fleiss-Cohen")

           value    ASE     z Pr(>|z|)
Unweighted 0.371 0.0603  6.15 7.56e-10
Weighted   0.764 0.0400 19.12 1.67e-81
```

(b) Use `agreementplot()` for a graphical display of agreement here.
★ The default for `agreementplot()` shows the weighted display, so you can use `weights=1` for the unweighted version.

```
> agreementplot(Mammograms, main="Unweighted", weights=1)
> agreementplot(Mammograms, main="Weighted")
```



(c) Compare the Kappa measures with the results from `assocstats()`. What is a reasonable interpretation of each of these measures?

★ The contingency coefficient and Cramer's V assess only association, so these could be large when there is little agreement. The values of these statistics are not directly comparable.

```
> assocstats(Mammograms)

                     X^2 df   P(> X^2)
Likelihood Ratio 92.619  9 4.4409e-16
Pearson          83.516  9 3.2307e-14

Phi-Coefficient   : NA
Contingency Coeff.: 0.657
Cramer's V        : 0.503
```

Exercise 4.7 Agresti and Winner (1997) gave the data in Table 4.1 on the ratings of 160 movies by the reviewers Gene Siskel and Roger Ebert for the period from April 1995 through September 1996. The rating categories were Con ("thumbs down"), Mixed, and Pro ("thumbs up").

Table 4.1: Movie ratings by Siskel & Ebert, April 1995–September 1996. *Source*: Agresti and Winner (1997)

		Ebert			
		Con	Mixed	Pro	Total
Siskel	Con	24	8	13	45
	Mixed	8	13	11	32
	Pro	10	9	64	83
	Total	42	30	88	160

(a) Assess the strength of agreement between the raters using Cohen's κ, both unweighted and weighted.

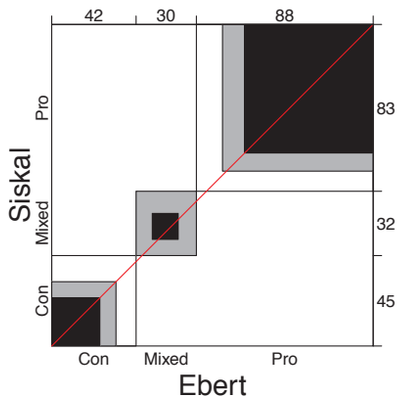
★

```
> ratings <- matrix(
+ c(
+   24,  8, 13,
+    8, 13, 11,
+   10,  9, 64
+  ), 3, 3, byrow=TRUE)
> dimnames(ratings) <- list(Siskal=c("Con", "Mixed", "Pro"),
+                           Ebert =c("Con", "Mixed", "Pro"))
> Kappa(ratings)

            value    ASE    z Pr(>|z|)
Unweighted 0.389 0.0598 6.50 7.87e-11
Weighted   0.427 0.0635 6.72 1.78e-11
```

(b) Use `agreementplot()` for a graphical display of agreement here.

★ The agreement plot shows that both Siskal and Ebert most commonly gave "thumbs up" ratings where they largely agreed. Their ratings differed most when one of them gave a Mixed rating.

```
> agreementplot(ratings)
```



(c) Assess the hypothesis that the ratings are *symmetric* around the main diagonal, using an appropriate χ^2 test. *Hint*: Symmetry for a square table $\boldsymbol{T}$ means that $t_{ij} = t_{ji}$ for $i \neq j$. The expected frequencies under the hypothesis of symmetry are the average of the off-diagonal cells, $\boldsymbol{E} = (\boldsymbol{T} + \boldsymbol{T}^{\mathsf{T}})/2$.
★

```
> T <- (ratings + t(ratings))/2
> (Chisq <- sum((ratings - T)^2 / T))

[1] 0.5913

> df <- nrow(T) * (nrow(T)-1) /2
> pchisq(Chisq,df, lower.tail = FALSE)

[1] 0.89842
```

(d) Compare the results with the output of `mcnemar.test()`.
★

```
> mcnemar.test(ratings)

McNemar's Chi-squared test

data:  ratings
McNemar's chi-squared = 0.591, df = 3, p-value = 0.9
```

Exercise 4.8 For the *VisualAcuity* data set:

(a) Use the code shown in the text to create the table form, `VA.tab`.
★

```
> data("VisualAcuity", package = "vcd")
> VA <- xtabs(Freq ~ right + left + gender, data = VisualAcuity)
> dimnames(VA)[1:2] <- list(c("high", 2, 3, "low"))
> names(dimnames(VA))[1:2] <- paste(c("Right", "Left"), "eye grade")
```

(b) Perform the CMH tests for this table.
★

```
> CMHtest(VA)

$`gender:male`
Cochran-Mantel-Haenszel Statistics for Right eye grade by Left eye grade
in stratum gender:male

                 AltHypothesis Chisq Df Prob
cor        Nonzero correlation  1555  1    0
rmeans  Row mean scores differ  1556  3    0
```

```
cmeans  Col mean scores differ  1557  3    0
general    General association  3303  9    0


$`gender:female`
Cochran-Mantel-Haenszel Statistics for Right eye grade by Left eye grade
in stratum gender:female

                 AltHypothesis Chisq Df Prob
cor         Nonzero correlation  3691  1    0
rmeans  Row mean scores differ  3709  3    0
cmeans  Col mean scores differ  3724  3    0
general    General association  8096  9    0
```

(c) Use the `woolf_test()` described in Section 4.3.2 to test whether the association between left and right eye acuity can be considered the same for men and women.
★ The Woolf test gives no evidence that the association differs for men and women.

```
> woolf_test(VA)

Woolf-test on Homogeneity of Odds Ratios (no 3-Way assoc.)

data:  VA
X-squared = 0.0892, df = 1, p-value = 0.77
```

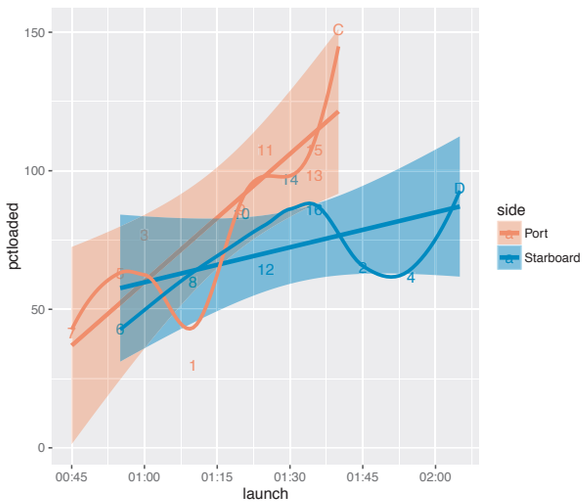
Exercise 4.9 The graph in Figure 4.23 may be misleading, in that it doesn't take into account of the differing capacities of the 18 life boats on the *Titanic*, given in the variable `cap` in the *`Lifeboats`* data.

(a) Calculate a new variable, `pctloaded`, as the percentage loaded relative to the boat capacity.
★

```
> Lifeboats$pctloaded <- with(Lifeboats, 100*total/cap)
```

(b) Produce a plot similar to Figure 4.23, showing the changes over time in this measure.
★ Boats on the port side began loading earlier, but were initally less than half full. Boats launched from the starbord side were more consistent over time. With percent loaded (0–100) as the response, a linear model is only a rough approximation.

```
> library(ggplot2)
> AES <- aes(x = launch, y = pctloaded, colour = side, label = boat)
> ggplot(data = Lifeboats, mapping = AES) +
+      geom_text() +
+      geom_smooth(method = "lm", aes(fill = side), size = 1.5) +
+      geom_smooth(method = "loess", aes(fill = side), se = FALSE,
+                  size = 1.2)
```



Test a model allowing different slopes and intercepts for port and starboard sides:

```
> summary(lm(pctloaded ~ side * as.numeric(launch), data=Lifeboats))
```

```
Call:
lm(formula = pctloaded ~ side * as.numeric(launch), data = Lifeboats)

Residuals:
   Min     1Q Median     3Q    Max
-45.34 -11.78   0.34  11.46  29.61

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     4.67e+07   1.10e+07    4.24  0.00083 ***
sideStarboard                  -3.39e+07   1.48e+07   -2.29  0.03800 *
as.numeric(launch)              2.56e-02   6.05e-03    4.24  0.00083 ***
sideStarboard:as.numeric(launch) -1.86e-02   8.12e-03   -2.29  0.03800 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 20.2 on 14 degrees of freedom
Multiple R-squared:  0.602,Adjusted R-squared:  0.517
F-statistic: 7.06 on 3 and 14 DF,  p-value: 0.004
```

Chapter 5 Mosaic Displays for n-Way Tables

Exercise 5.1 The data set *criminal* in the package logmult (Bouchet-Valat, 2015) gives the 4×5 table below of the number of men aged 15–19 charged with a criminal case for whom charges were dropped in Denmark from 1955–1958.

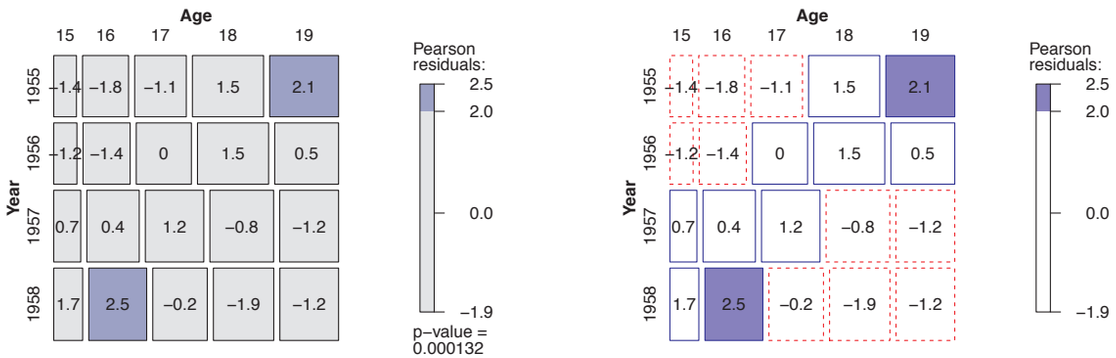
```
> data("criminal", package = "logmult")
> criminal

      Age
Year    15  16  17  18  19
  1955 141 285 320 441 427
  1956 144 292 342 441 396
  1957 196 380 424 462 427
  1958 212 424 399 442 430
```

(a) Use `loglm()` to test whether there is an association between `Year` and `Age`. Is there evidence that dropping of charges in relation to age changed over the years recorded here?
★ There is a significant association between `Year` and `Age`, so the row profiles of proportions differ over year.

```
> loglm(~Year + Age, data=criminal)

Call:
loglm(formula = ~Year + Age, data = criminal)

Statistics:
                     X^2 df   P(> X^2)
Likelihood Ratio 38.245 12 0.00014004
Pearson          38.410 12 0.00013155
```

(b) Use `mosaic()` with the option `shade=TRUE` to display the pattern of signs and magnitudes of the residuals. Compare this with the result of `mosaic()` using "Friendly shading," from the option `gp=shading_Friendly`. Describe verbally what you see in each regarding the pattern of association in this table.
★ It is helpful here to display all the residual contributions to association in the mosaic display using `labeling=labeling_residuals`.

```
> mosaic(criminal, shade=TRUE,
+        labeling=labeling_residuals, suppress=0)
> mosaic(criminal, gp=shading_Friendly,
+        labeling=labeling_residuals, suppress=0)
```



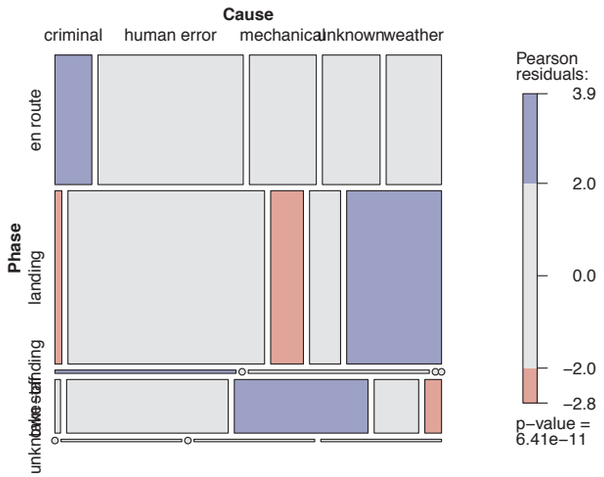
Although only two residuals exceed the default $|r_{ij}| > 2$ threshold for shading, there is clearly a systematic association between year and age shown by the signs of the residuals.
The Friendly shading option here gives a better picture of the pattern of associations, showing positive and negative residuals in the diagonally opposite corners of the plot. See Exercise 6.2 for further analysis of this data.

Exercise 5.2 The data set *AirCrash* in vcdExtra gives a database of all crashes of commercial airplanes between 1993–2015, classified by `Phase` of the flight and `Cause` of the crash. How can you best show is the nature of the association between these variables in a mosaic plot? Start by making a frequency table, `aircrash.tab`:

```
> data("AirCrash", package = "vcdExtra")
> aircrash.tab <- xtabs(~ Phase + Cause, data = AirCrash)
```

(a) Make a default mosaic display of the data with `shade=TRUE` and interpret the pattern of the high-frequency cells.
★

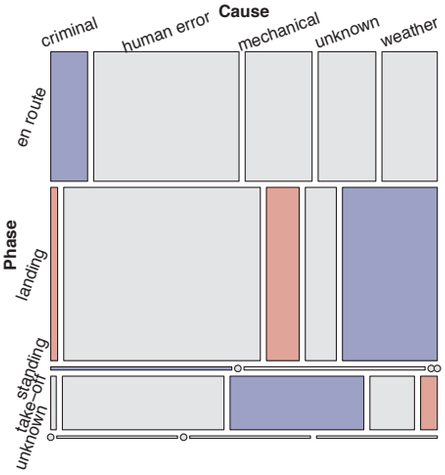
```
>  mosaic(aircrash.tab, shade=TRUE)
```

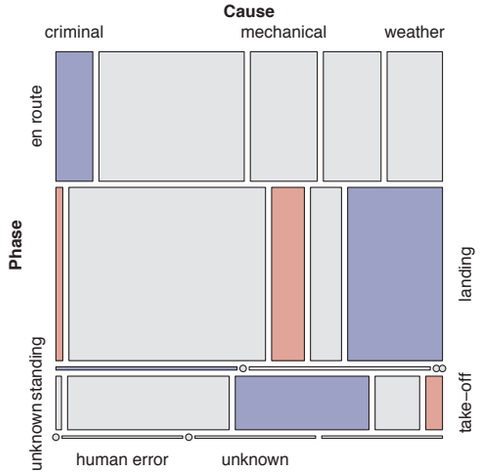


Four cells stand out as having greater than expected frequencies, if Phase and Cause were independent. Both take-off and en-route are positively associated with criminal activities. Crashes in landing are more associated with weather. It is difficult to interpret the unknown cells.

(b) The default plot has overlapping labels due to the uneven marginal frequencies relative to the lengths of the category labels. Experiment with some of the `labeling_args` options (`abbreviate`, `rot_labels`, etc.) to see if you can make the plot more readable. *Hint*: a variety of these are illustrated in Section 4.1 of `vignette("strucplot")`
★ Here are two alternatives that reduce the overplotting of labels:

```
> mosaic(aircrash.tab, shade=TRUE, rot_labels=c(20,90,0,70), legend=FALSE)
> mosaic(aircrash.tab, shade=TRUE, alternate_labels=TRUE, legend=FALSE)
```

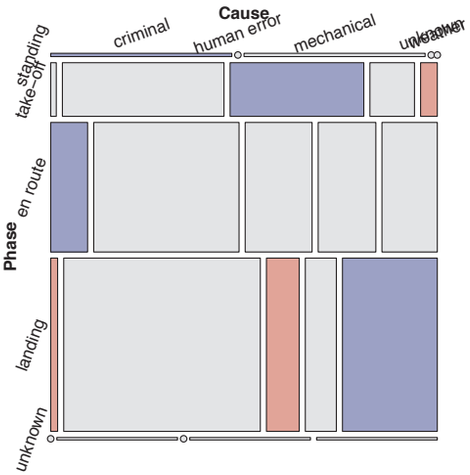


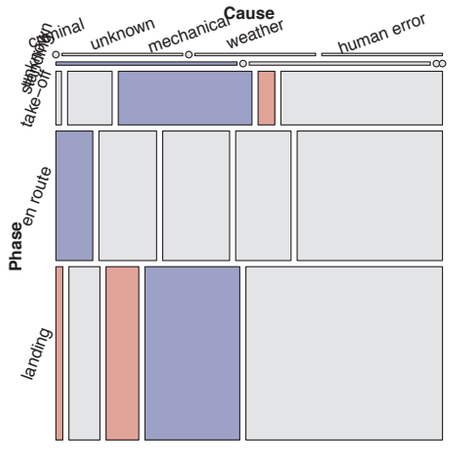


(c) The levels of `Phase` and `Cause` are ordered alphabetically (because they are factors). Experiment with other orderings of the rows/columns to make interpretation clearer, e.g., ordering `Phase` temporally or ordering both factors by their marginal frequency.

★ Ordering by `Phase` is slightly easier to interpret. Ordering both variables by marginal frequencies is also slightly better than the default, except that it leads to more overplotting of the labels.

```
> # reorder Phase temporarally
> roworder <- c(3, 4, 1, 2, 5)
> mosaic(aircrash.tab[roworder,], shade=TRUE, rot_labels=c(20,90,0,70), legend=FALSE)
>
> # marginal frequencies
> roworder <- order(rowSums(aircrash.tab))
> colorder <- order(colSums(aircrash.tab))
> mosaic(aircrash.tab[roworder, colorder], shade=TRUE, rot_labels=c(20,90,0,70), legend=FALSE)
```

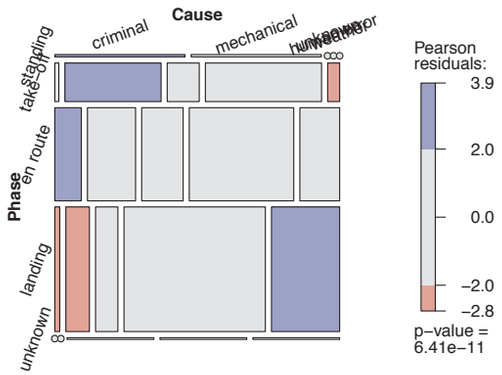




The best general approach, as was illustrated in Figure 1.10, uses ***effect ordering*** to order the factors according to their associations. One easy method for this (Friendly and Kwan, 2003) is to order the factor levels according to their scores on the first dimension of a correspondence analysis solution as illustrated below. This maximizes an opposite corner pattern of the residuals.

```
> library(ca)
> aircrash.ca <- ca(aircrash.tab)
> summary(aircrash.ca, rows=FALSE, columns=FALSE)

Principal inertias (eigenvalues):

 dim    value      %    cum%   scree plot
 1      0.123002  65.6  65.6   ****************
```

```
 2      0.051548  27.5  93.1  *******
 3      0.012340   6.6  99.7  **
 4      0.000562   0.3 100.0
        -------- -----
 Total: 0.187452 100.0

> # reorder by CA coordinates on Dim 1
> roworder <- order(aircrash.ca$rowcoord[,"Dim1"])
> colorder <- order(aircrash.ca$colcoord[,"Dim1"])
> aircrash.tab[roworder, colorder]

          Cause
Phase      criminal mechanical unknown human error weather
  standing        2          2       0           0       0
  take-off        1         24       8          29       3
  en route       16         29      25          63      24
  landing         4         19      18         114      55
  unknown         0          0       1           1       1

> mosaic(aircrash.tab[roworder, colorder], shade=TRUE, rot_labels=c(20,90,0,70))
```



Exercise 5.3 The Lahman package contains comprehensive data on baseball statistics for Major League Baseball from 1871 through 2012. For all players, the *Master* table records the handedness of players, in terms of throwing (L, R) and batting (B, L, R), where B indicates "both." The table below was generated using the following code:

```
> library(Lahman)
> data("Master", package = "Lahman")
> basehands <- with(Master, table(throws, bats))
```

	Bats		
Throws	B	L	R
L	177	2640	527
R	924	1962	10442

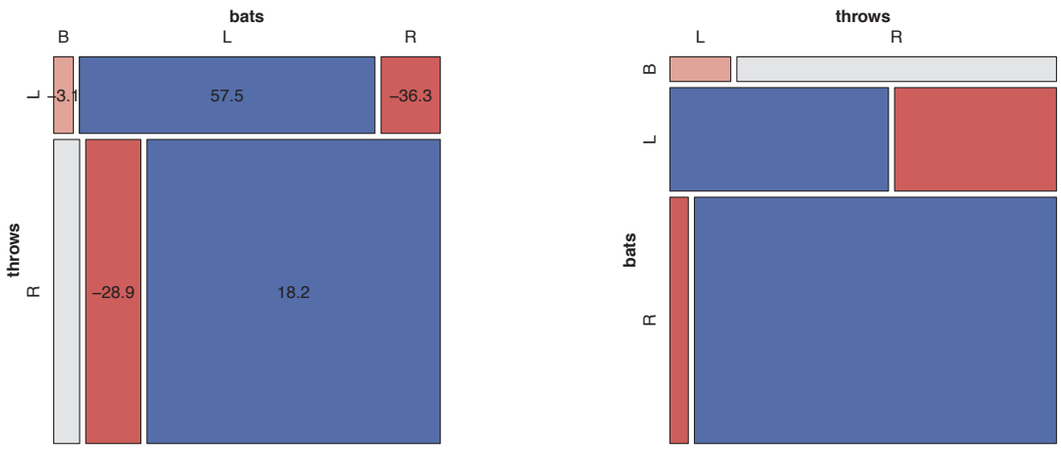
- Use the code above, or else enter these data into a frequency table in R.
  ★ These notes use a later version of the Lahman package (v. 4.0-1) with the code above, so the numbers used in the plots don't correspond to those in the table. The current version of the table is shown below.

```
> basehands

      bats
throws     B     L     R
     L   182  2742   550
     R   956  2031 10814
```

- Construct mosaic displays showing the relation of batting and throwing handedness, split first by batting and then by throwing.

★ By default, a two-way contingency table is split first by the row variable, then by the column variable. So, to split the other way, you can use `t()` on the table argument.

```
> mosaic(basehands, shade=TRUE, labeling=labeling_residuals(), legend=FALSE)
> mosaic(t(basehands), direction=c("h", "v"), shade=TRUE, legend=FALSE)
```



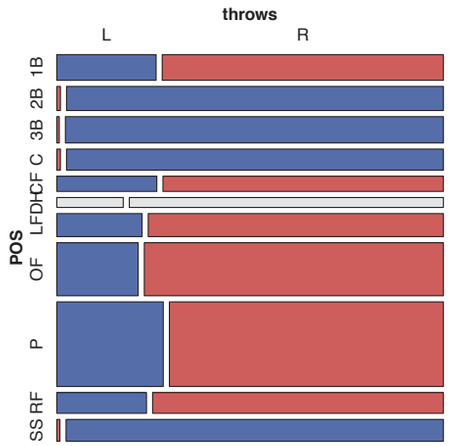
- From these displays, what can be said about players who throw with their left or right hands in terms of their batting handedness?
  ★ Players who throw with their left or right hands are most likely to bat in the same way. From the values of the cell residuals, left handers are more likely to be uni-handers than righties.

Exercise 5.4 * A related analysis concerns differences in throwing handedness among baseball players according to the fielding position they play. The following code calculates such a frequency table.

```
> library(Lahman)
> MasterFielding <- data.frame(merge(Master, Fielding, by = "playerID"))
> throwPOS <- with(MasterFielding, table(POS, throws))
```

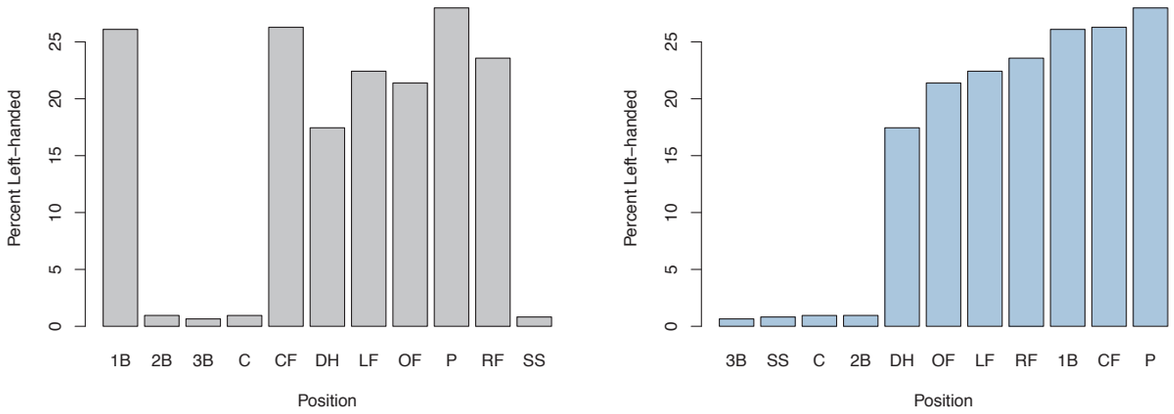
(a) Make a mosaic display of throwing hand vs. fielding position.
★ There is clearly a very strong association between throwing hand and fielding position. A peculiarity of the data is that designated hitters (DH) do not play a fielding position, but instead fill in for the pitcher in the batting order, so throwing hand is not really relevant here. This position might arguably be deleted from this analysis.

```
> mosaic(throwPOS, shade=TRUE, legend=FALSE)
```



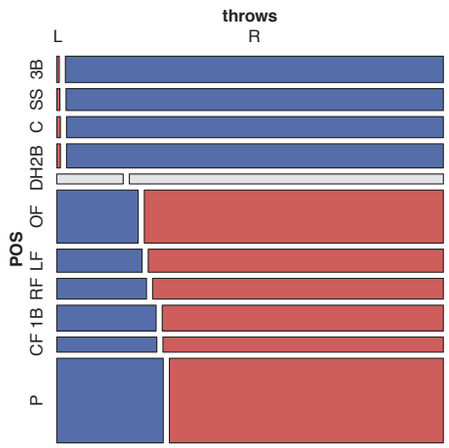
(b) Calculate the percentage of players throwing left-handed by position. Make a sensible graph of this data.
★ A barplot is simple and reasonable here. However, the levels of fielding position are ordered alphabetically, which makes interpretation harder. Sorting by `pctLeft` is better.

```
> pctLeft <- 100 * throwPOS[,1] / rowSums(throwPOS)
> pctLeft

    1B     2B     3B      C     CF     DH     LF     OF      P     RF     SS
26.100  0.955  0.650  0.952 26.284 17.449 22.419 21.385 27.984 23.571  0.818

> ord <- order(pctLeft)
> barplot(pctLeft, xlab="Position", ylab="Percent Left-handed")
> barplot(pctLeft[ord], xlab="Position", ylab="Percent Left-handed", col="lightblue")
```



(c) Re-do the mosaic display with the positions sorted by percentage of left-handers.
★

```
> mosaic(throwPOS[ord,], shade=TRUE, legend=FALSE)
```



(d) Is there anything you can say about positions that have very few left-handed players?
★ All infield positions except for 1^{st} base have a very small percentage of players who throw left-handed. Given the marginal distributions of handeness and position, outfielders, pitchers and 1^{st} basemen are more likely to throw left-handed than if these variables were independent.

Exercise 5.5 For the *Bartlett* data described in Example 5.12, fit the model of no three-way association, H_4 in Table 5.2.

(a) Summarize the goodness of fit for this model, and compare to simpler models that omit one or more of the two-way terms.

★

(b) Use a mosaic-like display to show the lack of fit for this model.

★

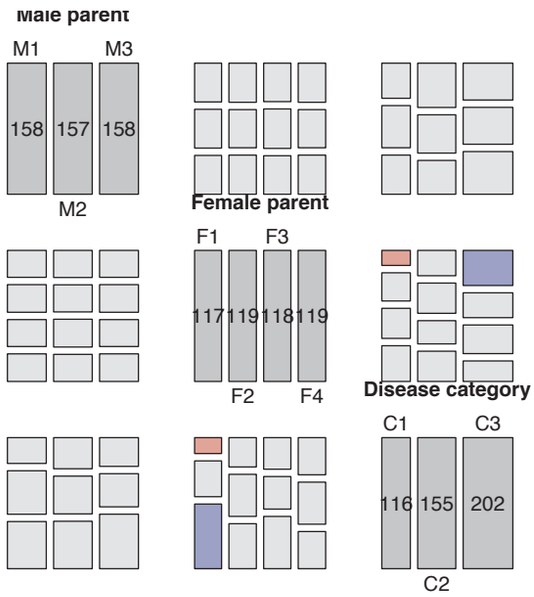
Exercise 5.6 Red core disease, caused by a fungus, is not something you want if you are a strawberry. The data set *jansen.strawberry* from the agridat (Wright, 2015) package gives a frequency data frame of counts of damage from this fungus from a field experiment reported by Jansen (1990). See the help file for details. The following lines create a $3 \times 4 \times 3$ table of crossings of 3 male parents with 4 (different) female parents, recording the number of plants in four blocks of 9 or 10 plants each showing red core disease in three ordered categories, C1, C2, or C3.

```
> data("jansen.strawberry", package = "agridat")
>
> dat <- jansen.strawberry
> dat <- transform(dat, category = ordered(category,
+                                          levels = c('C1','C2','C3')))
> levels(dat$male) <- paste0("M", 1:3)
> levels(dat$female) <- paste0("F", 1:4)
>
> jansen.tab <- xtabs(count ~ male + female + category, data = dat)
> names(dimnames(jansen.tab)) <- c("Male parent", "Female parent",
+                                  "Disease category")
> ftable(jansen.tab)

                         Disease category C1 C2 C3
Male parent Female parent
M1          F1                            6 13 20
            F2                            8 15 17
            F3                           13 10 16
            F4                            8 21 11
M2          F1                            5 13 21
            F2                            9 16 14
            F3                           16  9 15
            F4                           12 13 14
M3          F1                            5 10 24
            F2                           13 12 15
            F3                            3 14 22
            F4                           18  9 13
```

(a) Use `pairs(jansen.tab, shade=TRUE)` to display the pairwise associations among the three variables. Describe how disease category appears to vary with male and female parent. Why is there no apparent association between male and female parent?

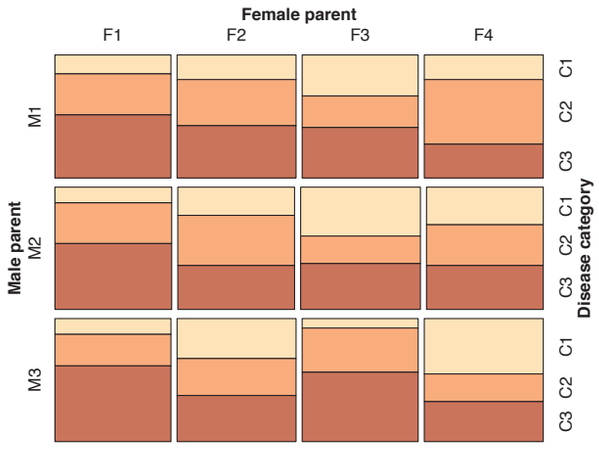
★ This was a designed experiment, with male and female parents completely crossed to create 12 populations. Disease categories seem to be associated with female parents, with more serious disease (C3) more prevalent in parent F4.

```
> library(vcd)
> pairs(jansen.tab, shade=TRUE)
```



(b) As illustrated in Figure 5.6, use `mosaic()` to prepare a 3-way mosaic plot with the tiles colored in increasing shades of some color according to disease category. Describe the pattern of category C3 in relation to male and female parent. (Hint: the `highlighting` arguments are useful here.)
★

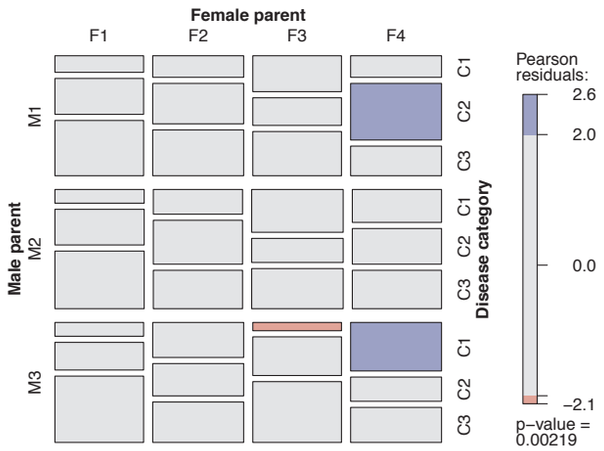
```
> cols <- c("moccasin","lightsalmon1","indianred")
> mosaic(jansen.tab, highlighting=3, highlighting_fill=cols)
```



(c) With `category` as the response variable, the minimal model for association is [MF][C], or `~ 1*2 + 3`. Fit this model using `loglm()` and display the residuals from this model with `mosaic()`. Describe the pattern of lack of fit of this model.
★

```
> # baseline model
> library(MASS)
> loglm(~1*2+3,data=jansen.tab)

Call:
loglm(formula = ~1 * 2 + 3, data = jansen.tab)

Statistics:
                     X^2 df  P(> X^2)
Likelihood Ratio 45.63 22 0.0022062
Pearson          45.65 22 0.0021927
```

```
> mosaic(jansen.tab, expected = ~ 1*2+3)
```



Exercise 5.7 The data set *caith* in MASS (Ripley, 2015) gives another classic 4×5 table tabulating hair color and eye color, this for people in Caithness, Scotland, originally from Fisher (1940). The data is stored as a data frame of cell frequencies, whose rows are eye colors and whose columns are hair colors.

```
> data("caith", package = "MASS")
> caith

       fair red medium dark black
blue    326  38    241  110     3
light   688 116    584  188     4
medium  343  84    909  412    26
dark     98  48    403  681    85
```

(a) The `loglm()` and `mosaic()` functions don't understand data in this format, so use `Caith <- as.matrix(caith)` to convert to array form. Examine the result, and use `names(dimnames(Caith))<-c()` to assign appropriate names to the row and column dimensions.
★

```
> Caith <- as.matrix(caith)
> dimnames(Caith)

[[1]]
[1] "blue"   "light"  "medium" "dark"

[[2]]
[1] "fair"   "red"    "medium" "dark"   "black"

> names(dimnames(Caith)) <- c("Eye", "Hair")
```

(b) Fit the model of independence to the resulting matrix using `loglm()`.
★

```
> (caith.mod <- loglm(~Hair+Eye, data=Caith, fitted=TRUE))

Call:
loglm(formula = ~Hair + Eye, data = Caith, fitted = TRUE)

Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 1218.3 12         0
Pearson          1240.0 12         0
```

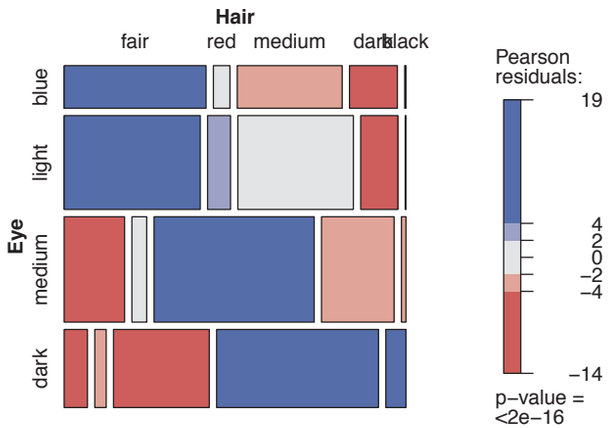
(c) Calculate and display the residuals for this model.
★

```
> residuals(caith.mod)
```

```
        Hair
Eye        fair     red medium   dark black
  blue     8.63 -0.0193  -2.67  -6.00 -3.94
  light   11.60  3.3125  -1.73 -12.19 -6.63
  medium  -6.56 -1.0691   7.40  -2.19 -2.20
  dark   -16.19 -2.7683  -5.41  16.28  8.46
```

(d) Create a mosaic display for this data.
★

```
> mosaic(Caith, shade=TRUE)
```



Exercise 5.8 The *HairEyePlace* data in vcdExtra gives similar data on hair color and eye color, for both Caithness and Aberdeen as a $4 \times 5 \times 2$ table.

(a) Prepare separate mosaic displays, one for each of Caithness and Aberdeen. Comment on any difference in the pattern of residuals.
★

(b) Construct conditional mosaic plots, using the formula `~ Hair + Eye | Place` and both `mosaic()` and `cotabplot()`. It is probably more useful here to suppress the legend in these plots. Comment on the difference in what is shown in the two displays.
★

Exercise 5.9 Bertin (1983, pp. 30–31) used a 4-way table of frequencies of traffic accident victims in France in 1958 to illustrate his scheme for classifying data sets by numerous variables, each of which could have various types and could be assigned to various visual attributes. His data are contained in *Accident* in vcdExtra, a frequency data frame representing his $5 \times 2 \times 4 \times 2$ table of the variables `age`, `result` (died or injured), `mode` of transportation, and `gender`.

```
> data("Accident", package = "vcdExtra")
> str(Accident, vec.len=2)

'data.frame': 80 obs. of  5 variables:
 $ age   : Ord.factor w/ 5 levels "0-9"<"10-19"<..: 5 5 5 5 5 ...
 $ result: Factor w/ 2 levels "Died","Injured": 1 1 1 1 1 ...
 $ mode  : Factor w/ 4 levels "4-Wheeled","Bicycle",..: 4 4 2 2 3 ...
 $ gender: Factor w/ 2 levels "Female","Male": 2 1 2 1 2 ...
 $ Freq  : int  704 378 396 56 742 ...
```

(a) Use `loglm()` to fit the model of mutual independence, `Freq ~ age+mode+gender+result` to this data set.
★ You can use `loglm()` directly on the frequency data frame, with `Freq` as the response:

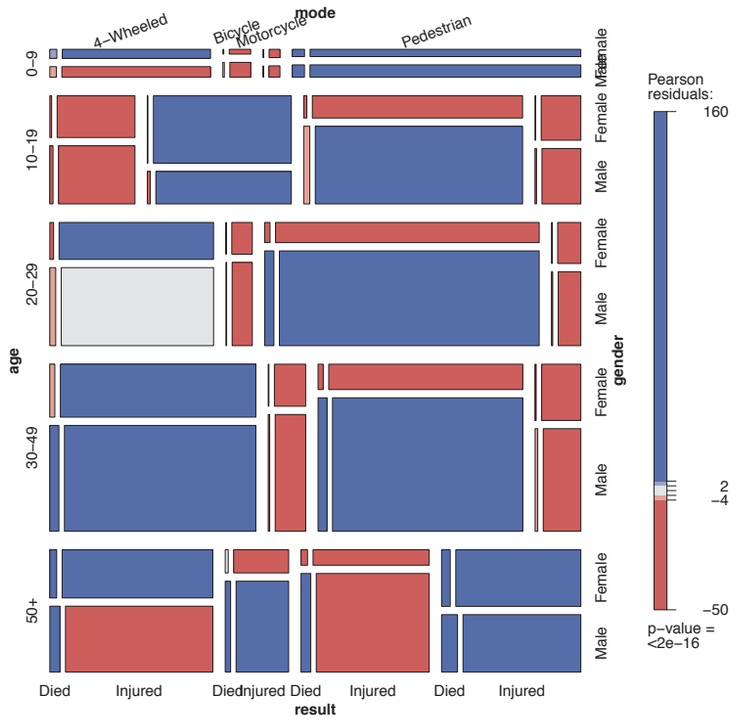
```
> library(MASS)
> loglm(Freq ~ age + mode + gender + result, data = Accident)

Call:
loglm(formula = Freq ~ age + mode + gender + result, data = Accident)

Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 60320 70         0
Pearson          76865 70         0
```

Or, convert to an array first with `xtabs()`

```
> accident.tab <- xtabs(Freq ~ age + mode + gender + result, data = Accident)
> loglm(~ age + mode + gender + result, data = accident.tab)

Call:
loglm(formula = ~age + mode + gender + result, data = accident.tab)

Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 60320 70         0
Pearson          76865 70         0
```

(b) Use `mosaic()` to produce an interpretable mosaic plot of the associations among all variables under the model of mutual independence. Try different orders of the variables in the mosaic. (*Hint*: the `abbreviate` component of the `labeling_args` argument to `mosaic()` will be useful to avoid some overlap of the category labels.)
★ In this data set, `mode` is arguably an ordered factor, and better results will come from reordering its levels, from Pedestrian to 4-Wheeled vehicle. The order of variables given in `xtabs()` gives a reasonable result. The label overlap can be avoided by rotating the labels for `mode`.

```
> Accident$mode <- ordered(Accident$mode,
+     levels=levels(Accident$mode)[c(4,2,3,1)])
> mosaic(accident.tab, shade=TRUE, rot_labels = c(20, 90, 00, 90))
```



(c) Treat `result` (`"Died"` vs. `"Injured"`) as the response variable, and fit the model `Freq ~ age*mode*gender + result` that asserts independence of `result` from all others jointly.
★ This fits much better than the mutual independence model, but still has a terrible fit. There still remain important associations between `result` and the other variables

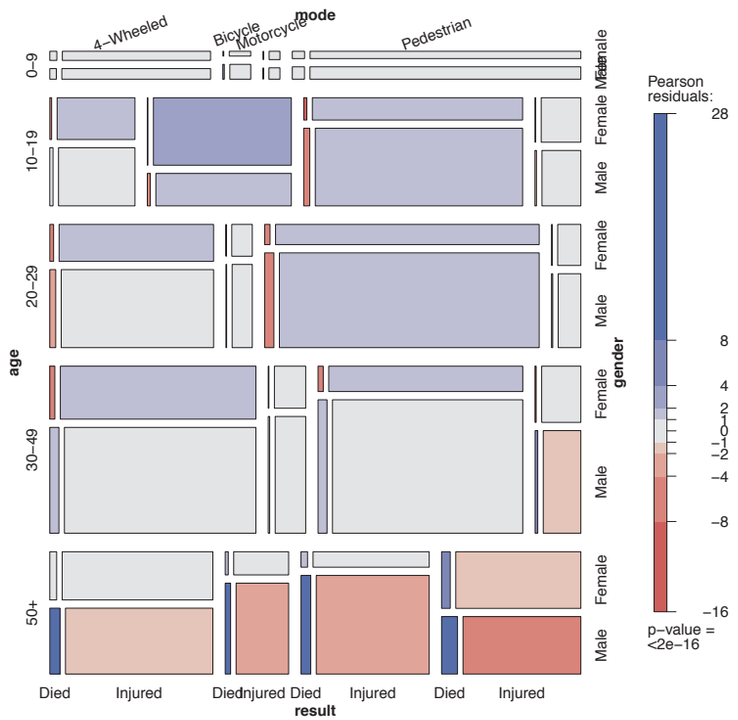
```
> loglm(Freq ~ age * mode * gender + result, data = Accident)
```

```
Call:
loglm(formula = Freq ~ age * mode * gender + result, data = Accident)

Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 2217.7 39        0
Pearson          2347.6 39        0
```

(d) Construct a mosaic display for the residual associations in this model. Which combinations of the predictor factors are more likely to result in death?

★ The largest postive residuals appear in the 50+ age group, where males are more likely to have died, regardless of `mode`. It can also be seen that in the 30–49 age group, more males die in bicycle and motorcycle accidents. Other permutations of the table variables or other displays like doubledecker plots can highlight other features.

```
> mosaic(accident.tab, expected = ~age * mode * gender + result,
+        shade=TRUE, rot_labels = c(20, 90, 00, 90),
+        gp_args=list(interpolate=c(1,2,4,8)))
```



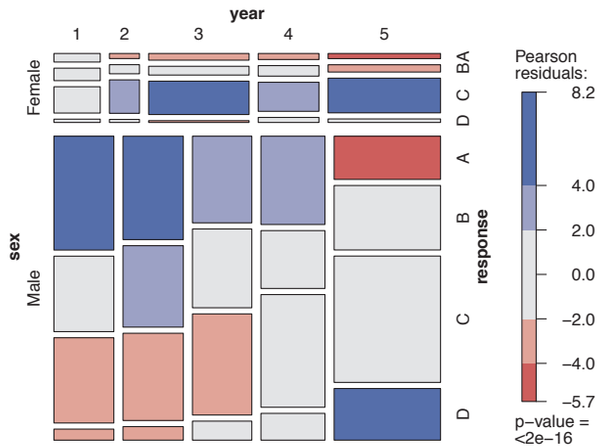
Exercise 5.10 The data set *Vietnam* in vcdExtra gives a $2 \times 5 \times 4$ contingency table in frequency form reflecting a survey of student opinion on the Vietnam War at the University of North Carolina in May 1967. The table variables are sex, year in school, and response, which has categories: (A) Defeat North Vietnam by widespread bombing and land invasion; (B) Maintain the present policy; (C) De-escalate military activity, stop bombing and begin negotiations; (D) Withdraw military forces immediately. How does the chosen response vary with sex and year?

```
> data("Vietnam", package = "vcdExtra")
> str(Vietnam)

'data.frame': 40 obs. of  4 variables:
 $ sex     : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
 $ year    : int  1 1 1 1 2 2 2 2 3 3 ...
 $ response: Factor w/ 4 levels "A","B","C","D": 1 2 3 4 1 2 3 4 1 2 ...
 $ Freq    : int  13 19 40 5 5 9 33 3 22 29 ...
```

(a) With `response` (R) as the outcome variable and `year` (Y) and `sex` (S) as predictors, the minimal baseline loglinear model is the model of joint independence, [R][YS]. Fit this model, and display it in a mosaic plot.

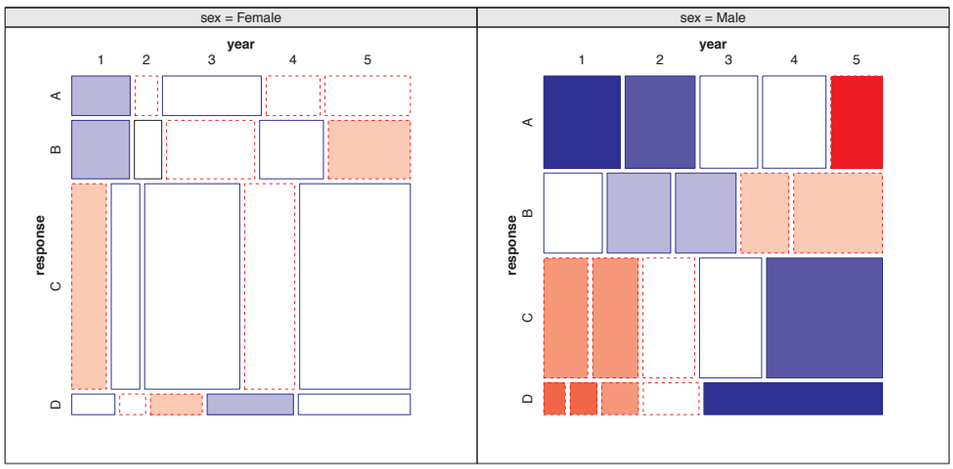
★ The baseline model does not fit well, so `response` is associated with either `year` or `sex` or both. Note that when fitting with `loglm()` (or `glm()`), both `mosaic(mod)` and `plot(mod)` give the corresponding mosaic plot for a model object, `mod`.

```
> library(MASS)
> library(vcdExtra)
> (viet.mod <- loglm(Freq ~ sex * year + response, data=Vietnam))

Call:
loglm(formula = Freq ~ sex * year + response, data = Vietnam)

Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 361.72 27        0
Pearson          366.36 27        0

> mosaic(viet.mod)
> # plot(viet.mod)     # same
```



(b) Construct conditional mosaic plots of the `response` versus `year` separately for males and females. Describe the associations seen here.
★ We use `cotabplot()` here, with the formula `~ response + year | sex` to condition on sex. For males, there is a clear association between response and year, with 1^{st} year students prefering the more military response A, and later-year students tending toward the more peaceful response D. For females, the association between response and year is weak, and does not show a coherent pattern.

```
> cotabplot(~ response + year | sex, data=vietnam.tab,
+                                gp=shading_Friendly, gp_args=list(interpolate=1:4), legend=FALSE)
```



(c) Follow the methods shown in Example 5.10 to fit separate models of independence for the levels of `sex`, and

the model of conditional independence, $R \perp Y \mid S$. Verify that the decomposition of G^2 in Eqn. (5.6) holds for these models.

★ Splitting the data by sex, using `apply()` shows that the association between response and year is strongly significant for males, but not for females.

```
> mods.list <-
+         apply(vietnam.tab, "sex",
+               function(x) loglm(~ response + year, data=x))
> mods.list

$Female
Call:
loglm(formula = ~response + year, data = x)

Statistics:
                    X^2 df P(> X^2)
Likelihood Ratio 13.263 12  0.35022
Pearson          13.294 12  0.34803

$Male
Call:
loglm(formula = ~response + year, data = x)

Statistics:
                    X^2 df P(> X^2)
Likelihood Ratio 203.05 12        0
Pearson          198.50 12        0
```

(d) Construct a useful 3-way mosaic plot of the data for the model of conditional independence.

★

Exercise 5.11 Consider the models for 4-way tables shown in Table 5.3.

(a) For each model, give an independence interpretation. For example, the model of mutual independence corresponds to $A \perp B \perp C \perp D$.

★ The basic idea of the notation is that terms in separate []s are said to be independent under a given model. Variables within a [] term are allowed to be associated

- mutual: [A] [B] [C] [D] $\leftrightarrow A \perp B \perp C \perp D$
- joint: [ABC] [D] $\leftrightarrow (ABC) \perp D$
- conditional: [AD] [BD] [CD] $\leftrightarrow (AD) \perp (BD) \perp (CD)$
- markov (order 1): [AB] [BC] [CD] $\leftrightarrow (AB) \perp (BC) \perp (CD)$
- markov (order 2): [ABC] [BCD] $\leftrightarrow (ABC) \perp (BCD)$
- saturated: no independence relationship

(b) Use the functions shown in the table together with `loglin2formula()` to print the corresponding model formulas for each.

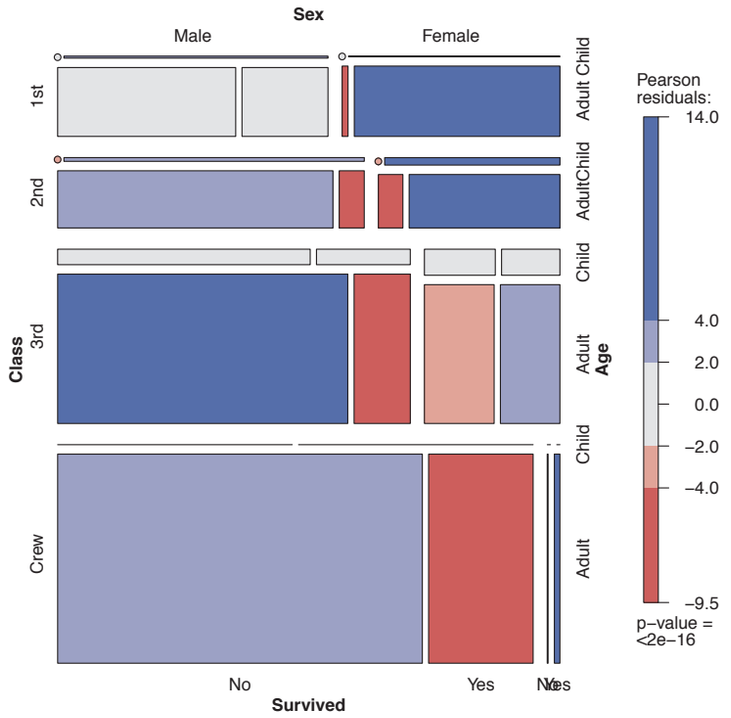
★ The model generating functions, `mutual()`, `joint()`, etc. provide a simple way to specify loglinear models for `loglm()` and `mosaic()`.

```
> loglin2formula(mutual(4, factors=LETTERS[1:4]))
~A + B + C + D
> loglin2formula(joint(4, factors=LETTERS[1:4]))
~A:B:C + D
> loglin2formula(joint(4, factors=LETTERS[1:4], with=1))
~B:C:D + A
> loglin2formula(conditional(4, factors=LETTERS[1:4]))
~A:D + B:D + C:D
> loglin2formula(conditional(4, factors=LETTERS[1:4], with=1))
~B:A + C:A + D:A
> loglin2formula(markov(4, factors=LETTERS[1:4]))
~A:B + B:C + C:D
> loglin2formula(markov(4, factors=LETTERS[1:4], order=2))
~A:B:C + B:C:D
> loglin2formula(saturated(4, factors=LETTERS[1:4]))
~A:B:C:D
```

Exercise 5.12 The dataset *Titanic* classifies the 2,201 pasengers and crew of the *Titanic* by Class (1st, 2nd, 3rd, Crew), Sex, Age, and Survived. Treating Survived as the response variable,

(a) Fit and display a mosaic plot for the baseline model of joint independence, [CGA][S]. Describe the remaining pattern of associations.
★

```
> # what is the formula for the joint independence here?
> form1 <- loglin2formula(joint(4, factors=names(dimnames(Titanic))))
> form1

~Class:Sex:Age + Survived

> library(MASS)
> mod1 <- loglm(formula=~Class:Sex:Age + Survived, data=Titanic)
> mod1

Call:
loglm(formula = ~Class:Sex:Age + Survived, data = Titanic)

Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 671.96 15        0
Pearson             NaN 15      NaN

> mosaic(Titanic, expected=~Class:Sex:Age + Survived, shade=TRUE)
```



(b) Do the same for a "main effects" model that allows two-way associations between each of C, G, and A with S.
★

(c) What three-way association term should be added to this model to allow for greater survival among women and children? Does this give an acceptable fit?
★

(d) Test and display models that allow additional three-way associations until you obtain a reasonable fit.
★

Chapter 6 Correspondence Analysis

These solutions use an updated version of the `ca` package, v. 0.64 or greater. In particular, MCA plots are now simpler, using a new `mcaplot()` function, presently in vcdExtra, and lines can be added to MCA plots using `multilines()`. Coordinates for CA/MCA plots can more readily be extracted using `cacoord()`.

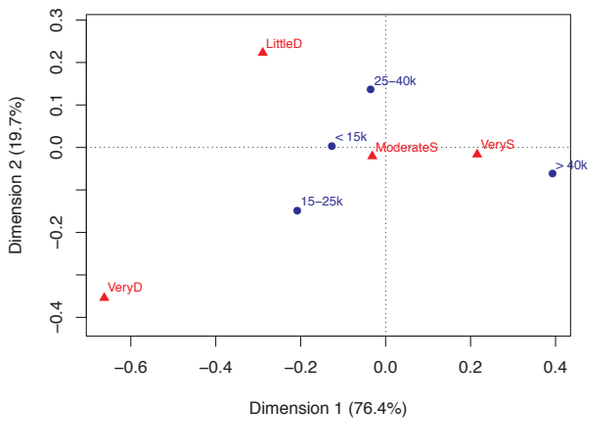
Exercise 6.1 The *JobSat* data in vcdExtra gives a 4×4 table recording job satisfaction in relation to income.

(a) Carry out a simple correspondence analysis on this table. How much of the inertia is accounted for by a one-dimensional solution? How much by a two-dimensional solution?
★ The 1D solution accounts for 76.4

```
> data("JobSat", package="vcdExtra")
> library(ca)
> jobsat.ca <- ca(JobSat)
>     # just show the scree plot
> summary(jobsat.ca, rows=FALSE, columns=FALSE)

Principal inertias (eigenvalues):

 dim    value      %    cum%   scree plot
 1      0.047496  76.4  76.4   ********************
 2      0.012248  19.7  96.1   *****
 3      0.002397   3.9 100.0   *
        -------- -----
 Total: 0.062141 100.0
```

(b) Plot the 2D CA solution. To what extent can you consider the association between job satisfaction and income "explained" by the ordinal nature of these variables?
★

```
> plot(jobsat.ca)
```



Job `satisfaction` is ordered as expected by its ordinal levels along Dimension 1. The levels of `income` in this plot do not appear to be ordered according to the quantitative levels they represent.

Exercise 6.2 Refer to Exercise 5.1 in Chapter 5. Carry out a simple correspondence analysis on the 4×5 table *criminal* from the logmult package.

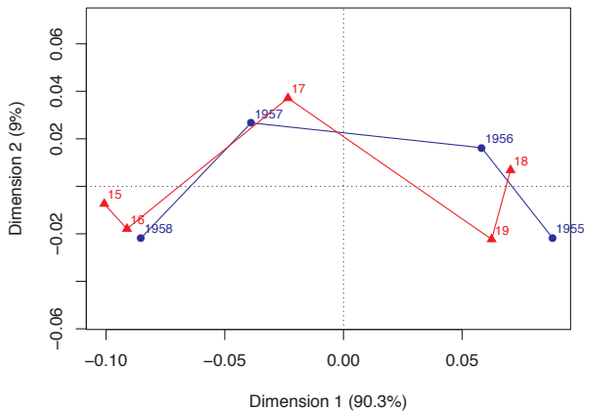
(a) What percentages of the Pearson χ^2 for association are explained by the various dimensions?
★

```
> data("criminal", package = "logmult")
> criminal.ca <- ca(criminal)
>     # just show the scree plot
> summary(criminal.ca, rows=FALSE, columns=FALSE)
```

```
Principal inertias (eigenvalues):

 dim    value      %   cum%   scree plot
 1      0.004939  90.3  90.3  ***********************
 2      0.000491   9.0  99.3  **
 3      3.8e-050   0.7 100.0
        -------- -----
 Total: 0.005468 100.0
```

(b) Plot the 2D correspondence analysis solution. Describe the pattern of association between year and age.
★

```
> plot(criminal.ca, lines=TRUE)
```



The category points for both year and age vary systematically over Dimension 1. There were more younger men in later years, and more older in earlier years.

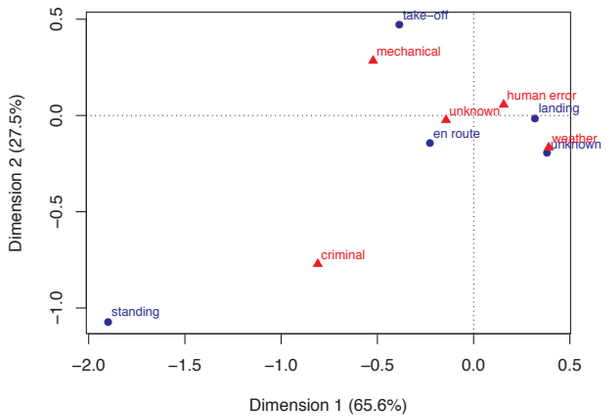
Exercise 6.3 Refer to Exercise 5.2 for a description of the *AirCrash* data from the vcdExtra package. Carry out a simple correspondence analysis on the 5×5 table of `Phase` of the flight and `Cause` of the crash.

(a) What percentages of the Pearson χ^2 for association are explained by the various dimensions?
★ `aircrash.tab` was calculated in Exercise 5.2.

```
> aircrash.tab

          Cause
Phase      criminal human error mechanical unknown weather
  en route       16          63         29      25      24
  landing         4         114         19      18      55
  standing        2           0          2       0       0
  take-off        1          29         24       8       3
  unknown         0           1          0       1       1

> aircrash.ca <- ca(aircrash.tab)
>      # just show the scree plot
> summary(aircrash.ca, rows=FALSE, columns=FALSE)

Principal inertias (eigenvalues):

 dim    value      %   cum%   scree plot
 1      0.123002  65.6  65.6  ****************
 2      0.051548  27.5  93.1  *******
 3      0.012340   6.6  99.7  **
 4      0.000562   0.3 100.0
        -------- -----
 Total: 0.187452 100.0
```

(b) Plot the 2D correspondence analysis solution. Describe the pattern of association between phase and cause. How would you interpret the dimensions?
★

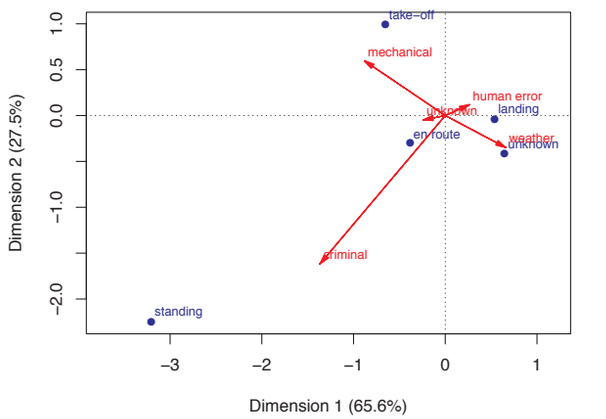
```
> plot(aircrash.ca)
```



Dimension 1 largely distinguishes standing from landing, where the main causes are human error and weather problems. Dimension 2 largely distinguishes standing from take-off, where mechanical problems are more prevalent.

(c) The default plot method uses `map="symmetric"` with points for both rows and columns. Try using `map="symbiplot"` with vectors (`arrows=`) for either rows or columns. (Read `help(plot.ca)` for a description of these options.)
★

```
> plot(aircrash.ca, map="symbiplot", arrows=c(FALSE, TRUE))
```



Exercise 6.4 The data set *caith* in MASS gives a classic table tabulating hair color and eye color of people in Caithness, Scotland, originally from Fisher (1940).

(a) Carry out a simple correspondence analysis on this table. How many dimensions seem necessary to account for most of the association in the table?
★

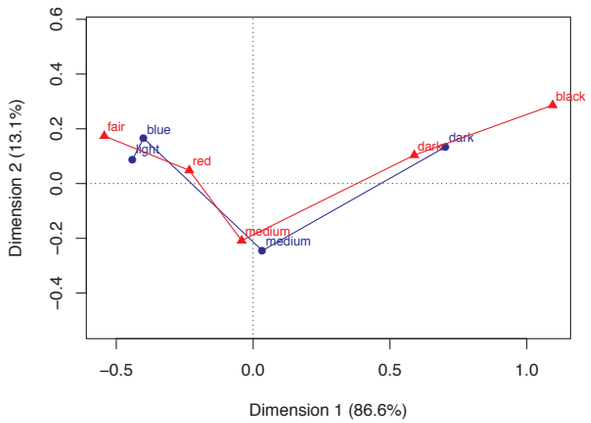
```
> data("caith", package="MASS")
> caith.ca <- ca(caith)
> summary(caith.ca, rows=FALSE, columns=FALSE)

Principal inertias (eigenvalues):
```

```
dim    value      %   cum%   scree plot
1      0.199245  86.6  86.6  **********************
2      0.030087  13.1  99.6  ***
3      0.000859   0.4 100.0
       -------- -----
Total: 0.230191 100.0
```

One could argue that a 1D solution is adequate here. The 2D solution is essentially complete.

(b) Plot the 2D solution. The interpretation of the first dimension should be obvious; is there any interpretation for the second dimension?
★

```
> plot(caith.ca, lines=TRUE)
```



Dimension 1 is the obvious light–dark dimension for hair color and eye color (but not that blue and light eyes have quite similar positions). Dimension 2 could be called "extremes vs. middle", but in CA results with largely 1D association, this effect is common, often called the "horseshoe" or "arch" effect.

Exercise 6.5 The same data, plus a similar table for Aberdeen, are given as a three-way table as *HairEyePlace* in vcdExtra.

(a) Carry out a similar correspondence analysis to the last exercise for the data from Aberdeen. Comment on any differences in the placement of the category points.
★

(b) Analyze the three-way table, stacked to code hair color and place interactively, i.e., for the loglinear model [Hair Place][Eye]. What does this show?
★

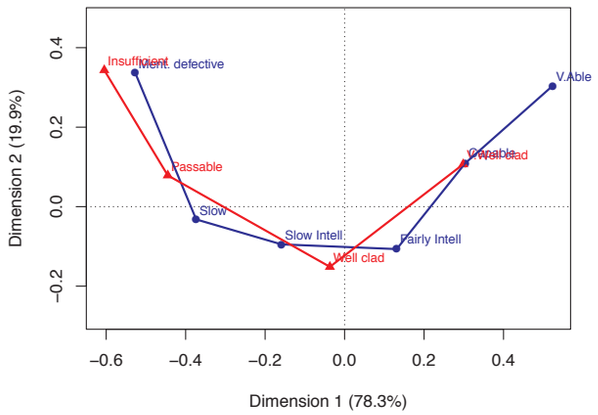
Exercise 6.6 The data set *Gilby* in vcdExtra gives a classic (but now politically incorrect) 6×4 table of English schoolboys classified according to their clothing and their teacher's rating of "dullness" (lack of intelligence).

(a) Compute and plot a correspondence analysis for this data. Write a brief description and interpretation of these results.
★

```
> data("Gilby", package="vcdExtra")
> gilby.ca <- ca(Gilby)
> summary(gilby.ca)

Principal inertias (eigenvalues):

 dim    value      %   cum%   scree plot
 1      0.079346  78.3  78.3  ********************
 2      0.020118  19.9  98.1  *****
 3      0.001881   1.9 100.0
        -------- -----
 Total: 0.101346 100.0
```

```
Rows:
    name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
1 | Mntd |   75  996  292 | -527 707 264 |  337 289 426 |
2 | Slow |  127  943  187 | -374 936 223 |  -32   7   7 |
3 | SlwI |  236  938   85 | -159 691  75 |  -95 247 106 |
4 | FrlI |  310  999   88 |  131 601  67 | -107 398 176 |
5 | Cpbl |  217 1000  223 |  304 885 253 |  109 115 129 |
6 | VAbl |   34  990  125 |  524 740 118 |  304 250 157 |

Columns:
    name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
1 | VWll |  369 1000  366 |  299 886 414 |  107 113 208 |
2 | Wllc |  435  989  106 |  -37  56   8 | -151 934 496 |
3 | Pssb |  154  976  317 | -445 946 383 |   79  30  47 |
4 | Insf |   42  955  212 | -605 722 195 |  344 233 248 |
```

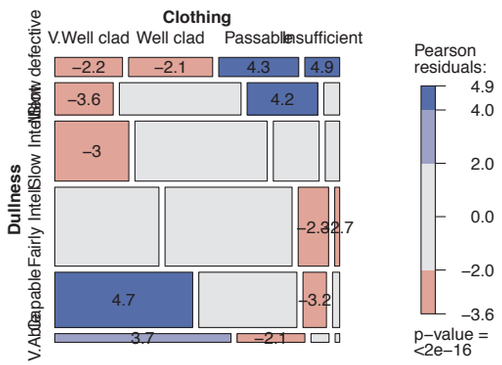
```
> plot(gilby.ca, lines="TRUE", lwd=2)
```



The association between clothing and dullness is largely one dimensional, but there is a moderately large horseshoe effect. On Dimension 1, the categories of both variables are approximately equally spaced.

(b) Make an analogous mosaic plot of this table. Interpret this in relation to the correspondence analysis plot. ★

```
> mosaic(Gilby, shade=TRUE, labeling=labeling_residuals)
```



The mosaic nearly shows the opposite corner pattern associated with a unidimensional association

of two ordered variables, but the largest residuals are not systematically confined to the diagonally opposite cells. The mosaic shows the marginal frequencies of dullness and the cell frequencies by the area of the tiles, while this information is not available in the CA plot.

Exercise 6.7 For the mental health data analyzed in Example 6.2, construct a shaded sieve diagram and mosaic plot. Compare these with the correspondence analysis plot shown in Figure 6.2. What features of the data and the association between SES and mental health status are shown in each?

★

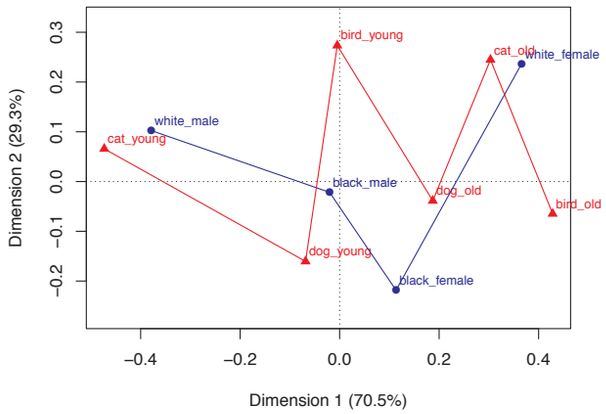
Exercise 6.8 Simulated data are often useful to help understand the connections between data, analysis methods, and associated graphic displays. Section 6.3.1 illustrated interactive coding in R, using a simulated 4-way table of counts of pets, classified by age, color, and sex, but with no associations because the counts had a constant Poisson mean, $\lambda = 15$.

(a) Re-do this example, but in the call to `rpois()`, specify a non-negative vector of Poisson means to create some associations among the table factors.
★ First, create a data.frame of the factor levels. Then you can use these variables to create the Poisson means in a way that varies across some of the category combinations to create associations. Here we try to creates associations between Pet and Color (more black dogs than other combinations) and between Age and Sex (more young male pets). These cell means are then used in the call to `rpois()`.

```
> dim <- c(3, 2, 2, 2)
> factors <- expand.grid(Pet=c("dog","cat","bird"),
+                        Age=c("young","old"),
+                        Color=c("black", "white"),
+                        Sex=c("male", "female"))
> means <-
+ with(factors, {
+        20 +
+        ifelse(Pet=="dog", 10, 0) +
+        ifelse((Pet=="dog" & Color=="black"), 10, -10) +
+        ifelse((Age=="young" & Sex=="male"), 5, -5)
+        })
>
> set.seed(1234)
> tab <- array(rpois(prod(dim), means), dim=dim)
> dimnames(tab) <- list(Pet=c("dog","cat","bird"),
+                        Age=c("young","old"),
+                        Color=c("black", "white"),
+                        Sex=c("male", "female"))
> # stack
> as.matrix(ftable(Pet + Age ~ Color + Sex, tab))

             Pet_Age
Color_Sex      dog_young dog_old cat_young cat_old bird_young bird_old
  black_male          36      30        16       6         16        5
  black_female        30      25         8       4          7        5
  white_male          22      13        18       3         12        1
  white_female        11      18         3       6         11        4
```

(b) Use CA methods to determine if and how the structure you created in the data appears in the results.
★

```
> library(ca)
> pets.ca <- ca(as.matrix(ftable(Pet + Age ~ Color + Sex, tab)))
> plot(pets.ca, lines=TRUE)
```



Exercise 6.9 The *TV* data was analyzed using CA in Example 6.4, ignoring the variable `Time`. Carry out analyses of the 3-way table, reducing the number of levels of `Time` to three hourly intervals as shown below.

```
> data("TV", package="vcdExtra")
> # reduce number of levels of Time
> TV.df <- as.data.frame.table(TV)
> levels(TV.df$Time) <- rep(c("8", "9", "10"), c(4, 4, 3))
> TV3 <- xtabs(Freq ~ Day + Time + Network, TV.df)
> structable(Day ~ Network + Time, TV3)

             Day Monday Tuesday Wednesday Thursday Friday
Network Time
ABC     8            536     861       744      735   1119
        9           1401    1205      1022      682    907
        10           910    1044       668      349    711
CBS     8           1167     646       550      680    509
        9            967     959       409      385    544
        10           789     798       324      270    426
NBC     8            858    1090       512     1927    823
        9            946     890       831     1858    590
        10           825     588       869     2101    585
```

(a) Use the stacking approach (Section 6.3) to perform a CA of the table with `Network` and `Time` coded interactively. You can create this using the `as.matrix()` method for a "structable" object.

```
> TV3S <- as.matrix(structable(Day ~ Network + Time, TV3), sep=":")
```

★

```
> TV3S.ca <-ca(TV3S)
> summary(TV3S.ca, rows=FALSE, columns=FALSE)

Principal inertias (eigenvalues):

 dim    value      %    cum%   scree plot
 1      0.089629  75.2   75.2  ******************
 2      0.018576  15.6   90.8  ****
 3      0.008992   7.5   98.3  **
 4      0.002001   1.7 100.0
        -------- -----
 Total: 0.119198 100.0
```

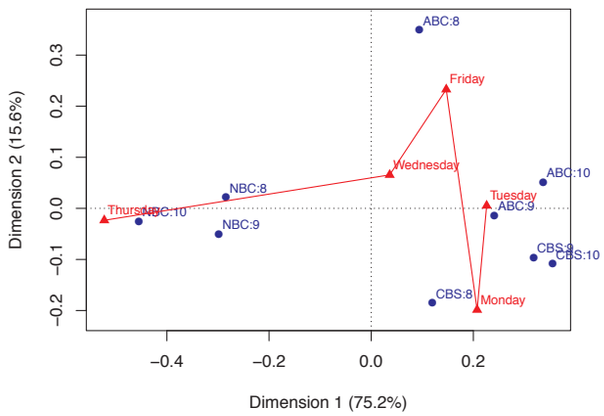
(b) What loglinear model is analyzed by this approach?

★ The model is the joint independence model, [Day][Network Time], asserting that the frequencies of watching the combinations of networks in different time slots do not vary with day of the week.

(c) Plot the 2D solution. Compare this to the CA plot of the two-way table in Figure 6.4.

★

```
> plot(TV3S.ca, lines=c(FALSE, TRUE))
```

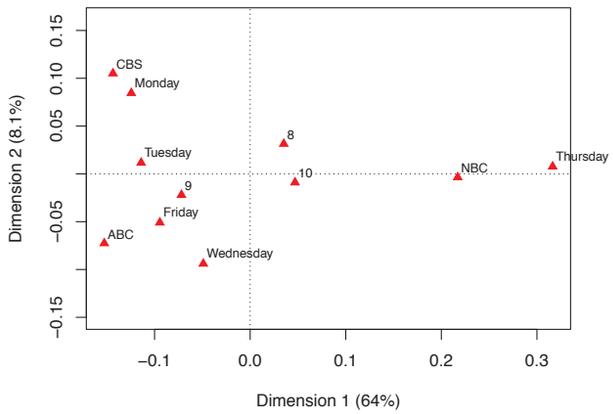


Overall, the plot is somewhat similar to that of Figure 6.4 except for the (arbitrary) reversal of Dimension 1. The category points for NBC cluster near Thursday, while those for ABC and CBS are at the other end of that dimension and Dimension 2 is related to the difference in viewership to those channels.

(d) Carry out an MCA analysis using `mjca()` of the three-way table `TV3`. Plot the 2D solution, and compare this with both the CA plot and the solution for the stacked three-way table.

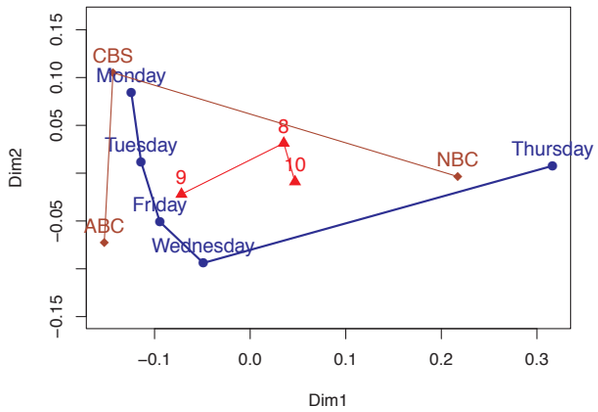
★ The MCA analysis differs in that it includes all pairwise associations of Day, Time and Network rather than just the joint independence model analyzed by the stacking approach.

```
> TV3.mca <- mjca(TV3)
> plot(TV3.mca, collabels="level")
```



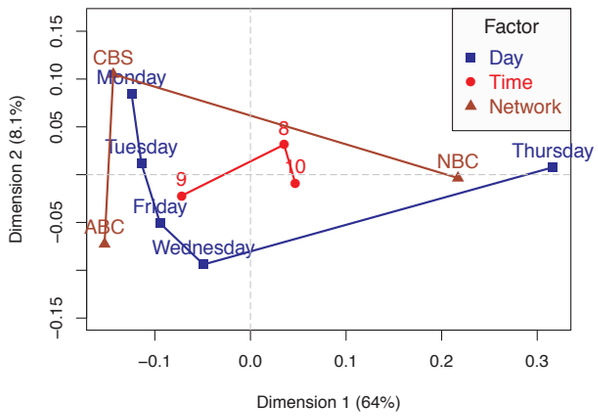
The default plot is somewhat difficult to read because the factor points s aren't distinguished by color or shape. A customized plot can be constructed as follows (as illustrated in the text, e.g., for Figure 6.10).

```
> res <- plot(TV3.mca, labels=0, pch='.', cex.lab=1.2)
> coords <- data.frame(res$cols, TV3.mca$factors)
> nlev <- rle(as.character(coords$factor))$lengths
> fact <- unique(as.character(coords$factor))
>
> cols <- c("blue", "red", "brown")
> lwd <- 2
> plot(Dim2 ~ Dim1, type='n', data=coords, asp=1)
> points(coords[,1:2], pch=rep(16:18, nlev), col=rep(cols, nlev), cex=1.2)
> text(coords[,1:2], labels=coords$level, col=rep(cols, nlev), pos=3, cex=1.2, xpd=TRUE)
> multilines(coords[, c("Dim1", "Dim2")], group=coords$factor, col=cols, lwd=lwd)
```



A similar plot can now be produced more simply using:

```
> mcaplot(TV3.mca, legend=TRUE)
```



Exercise 6.10 Refer to the MCA analysis of the *PreSex* data in Example 6.8. Use the stacking approach to analyze the stacked table with the combinations of premarital and extramarital sex in the rows and the combinations of gender and marital status in the columns. As suggested in the exercise above, you can use `as.matrix(structable())` to create the stacked table.

```
> presexS<- as.matrix(structable(PremaritalSex + ExtramaritalSex ~ Gender + Marital, PreSex), sep=":")
> presexS

                    PremaritalSex:ExtramaritalSex
Gender:MaritalStatus Yes:Yes Yes:No No:Yes No:No
      Women:Divorced      17     54     36   214
      Women:Married        4     25      4   322
      Men:Divorced        28     60     17    68
      Men:Married         11     42      4   130
```

(a) What loglinear model is analyzed by this approach? Which associations are included and which are excluded in this analysis?

★ The model is that of independence between the combinations of the row variables and the column variables, i.e., [Pre Extra][Gender Marital]. The associations that remain indicate how the combinations of pre- and extra-marital sex are related to the the combinations of gender and marital status. The

association between pre-marital sex and extra-marital sex is excluded, as is the association between gender and marital status.

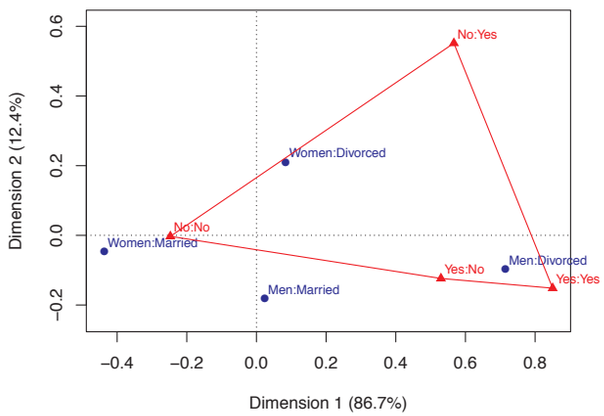
(b) Plot the 2D CA solution for this analysis. You might want to draw lines connecting some of the row points or column points to aid in interpretation.

★ The 2D solution accounts for 99.1% of association between these sets of variables.

```
> presexS.ca <- ca(presexS)
> summary(presexS.ca)

Principal inertias (eigenvalues):

 dim    value      %    cum%   scree plot
 1      0.153126  86.7  86.7   ***********************
 2      0.021929  12.4  99.1   ***
 3      0.001552   0.9 100.0
        -------- -----
 Total: 0.176606 100.0


Rows:
    name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
1 | WmnD |  310  995   90 |   84 138  14 |  210 857 622 |
2 | WmnM |  343  995  376 | -437 984 427 |  -46  11  34 |
3 | MnDv |  167  996  494 |  715 978 558 |  -97  18  72 |
4 | MnMr |  181  878   39 |   24  15   1 | -182 863 273 |

Columns:
    name   mass  qlt  inr    k=1  cor ctr    k=2 cor ctr
1 | YsYs |   58  979  250 |  850  948 274 | -152  30  61 |
2 | YesN |  175  989  296 |  530  937 320 | -124  51 122 |
3 | NoYs |   59 1000  209 |  567  514 124 |  552 486 817 |
4 | NoNo |  708 1000  245 | -247 1000 283 |   -3   0   0 |
```

Here is one version of a plot, drawing lines connecting the pairs of Gender and Marital status.

```
> res <- plot(presexS.ca)
> # join pairs of column points
> lines(res$cols[1:2,], col="red")
> lines(res$cols[3:4,], col="red")
> lines(res$cols[c(1,3),], col="red")
> lines(res$cols[c(2,4),], col="red")
```



(c) How does this analysis differ from the MCA analysis shown in Figure 6.10?

★ The MCA analysis treats all four factors individually, analyzing all bivariate associations. The stacked approach here treats them in two sets, analyzing only the associations *between* sets.

Exercise 6.11 Refer to Exercise 5.10 for a description of the `Vietnam` data set in vcdExtra.

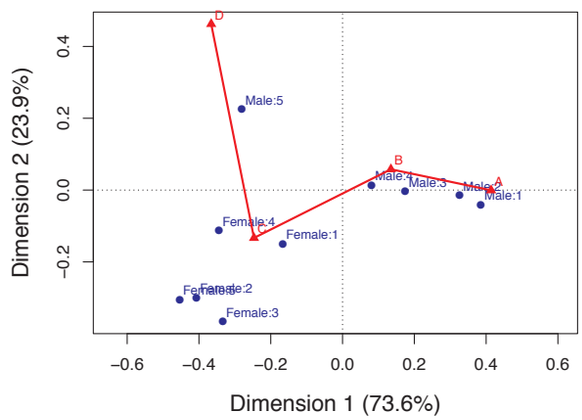
(a) Using the stacking approach, carry out a correspondence analysis corresponding to the loglinear model [R][YS], which asserts that the response is independent of the combinations of year and sex.

★ Two dimensions account for 97.5% of the association between response and the combinations of year and sex.

```
> data(Vietnam, package="vcdExtra")
> vietnam.tab <- xtabs(Freq ~ sex + year + response, data=Vietnam)
> vietnam.stacked <- as.matrix(structable(response ~ sex + year, vietnam.tab), sep=":")
>
> vietnam.ca <- ca(vietnam.stacked)
> summary(vietnam.ca, rows=FALSE, columns=FALSE)

Principal inertias (eigenvalues):

 dim    value      %    cum%   scree plot
 1      0.085680  73.6  73.6   ******************
 2      0.027881  23.9  97.5   ******
 3      0.002854   2.5 100.0   *
        -------- -----
 Total: 0.116415 100.0
```

(b) Construct an informative 2D plot of the solution, and interpret in terms of how the response varies with year for males and females.
★

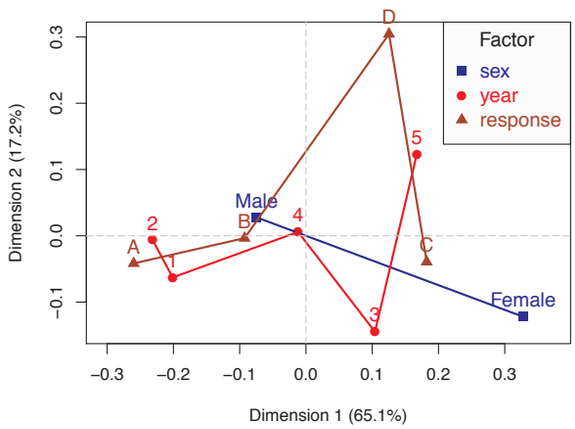
```
> plot(vietnam.ca, cex.lab=1.3, lines=c(FALSE, TRUE), lwd=2)
```



Dimension 1 corresponds to the ordering of the response categories, from the "dove" response D ("Withdraw military forces immediately") to the "hawk" response A ("Defeat North Vietnam by widespread bombing ..."). Males are ordered progressively from hawk to dove by their years in schools, with a big gap between year 4 and 5 (graduate students). Females cluster around response C ("De-escalate military activity ...") with no obvious trend over year in school.

(c) Use `mjca()` to carry out an MCA on the three-way table. Make a useful plot of the solution and interpret in terms of the relationship of the response to year and sex.
★ The MCA solution only accounts for 82.3% of the bivariate associations in 2D, and not much better in 3D.

```
> vietnam.mca <- mjca(vietnam.tab)
> summary(vietnam.mca, rows=FALSE, columns=FALSE)

Principal inertias (eigenvalues):

 dim    value      %    cum%   scree plot
 1      0.028219  65.1  65.1   ********************
 2      0.007445  17.2  82.3   *****
 3      0.000380   0.9  83.2
 4      1e-06000   0.0  83.2
        -------- -----
 Total: 0.043317

> mcaplot(vietnam.mca, legend=TRUE)
```



The plot of the MCA solution is somewhat difficult to interpret, because neither the response categories nor year are ordered as one would expect. This turns out to be an example where MCA is not that useful, because the association of response and year is different for males and females— i.e., there is a three-way association in this table. MCA, however, only accounts for pairwise associations among the table variables.

Exercise 6.12 Refer to Exercise 5.9 for a description of the *Accident* data set in vcdExtra. The data set is in the form of a frequency data frame, so first convert to table form.

```
> accident.tab <- xtabs(Freq ~ age + result + mode + gender, data=Accident)
```

(a) Use `mjca()` to carry out an MCA on the four-way table `accident.tab`. ★

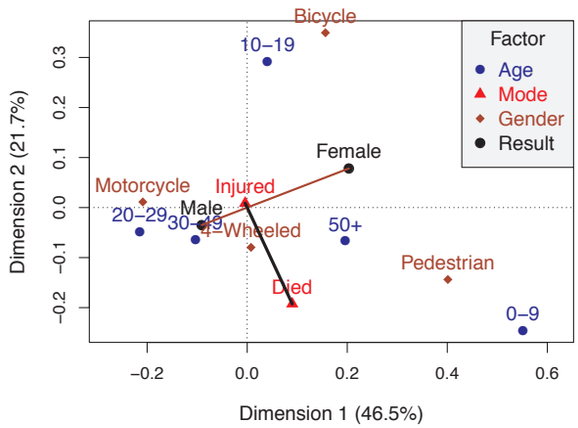
```
> accident.mca <- mjca(accident.tab)
> summary(accident.mca)

Principal inertias (eigenvalues):

 dim    value      %    cum%   scree plot
 1      0.025429  46.5  46.5   ****************
 2      0.011848  21.7  68.1   *******
 3      0.001889   3.5  71.6   *
 4      0.000491   0.9  72.5
        -------- -----
 Total: 0.054700


Columns:
                name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
1   |         age:0-9 |   13  672  107 |  551 561 152 | -246 111  65 |
2   |       age:10-19 |   49  678   91 |   40  13   3 |  292 665 354 |
3   |       age:20-29 |   56  784   85 | -215 747 102 |  -48  37  11 |
4   |       age:30-49 |   76  546   75 | -103 396  32 |  -63 149  26 |
5   |         age:50+ |   56  687   85 |  196 616  84 |  -67  72  21 |
6   |     result:Died |   11  515  100 |   90  92   3 | -192 422  34 |
7   |  result:Injured |  239  515    5 |   -4  92   0 |    9 422   2 |
8   |  mode:4-Wheeled |   81  230   73 |    8   2   0 |  -80 228  43 |
9   |    mode:Bicycle |   31  762   98 |  156 127  30 |  349 635 320 |
10  | mode:Motorcycle |   99  686   70 | -209 684 170 |   11   2   1 |
11  | mode:Pedestrian |   38  677  100 |  401 600 241 | -144  77  66 |
12  |   gender:Female |   77  788   77 |  203 686 126 |   78 101  40 |
13  |     gender:Male |  173  788   35 |  -91 686  56 |  -35 101  18 |
```

The adjusted inertias indicate that a 2D solution accounts for only 68.1% of the pairwise associations. The qualities (`qlt`) of the factor levels indicate that the categories are only moderately well-represented in a 2D plot.

(b) Construct an informative 2D plot of the solution, and interpret in terms of how the variable `result` varies in relation to the other factors.

★

```
> res <- plot(accident.mca, labels=0, pch='.', cex.lab=1.2)
> coords <- data.frame(res$cols, accident.mca$factors)
> cols <- c("blue", "red", "brown", "black")
> nlev <- accident.mca$levels.n
>
> points(coords[,1:2], pch=rep(16:19, nlev), col=rep(cols, nlev), cex=1.2)
> text(coords[,1:2], label=coords$level, col=rep(cols, nlev), pos=3, cex=1.2, xpd=TRUE)
> lines(Dim2 ~ Dim1, data=coords, subset=factor=="gender", lty=1, lwd=2, col="brown")
> lines(Dim2 ~ Dim1, data=coords, subset=factor=="result", lty=1, lwd=3, col="black")
>
> legend("topright", legend=c("Age", "Mode", "Gender", "Result"),
+        title="Factor", title.col="black",
+        col=cols, text.col=cols, pch=16:19,
+        bg="gray95", cex=1.2)
```



In the figure above, one interpretation of the dimensions is in terms of the age categories: Dimension 1 for young adults vs. old and young, Dimension 2 for teenage vs. the rest. In these terms, Dimension 1 shows associations among males, aged 20–49, riding a motorcycle or 4-wheeled vehicle and more likely to be injured, vs. females, either old or very young, as pedestrians and more likely to have died. Dimension 2 contrasts bicycle accidents involving youth aged 10–19 who are more likely to be just injured against the other categories.

Exercise 6.13 The *UCBAdmissions* data was featured in numerous examples in Chapter 4 (e.g., Example 4.11, Example 4.15) and Chapter 5 (e.g., Example 5.14, Example 5.18).

(a) Use `mjca()` to carry out an MCA on the three-way table `UCBAdmissions`.

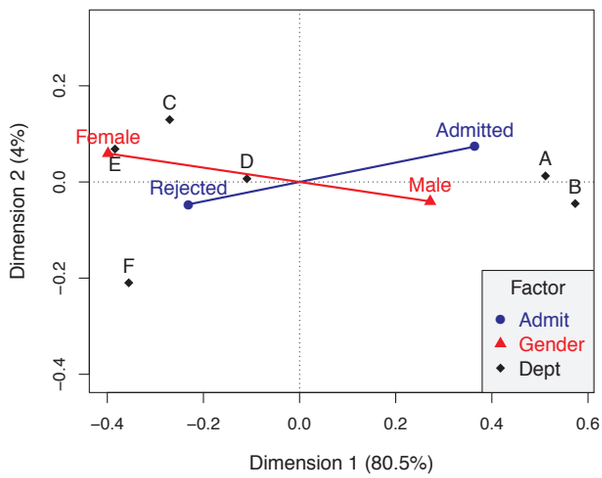
★

```
> ucb.mca <- mjca(UCBAdmissions)
> summary(ucb.mca)

Principal inertias (eigenvalues):

 dim    value      %    cum%   scree plot
 1      0.114945  80.5  80.5   ************************
 2      0.005694   4.0  84.5   *
 3      00000000   0.0  84.5
 4      00000000   0.0  84.5
 5      00000000   0.0  84.5
        -------- -----
 Total: 0.142840


Columns:
              name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
1   | Admit:Admitted |  129  911   93 |  365 875 150 |   74  36 123 |
2   | Admit:Rejected |  204  911   59 | -231 875  95 |  -47  36  78 |
3   |  Gender:Female |  135  863   95 | -399 845 187 |   59  19  84 |
4   |    Gender:Male |  198  863   65 |  272 845 127 |  -40  19  57 |
```

```
5  |         Dept:A |   69  838  117 |  512 837 156 |   13   1   2 |
6  |         Dept:B |   43  829  124 |  573 824 123 |  -45   5  15 |
7  |         Dept:C |   68  731  108 | -270 594  43 |  130 137 199 |
8  |         Dept:D |   58  832  106 | -110 828   6 |    7   3   0 |
9  |         Dept:E |   43  812  117 | -384 787  55 |   69  25  35 |
10 |         Dept:F |   53  737  116 | -355 547  58 | -210 190 406 |
```

(b) Plot the 2D MCA solution in a style similar to that shown in Figure 6.10 and Figure 6.11.
★

```
> op <- par(mar=c(5,4,1,1)+.1)
> res <- plot(ucb.mca, labels=0, pch='.', cex.lab=1.2)
>
> coords <- data.frame(res$cols, ucb.mca$factors)
> cols <- c("blue", "red", "black")
> nlev <- ucb.mca$levels.n
> pos <- rep(3, nrow(coords)); pos[9]<-1
>
> points(coords[,1:2], pch=rep(16:18, nlev), col=rep(cols, nlev), cex=1.2)
> text(coords[,1:2], labels=coords$level, col=rep(cols, nlev), pos=pos, cex=1.2, xpd=TRUE)
>
> lines(Dim2 ~ Dim1, data=coords, subset=factor=="Admit", lty=1, lwd=2, col=cols[1])
> lines(Dim2 ~ Dim1, data=coords, subset=factor=="Gender",  lty=1, lwd=2, col=cols[2])
>
> legend("bottomright", legend=c("Admit", "Gender", "Dept"),
+        title="Factor", title.col="black",
+        col=cols, text.col=cols, pch=16:18,
+        bg="gray95", cex=1.2)
> par(op)
```



(c) Interpret the plot. Is there some interpretation for the first dimension? What does the plot show about the relation of admission to the other factors?

★ The first dimension largely corresponds to Admission, showing the overall association of Males more likely to be admitted, Females more likely to be rejected. Note that the departments, labeled A–F, were actually ordered by overall rate of admission, but this ordering does not appear along Dimension 1 in the plot.

Chapter 7 Logistic Regression Models

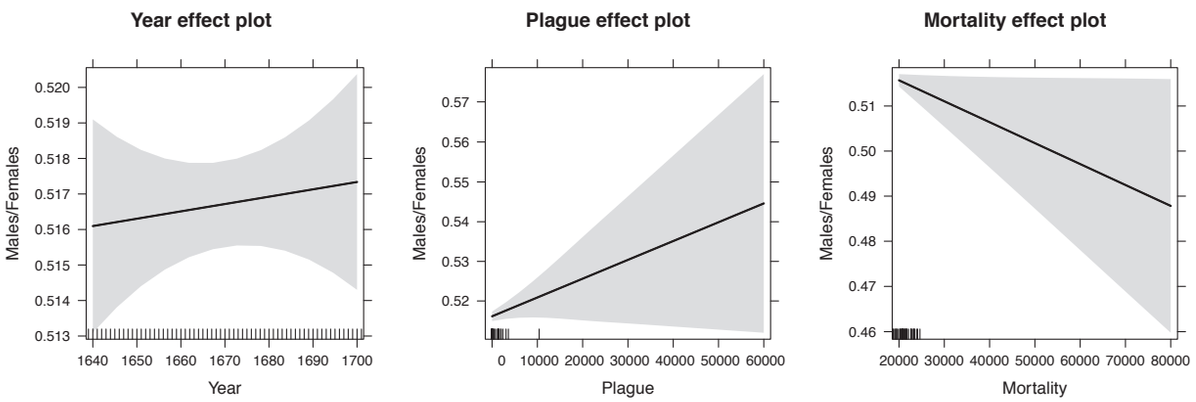
Exercise 7.1 Arbuthnot's data on the sex ratio of births in London was examined in Example 3.1. Use a binomial logistic regression model to assess whether the proportion of male births varied with the variables `Year`, `Plague`, and `Mortality` in the *Arbuthnot* data set. Produce effect plots for the terms in this model. What do you conclude?

★ For the binomial logistic model, use `cbind(Males, Females)` for the response variable in the model.

```
> data(Arbuthnot, package="HistData")
> arbuth.mod <- glm(cbind(Males, Females) ~ Year + Plague + Mortality,
+                   data=Arbuthnot, family=binomial)
> summary(arbuth.mod)

Call:
glm(formula = cbind(Males, Females) ~ Year + Plague + Mortality,
    family = binomial, data = Arbuthnot)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-3.184  -0.996  -0.005   0.850   3.714

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.07e-02   3.09e-01   -0.13    0.895
Year         8.28e-05   1.93e-04    0.43    0.668
Plague       1.91e-06   1.13e-06    1.68    0.093 .
Mortality   -1.86e-06   9.26e-07   -2.01    0.045 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 169.74  on 81  degrees of freedom
Residual deviance: 156.31  on 78  degrees of freedom
AIC: 963.8

Number of Fisher Scoring iterations: 3

> LRstats(arbuth.mod)

Likelihood summary table:
           AIC BIC LR Chisq Df Pr(>Chisq)
arbuth.mod 964 973      156 78    3.6e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The effects of `Year` and `Plague` are small and non-significant. The Male/Female proportion appears to decrease with increasing `Mortality`.

```
> library(effects)
> arbuth.eff <- allEffects(arbuth.mod)
> plot(arbuth.eff, ylab="Males/Females", rows=1, cols=3)
```



In the plots for `Plague` and `Mortality`, it is apparent that both are extremely skewed. One alternative is to represent these as `log(Plague+1)` and `log(Mortality)` in the model. Overall, these effects are quite small, but the main effects model `arbuth.mod` is better than the null model.

```
>   # compare with null model
> arbuth.mod0 <- glm(cbind(Males, Females) ~ 1,
+                    data=Arbuthnot, family=binomial)
> anova(arbuth.mod0, arbuth.mod, test="Chisq")

Analysis of Deviance Table

Model 1: cbind(Males, Females) ~ 1
Model 2: cbind(Males, Females) ~ Year + Plague + Mortality
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        81        170
2        78        156  3     13.4   0.0038 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exercise 7.2 For the Donner Party data in *Donner*, examine Grayson's 1990 claim that survival in the Donner Party was also mediated by the size of the family unit. This takes some care, because the `family` variable in the *Donner* data is a simplified grouping based on the person's name and known alliances among families from the historical record. Use the following code to compute a `family.size variable` from each individual's last name:

```
> data("Donner", package="vcdExtra")
> Donner$survived <-factor(Donner$survived, labels=c("no", "yes"))
> # use last name for family
> lname <-strsplit(rownames(Donner), ",")
> lname <-sapply(lname, function(x) x[[1]])
> Donner$family.size <-as.vector(table(lname)[lname])
```

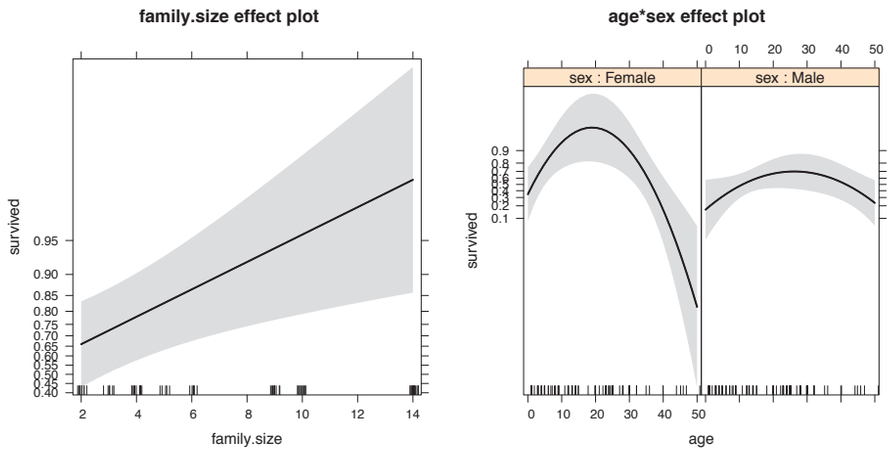
(a) Choose one of the models (`donner.mod4`, `donner.mod6`) from Example 7.9 that include the interaction of age and sex and nonlinear terms in age. Fit a new model that adds a main effect of `family.size`. What do you conclude about Grayson's claim?
★

```
> library(car)
> donner.mod4a <- glm(survived ~ poly(age,2) * sex + family.size,
+                     data=Donner, family=binomial)
> Anova(donner.mod4a)

Analysis of Deviance Table (Type II tests)

Response: survived
                 LR Chisq Df Pr(>Chisq)
poly(age, 2)        14.94  2    0.00057 ***
sex                  4.24  1    0.03944 *
family.size         11.54  1    0.00068 ***
poly(age, 2):sex    12.24  2    0.00220 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Family size seems to have an effect on survival.

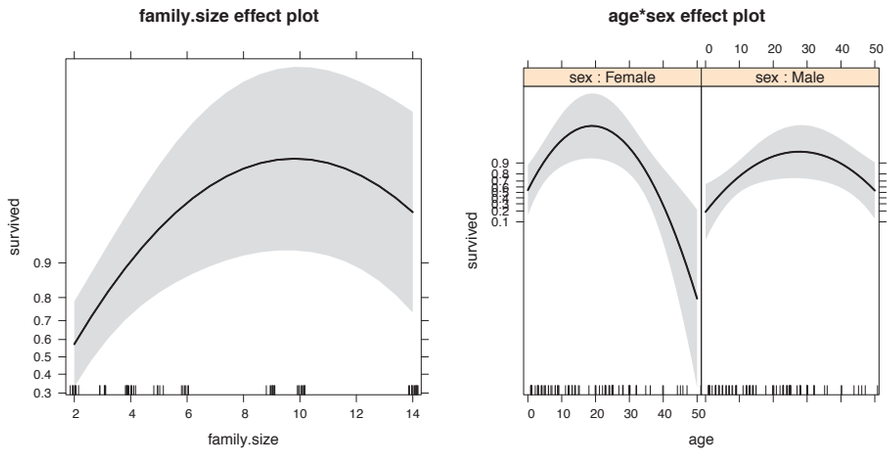
(b) Produce an effect plot for this model.
★

```
> library(effects)
> donner.eff4a <- allEffects(donner.mod4a, xlevels=list(age=seq(0,50,5)))
> plot(donner.eff4a, ticks=list(n=8))
```



(c) Continue, by examining whether the effect of family size can be taken as linear, or whether a nonlinear term should be added.
★

```
> donner.mod4b <- glm(survived ~ poly(age,2) * sex + poly(family.size, 2),
+                     data=Donner, family=binomial)
> Anova(donner.mod4b)

Analysis of Deviance Table (Type II tests)

Response: survived
                     LR Chisq Df Pr(>Chisq)
poly(age, 2)            17.64  2    0.00015 ***
sex                      2.33  1    0.12688
poly(family.size, 2)    24.81  2    4.1e-06 ***
poly(age, 2):sex        12.25  2    0.00219 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> donner.eff4b <- allEffects(donner.mod4b, xlevels=list(age=seq(0,50,5)))
> plot(donner.eff4b, ticks=list(n=8))
```



Exercise 7.3 Use component+residual plots (Section 7.5.3) to examine the additive model for the *ICU* data given by

```
> icu.glm2 <- glm(died ~ age + cancer  + admit + uncons,
+                 data=ICU, family=binomial)
```

(a) What do you conclude about the linearity of the (partial) relationship between age and death in this model?
★

(b) An alternative strategy is to allow some nonlinear relation for age in the model using a quadratic (or cubic) term like `poly(age, 2)` (or `poly(age, 3)`) in the model formula. Do these models provide evidence for a nonlinear effect of age on death in the ICU?

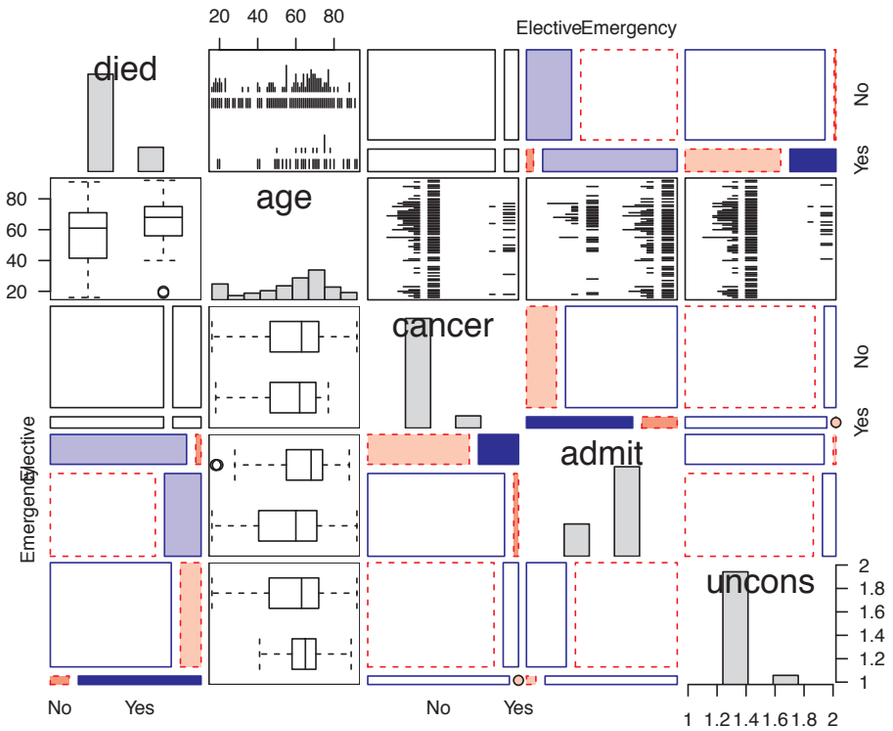
★

Exercise 7.4 Explore the use of other marginal and conditional plots to display the relationships among the variables predicting death in the ICU in the model `icu.glm2`. For example, you might begin with a marginal `gpairs()` plot showing all bivariate marginal relations, something like this:

```
> library(gpairs)
> gpairs(ICU[,c("died", "age", "cancer", "admit", "uncons")],
+   diag.pars=list(fontsize=16, hist.color="lightgray"),
+   mosaic.pars=list(gp=shading_Friendly,
+                    gp_args=list(interpolate=1:4)))
```

★ First, begin with the `gpairs()` plot:

```
> library(gpairs)
> gpairs(ICU[,c("died", "age", "cancer", "admit", "uncons")],
+   diag.pars=list(fontsize=16, hist.color="lightgray"),
+   mosaic.pars=list(gp=shading_Friendly,
+                    gp_args=list(interpolate=1:4)))
```



Exercise 7.5 The data set *Caesar* in vcdExtra gives a 3×2^3 frequency table classifying 251 women who gave birth by Caesarian section by `Infection` (three levels: none, Type 1, Type2) and `Risk`, whether `Antibiotics` were used, and whether the Caesarian section was `Planned` or not. `Infection` is a natural response variable. In this exercise, consider only the binary outcome of infection vs. no infection.

```
> data("Caesar", package="vcdExtra")
> Caesar.df <- as.data.frame(Caesar)
> Caesar.df$Infect <- as.numeric(Caesar.df$Infection %in%
+                                c("Type 1", "Type 2"))
```

(a) Fit the main-effects logit model for the binary response `Infect`. Note that with the data in the form of a frequency data frame you will need to use `weights=Freq` in the call to `glm()`. (It might also be convenient to reorder the levels of the factors so that `"No"` is the baseline level for each.)

★

```
>   # reorder levels
> Caesar.df$Infection <- factor(Caesar.df$Infection, levels=c("None", "Type 1", "Type 2"))
> Caesar.df$Risk <- factor(Caesar.df$Risk, levels=c("No", "Yes"))
> Caesar.df$Antibiotics <- factor(Caesar.df$Antibiotics, levels=c("No", "Yes"))
> Caesar.df$Planned <- factor(Caesar.df$Planned, levels=c("No", "Yes"))
>   # Logistic regression : None vs (Type 1, Type 2)
> caesar.glm <- glm(Infect ~ Risk + Antibiotics + Planned, weights = Freq,
+                   data=Caesar.df, family=binomial)
> caesar.glm

Call:  glm(formula = Infect ~ Risk + Antibiotics + Planned, family = binomial,
    data = Caesar.df, weights = Freq)

Coefficients:
   (Intercept)         RiskYes  AntibioticsYes      PlannedYes
        -0.793           1.827          -3.001          -0.906

Degrees of Freedom: 16 Total (i.e. Null);  13 Residual
Null Deviance:      301
Residual Deviance: 236  AIC: 244
```

(b) Use `summary()` or car (Fox and Weisberg, 2015)::`Anova()` to test the terms in this model.
★ By both the Wald tests from `summary()` and the Type II LR tests from `car::Anova` all three factors have significant effects on the probability of infection.

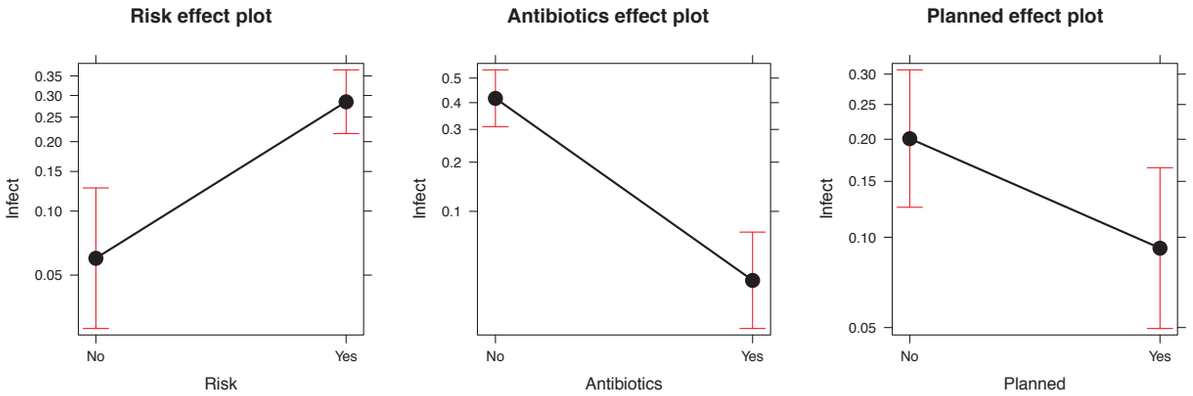
```
> library(car)
> summary(caesar.glm)

Call:
glm(formula = Infect ~ Risk + Antibiotics + Planned, family = binomial,
    data = Caesar.df, weights = Freq)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-6.747  -0.443   0.000   3.234   5.420

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)     -0.793      0.478   -1.66    0.097 .
RiskYes          1.827      0.436    4.19  2.8e-05 ***
AntibioticsYes  -3.001      0.459   -6.53  6.4e-11 ***
PlannedYes      -0.906      0.408   -2.22    0.026 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 300.85  on 16  degrees of freedom
Residual deviance: 236.36  on 13  degrees of freedom
AIC: 244.4

Number of Fisher Scoring iterations: 6

> Anova(caesar.glm)

Analysis of Deviance Table (Type II tests)

Response: Infect
           LR Chisq Df Pr(>Chisq)
Risk           20.6  1    5.8e-06 ***
Antibiotics    56.5  1    5.7e-14 ***
Planned         5.2  1      0.022 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(c) Interpret the coefficients in the fitted model in terms of their effect on the odds of infection.
★ From the coefficients in the model given above, Risk factors increase the log odds of infection by 1.83; treatment with Antibiotics decreases the log odds by 3.0; a planned C-section descreases the log odds of infection by 0.91.
These effects can perhaps be more easily interpreted in terms of the odds ratios calculated below, that give the multiple of the odds for the Yes group compared to the No group. For example, Risk can be said to multiply the odds by 6.22; Antibiotics multiplies the odds by 0.05, or a decrease of 95%; a planned C-section multiplies the odds of infection by 0.40, a decrease of 60%.

```
> exp(cbind(OddsRatio=coef(caesar.glm),
+           confint(caesar.glm)))

                OddsRatio    2.5 %   97.5 %
(Intercept)      0.452263 0.170362  1.13401
RiskYes          6.215158 2.736586 15.35990
AntibioticsYes   0.049734 0.019257  0.11742
PlannedYes       0.403978 0.176257  0.88092
```

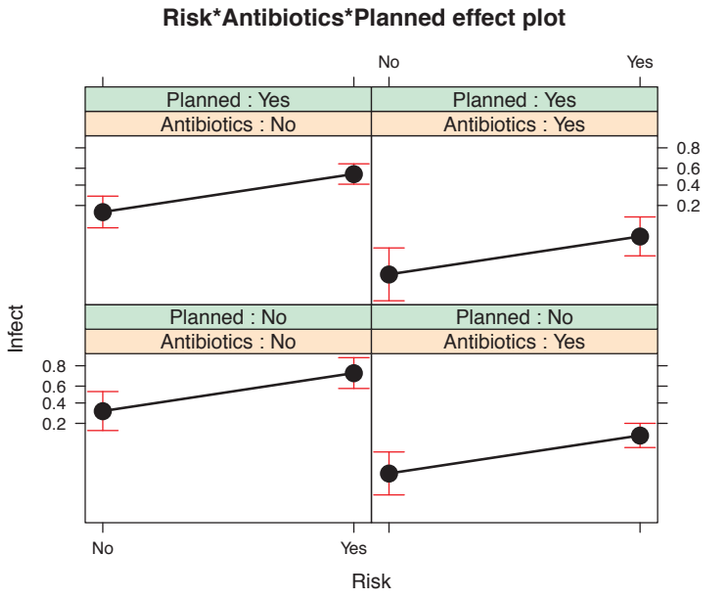
(d) Make one or more effects plots for this model, showing separate terms, or their combinations.
★ Two kinds of effect plots are useful here: `allEffects()`, that gives plots for each of the (main effect) terms in the model, and a full-model plot, showing predicted log odds of infection for all combinations of Risk, Antibiotics and Planned.

```
> library(effects)
> plot(allEffects(caesar.glm), rows=1, cols=3)
```



Full-model plot:

```
> plot(Effect(c("Risk", "Antibiotics", "Planned"), caesar.glm), layout=c(2,2))
```



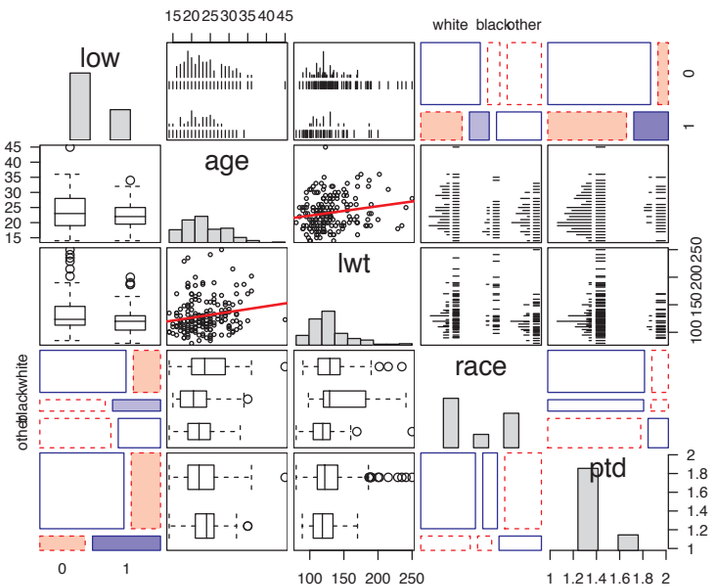
Exercise 7.6 The data set *birthwt* in the MASS package gives data on 189 babies born at Baystate Medical Center, Springfield, MA during 1986. The quantitative response is `bwt` (birth weight in grams), and this is also recorded as `low`, a binary variable corresponding to `bwt < 2500` (2.5 Kg). The goal is to study how this varies with the available predictor variables. The variables are all recorded as numeric, so in R it may be helpful to convert some of these into factors and possibly collapse some low frequency categories. The code below is just an example of how you might do this for some variables.

```
> data("birthwt", package="MASS")
> birthwt <- within(birthwt, {
+   low <- factor(low)
+   race <- factor(race, labels = c("white", "black", "other"))
+   ptd <- factor(ptl > 0)   # premature labors
+   ftv <- factor(ftv)       # physician visits
+   levels(ftv)[-(1:2)] <- "2+"
+   smoke <- factor(smoke>0)
+   ht <- factor(ht>0)
+   ui <- factor(ui>0)
+   })
```

(a) Make some exploratory plots showing how low birth weight varies with each of the available predictors. In some cases, it will probably be helpful to add some sort of smoothed summary curves or lines.

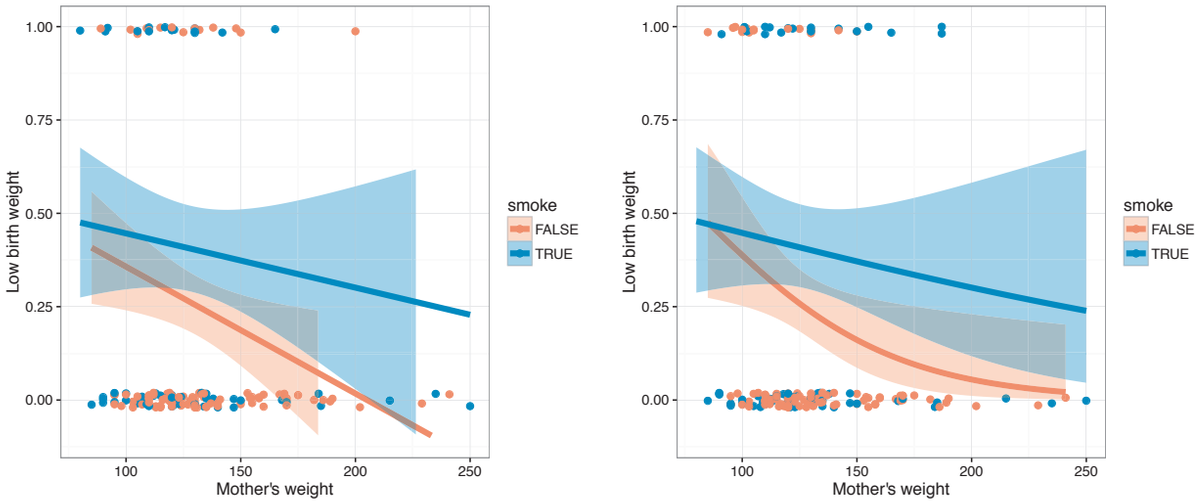
★ There are a wide variety of plots one could make for this data set. It is not unreasonable to start with a `gpairs()` plot for an overview. The first row and column shows the relations of low birth weight to the predictors. From this we can see that low birth weight (`low==1`) is associated with lower age, lower mother's weight (`lwt`), `race=="black"`, and previous premature labors (`ptd`).

```
> library(gpairs)
> vars <- c("low", "age", "lwt", "race", "ptd")
> gpairs(birthwt[,vars],
+   diag.pars=list(fontsize=16, hist.color="lightgray"),
+   lower.pars = list(scatter="lm"),
+   upper.pars = list(scatter="lm"),
+   mosaic.pars=list(gp=shading_Friendly,
+                    gp_args=list(interpolate=1:4)))
```



Other plots can explore particular relationships with low birth weight more thoroughly. Here we just show some example plots using ggplot2 (Wickham and Chang, 2015) for the relationship between `low` and mother's weight (`lwt`) conditioned by smoking status.

```
> library(ggplot2)
> gg <- ggplot(birthwt, aes(x=lwt, y=as.numeric(low)-1, color=smoke)) +
+         geom_point(position=position_jitter(height=0.05, width=0), size=2) +
+         ylim(-.1, 1) + theme_bw() +
+         xlab("Mother's weight") + ylab("Low birth weight")
>
> # use lm smoother
>         gg + geom_smooth(method="lm", aes(fill=smoke), alpha=0.3, size=2)
> # use glm smoother
>         gg + geom_smooth(method="glm", method.args=list(family = binomial),
+                          aes(fill=smoke), alpha=0.3, size=2)
```



(b) Fit several logistic regression models predicting low birth weight from these predictors, with the goal of explaining this phenomenon adequately, yet simply.
★ Here, we just start with the main effects model, then eliminate non-significant terms. A more general analysis could use MASS::`stepAIC()`, test for non-linear relations, and the presence of interaction effects.

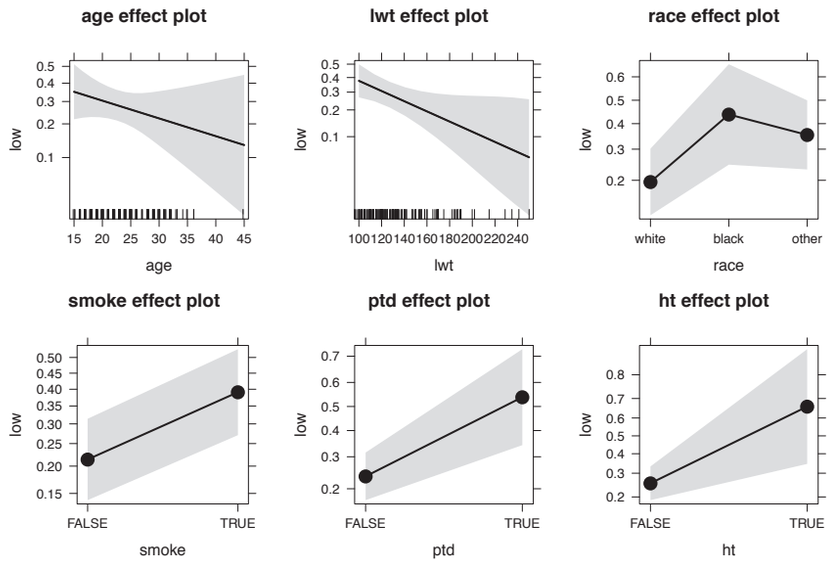
```
> # quick check on important effects
> bwt.mod0 <- glm(low ~ age + lwt + race + smoke + ptd + ht + ui + ftv,
+                 data=birthwt, family = binomial)
> summary(bwt.mod0)

Call:
glm(formula = low ~ age + lwt + race + smoke + ptd + ht + ui +
    ftv, family = binomial, data = birthwt)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.704  -0.807  -0.501   0.884   2.215

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.82302    1.24471    0.66   0.5085
age         -0.03723    0.03870   -0.96   0.3360
lwt         -0.01565    0.00708   -2.21   0.0271 *
raceblack    1.19241    0.53596    2.22   0.0261 *
raceother    0.74068    0.46174    1.60   0.1087
smokeTRUE    0.75553    0.42502    1.78   0.0755 .
ptdTRUE      1.34376    0.48062    2.80   0.0052 **
htTRUE       1.91317    0.72074    2.65   0.0079 **
uiTRUE       0.68020    0.46434    1.46   0.1430
ftv1        -0.43638    0.47939   -0.91   0.3627
ftv2+        0.17901    0.45638    0.39   0.6949
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 195.48  on 178  degrees of freedom
AIC: 217.5

Number of Fisher Scoring iterations: 4

> car::Anova(bwt.mod0)

Analysis of Deviance Table (Type II tests)

Response: low
      LR Chisq Df Pr(>Chisq)
age       0.94  1     0.3318
lwt       5.47  1     0.0193 *
race      5.75  2     0.0564 .
smoke     3.20  1     0.0737 .
ptd       8.11  1     0.0044 **
ht        7.46  1     0.0063 **
```

```
ui          2.11  1     0.1463
ftv         1.36  2     0.5071
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
>   # remove NS terms
> bwt.mod1 <- update(bwt.mod0, . ~ . - ui - ftv)
> Anova(bwt.mod1)

Analysis of Deviance Table (Type II tests)

Response: low
      LR Chisq Df Pr(>Chisq)
age       1.33  1     0.2487
lwt       5.40  1     0.0201 *
race      6.24  2     0.0441 *
smoke     4.62  1     0.0316 *
ptd       8.73  1     0.0031 **
ht        6.46  1     0.0111 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(bwt.mod1, bwt.mod0, test="Chisq")

Analysis of Deviance Table

Model 1: low ~ age + lwt + race + smoke + ptd + ht
Model 2: low ~ age + lwt + race + smoke + ptd + ht + ui + ftv
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       181        199
2       178        196  3     3.68      0.3
```

(c) Use some graphical displays to convey your findings.
★

```
> plot(allEffects(bwt.mod1), ci.style="bands")
```



Exercise 7.7 Refer to Exercise 5.9 for a description of the *Accident* data. The interest here is to model the probability that an accident resulted in death rather than injury from the predictors `age`, `mode`, and `gender`. With `glm()`, and the data in the form of a frequency table, you can use the argument `weight=Freq` to take cell frequency into account.

(a) Fit the main effects model, `result=="Died" ~ age + mode + gender`. Use car::`Anova()` to assess the model terms.
★ For analysis using `glm()` it is useful to consider the ordering of factors for the interpretation of coefficients, contrasts and plots. In the *Accident* data, `mode` is an unordered factor, but it is probably more useful to reorder the levels to make Pedestrian the reference level.

```
> levels(Accident$mode) <- c("Pedestrian", "Bicycle", "4-Wheeled", "Motorcycle" )
> acc.mod1 <- glm(result=="Died" ~ age + mode + gender,
+                 data=Accident, weight=Freq, family=binomial)
> Anova(acc.mod1)

Analysis of Deviance Table (Type II tests)

Response: result == "Died"
       LR Chisq Df Pr(>Chisq)
age        1179  4     <2e-16 ***
mode        137  3     <2e-16 ***
gender      468  1     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(b) Fit the model that allows all two-way interactions. Use `anova()` to test whether this model is significantly better than the main effects model.

★ In the all two-way model, all terms are significant and the addition of these terms is a significant improvement over the main effects model.

```
> acc.mod2 <- update(acc.mod1, . ~ .^2)
> Anova(acc.mod2)

Analysis of Deviance Table (Type II tests)

Response: result == "Died"
             LR Chisq Df Pr(>Chisq)
age              1101  4    < 2e-16 ***
mode              136  3    < 2e-16 ***
gender            419  1    < 2e-16 ***
age:mode          122 12    < 2e-16 ***
age:gender         47  4    1.6e-09 ***
mode:gender        22  3    6.7e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(acc.mod1, acc.mod2, test="Chisq")

Analysis of Deviance Table

Model 1: result == "Died" ~ age + mode + gender
Model 2: result == "Died" ~ age + mode + gender + age:mode + age:gender +
    mode:gender
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        71      64599
2        52      64384 19      214   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(c) Fit the model that also allows the three-way interaction of all factors. Does this offer any improvement over the two-way model?

★ In the three-way model, the term `age:mode:gender` is not significant, and `anova()` shows that it does not improve the two-way model. Using `LRstats()`, the two-way model has the best AIC and BIC.

```
> acc.mod3 <- update(acc.mod1, . ~ .^3)
> Anova(acc.mod3)

Analysis of Deviance Table (Type II tests)

Response: result == "Died"
                LR Chisq Df Pr(>Chisq)
age                 1101  4    < 2e-16 ***
mode                 136  3    < 2e-16 ***
gender               419  1    < 2e-16 ***
age:mode             122 12    < 2e-16 ***
age:gender            47  4    1.6e-09 ***
mode:gender           22  3    6.7e-05 ***
age:mode:gender       13 12       0.37
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(acc.mod1, acc.mod2, acc.mod3, test="Chisq")

Analysis of Deviance Table

Model 1: result == "Died" ~ age + mode + gender
Model 2: result == "Died" ~ age + mode + gender + age:mode + age:gender +
```

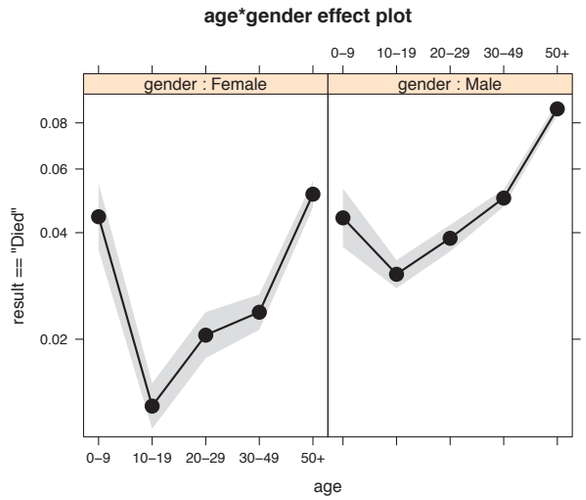
```
    mode:gender
Model 3: result == "Died" ~ age + mode + gender + age:mode + age:gender +
    mode:gender + age:mode:gender
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        71      64599
2        52      64384 19      214   <2e-16 ***
3        40      64371 12       13     0.37
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> LRstats(acc.mod1, acc.mod2, acc.mod3)

Likelihood summary table:
          AIC   BIC LR Chisq Df Pr(>Chisq)
acc.mod1 64617 64638    64599 71     <2e-16 ***
acc.mod2 64440 64507    64384 52     <2e-16 ***
acc.mod3 64451 64547    64371 40     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(d) Interpret the results of the analysis using effect plots for the two-way model, separately for each of the model terms. Describe verbally the nature of the `age*gender` effect. Which mode of transportation leads to greatest risk of death?

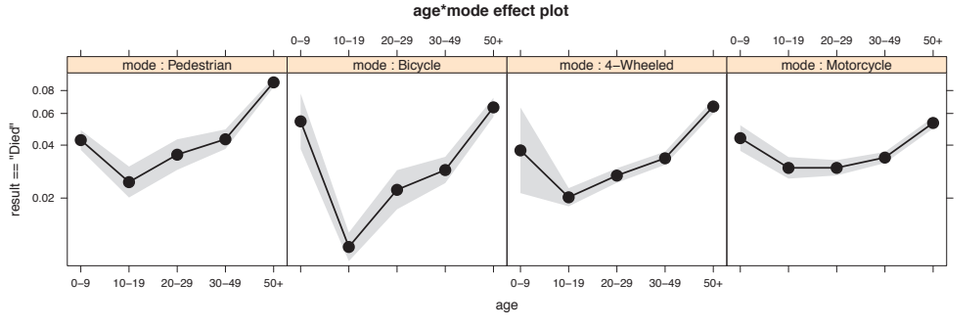
★ Controlling for mode, males and females have similar profiles, with the lowest probability of death in the 10–19 group, and highest in the 50+ age group.

```
> plot(Effect(c("age", "gender"), acc.mod2), ci.style="bands")
```

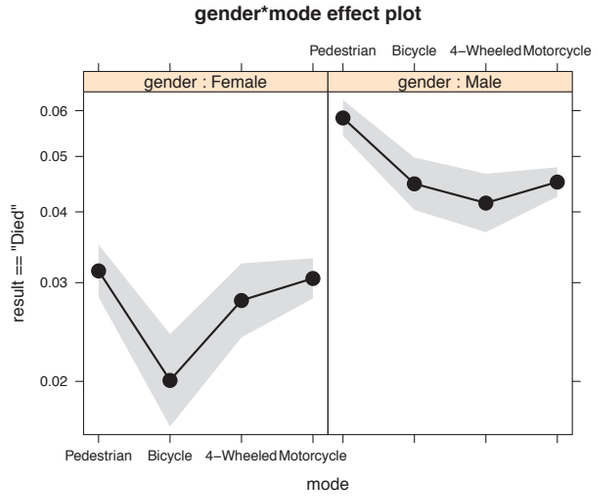


There are roughly similar profiles across age, with bicycle accidents showing the greatest differences among age groups.

```
> plot(Effect(c("age", "mode"), acc.mod2), ci.style="bands", layout=c(4,1))
```

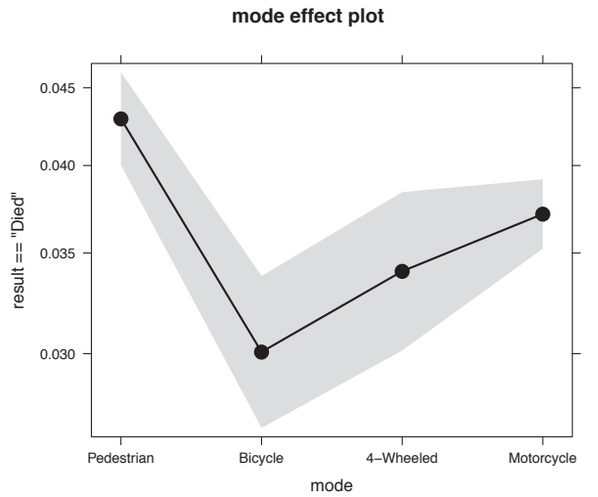



```
> plot(Effect(c("gender", "mode"), acc.mod2), ci.style="bands")
```



Overall, accidents involving motorcycles have the greatest probability of death, controlling for all other factors.

```
> plot(Effect("mode", acc.mod2), ci.style="bands")
```



Chapter 8 Models for Polytomous Responses

Exercise 8.1 For the women's labor force participation data (*Womenlf*), the response variable, partic, can be treated as ordinal by using

```
> Womenlf$partic <- ordered(Womenlf$partic,
+ levels=c('not.work', 'parttime', 'fulltime'))
```

Use the methods in Section 8.1 to test whether the proportional odds model holds for these data.

★ We start fitting the proportional odds model using polr():

```
> library(car)
> library(MASS)
> Womenlf$partic <- ordered(Womenlf$partic,
+                           levels = c("not.work", "parttime", "fulltime"))
> Wlf.polr <- polr(partic ~ hincome + children + region, data = Womenlf)
> Anova(Wlf.polr)

Analysis of Deviance Table (Type II tests)

Response: partic
         LR Chisq Df Pr(>Chisq)
hincome       8.5  1     0.0035 **
children     50.3  1    1.3e-12 ***
region        1.9  4     0.7547
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

to see that region has no significant effect on the comparison of the different response categories; we will ignore it in the further analysis. Now, we try comparing the proportional and non-proportional odds models by refitting them using vglm():

```
> library(VGAM)
> Wlf.po <- vglm(partic ~ hincome + children, data = Womenlf,
+                family = cumulative(parallel = TRUE))
>
> Wlf.npo <- vglm(partic ~ hincome + children, data = Womenlf,
+                 family = cumulative(parallel = FALSE))
> Wlf.ppo <- vglm(partic ~ hincome + children, data = Womenlf,
+                 family = cumulative(parallel = FALSE ~ hincome))
```

The warnings indicate that the non-proportional odds model does not converge, leaving the object returned by vglm() without log-likelihood, and making impossible the comparison using lrtest():

```
> logLik(Wlf.npo)

[1] NaN

> VGAM::lrtest(Wlf.po, Wlf.npo)

Likelihood ratio test

Model 1: partic ~ hincome + children
Model 2: partic ~ hincome + children
  #Df LogLik Df Chisq Pr(>Chisq)
1 522   -221
2 520        -2
```

It is, however, possible to fit a partial proportional odds model, relaxing the assumption for hincome:

```
> Wlf.ppo <- vglm(partic ~ hincome + children, data = Womenlf,
+                 family = cumulative(parallel = FALSE ~ hincome))
```

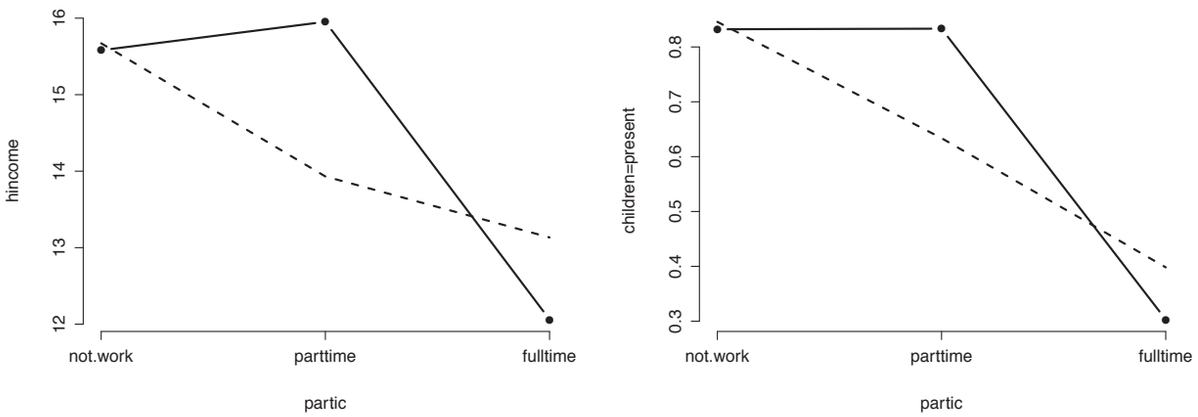
The comparison with lrtest() now yields:

```
> VGAM::lrtest(Wlf.po, Wlf.ppo)
```

```
Likelihood ratio test

Model 1: partic ~ hincome + children
Model 2: partic ~ hincome + children
  #Df LogLik Df Chisq Pr(>Chisq)
1 522   -221
2 521   -219 -1  4.14      0.042 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

showing that the partial proportional odds model is preferable to the proportional odds model. The graphical assessment using the method described in Section 8.1.4 provides further insights:

```
> library(rms)
> Wlf.po2 <- lrm(partic ~ hincome + children, data = Womenlf)
> Wlf.po2

Logistic Regression Model

lrm(formula = partic ~ hincome + children, data = Womenlf)
                      Model Likelihood     Discrimination    Rank Discrim.
                         Ratio Test           Indexes          Indexes
Obs           263    LR chi2      58.83    R2       0.236    C       0.723
 not.work     155    d.f.             2    g        1.083    Dxy     0.446
 parttime      42    Pr(> chi2) <0.0001    gr       2.954    gamma   0.461
 fulltime      66                          gp       0.236    tau-a   0.253
max |deriv| 4e-08                          Brier    0.209

                  Coef    S.E.   Wald Z Pr(>|Z|)
y>=parttime       1.8520 0.3863  4.79  <0.0001
y>=fulltime       0.9409 0.3699  2.54  0.0110
hincome          -0.0539 0.0195 -2.77  0.0057
children=present -1.9720 0.2869 -6.87  <0.0001

> plot.xmean.ordinaly(partic ~ hincome + children, data = Womenlf,
+                     lwd=2, pch=16, subn=FALSE)
```



The mean values for `hincome` do indeed not follow the fitted ones under the proportional odds model. Also, the first two categories, `not.work` and `parttime`, are not well distinguished.

Exercise 8.2 The data set *housing* in the MASS package gives a $3 \times 3 \times 4 \times 2$ table in frequency form relating (a) satisfaction (`Sat`) of residents with their housing (High, Medium, Low), (b) perceived degree of influence (`Infl`) they have on the management of the property (High, Medium, Low), (c) `Type` of rental (Tower, Atrium, Apartment, Terrace), and (d) contact (`Cont`) residents have with other residents (Low, High). Consider satisfaction as the ordinal response variable.

(a) Fit the proportional odds model with additive (main) effects of housing type, influence in management, and contact with neighbors to this data. (Hint: Using `polr()`, with the data in frequency form, you need to use the `weights` argument to supply the `Freq` variable.)

★ All three factors have significant effects on satisfaction in this simple model.

```
> data("housing", package="MASS")
> str(housing)

'data.frame': 72 obs. of  5 variables:
 $ Sat : Ord.factor w/ 3 levels "Low"<"Medium"<..: 1 2 3 1 2 3 1 2 3 1 ...
 $ Infl: Factor w/ 3 levels "Low","Medium",..: 1 1 1 2 2 2 3 3 3 1 ...
 $ Type: Factor w/ 4 levels "Tower","Apartment",..: 1 1 1 1 1 1 1 1 1 2 ...
 $ Cont: Factor w/ 2 levels "Low","High": 1 1 1 1 1 1 1 1 1 1 ...
 $ Freq: int  21 21 28 34 22 36 10 11 36 61 ...

> # proportional odds model
> library(MASS)
> PO.mod <- polr(Sat ~ Infl + Type + Cont, data=housing, weights=Freq)
> car::Anova(PO.mod)

Analysis of Deviance Table (Type II tests)

Response: Sat
     LR Chisq Df Pr(>Chisq)
Infl    108.2  2    < 2e-16 ***
Type     55.9  3    4.4e-12 ***
Cont     14.3  1    0.00016 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Although not asked in the question, it is important to assess whether the proportional odds assumption hold statistically (using VGAM), following the methods described in Section 8.1.3, or graphically (using rms (Harrell, Jr., 2015)), following Section 8.1.4. Here, we illustrate the statistical test using `vglm()`.

```
> library(VGAM)
>    # PO model
> PO.vglm <- vglm(Sat ~ Infl + Type + Cont, data=housing,
+                 family=cumulative(parallel=TRUE), weights=Freq)
>    # NPO model
> NPO.vglm <- vglm(Sat ~ Infl + Type + Cont, data=housing,
+                  family=cumulative(parallel=FALSE), weights=Freq)
>    # See if PO assumption holds
> VGAM::lrtest(PO.vglm, NPO.vglm)

Likelihood ratio test

Model 1: Sat ~ Infl + Type + Cont
Model 2: Sat ~ Infl + Type + Cont
  #Df LogLik Df Chisq Pr(>Chisq)
1 136  -1740
2 130  -1735 -6  8.57        0.2
```

(b) Investigate whether any of the two-factor interactions among `Infl`, `Type`, and `Cont` add substantially to goodness of fit of this model. (Hint: use `stepAIC()`, with the scope formula `~ .^2` and `direction="forward"`.)
★ Forward selection from the main effects model adds the interactions of `Infl:Type` and `Type:Cont`. That is, the effect of type of dwelling on satisfaction varies both with influence and contact with other residents.

```
> house.step <- stepAIC(PO.mod, scope = ~.^2, direction = "forward")

Start:  AIC=3495.2
Sat ~ Infl + Type + Cont

            Df  AIC
+ Infl:Type  6 3485
+ Type:Cont  3 3492
<none>         3495
+ Infl:Cont  2 3499

Step:  AIC=3484.6
Sat ~ Infl + Type + Cont + Infl:Type

            Df  AIC
+ Type:Cont  3 3483
<none>         3485
+ Infl:Cont  2 3489

Step:  AIC=3482.7
Sat ~ Infl + Type + Cont + Infl:Type + Type:Cont

            Df  AIC
<none>         3483
+ Infl:Cont  2 3487
```

The result of `stepAIC()` includes an `anova` summary of the steps.

```
> house.step$anova

Stepwise Model Path
Analysis of Deviance Table

Initial Model:
Sat ~ Infl + Type + Cont

Final Model:
Sat ~ Infl + Type + Cont + Infl:Type + Type:Cont


         Step Df Deviance Resid. Df Resid. Dev    AIC
1                              1673     3479.1 3495.1
2 + Infl:Type  6   22.509      1667     3456.6 3484.6
3 + Type:Cont  3    7.945      1664     3448.7 3482.7

> Anova(house.step)

Analysis of Deviance Table (Type II tests)

Response: Sat
          LR Chisq Df Pr(>Chisq)
Infl         106.5  2    < 2e-16 ***
Type          55.9  3    4.4e-12 ***
Cont          15.1  1     0.0001 ***
Infl:Type     21.8  6     0.0013 **
Type:Cont      7.9  3     0.0472 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The next step, refitting the model chosen by `stepAIC()` is not strictly necessary (because that model was saved as `house.step`), but using the argument `Hess=TRUE` saves computation for some of the plots below.

```
> PO.mod2 <- polr(formula = Sat ~ Infl + Type + Cont + Infl:Type + Type:Cont,
+                 data = housing, Hess=TRUE, weights=Freq)
```

(c) For your chosen model from the previous step, use the methods of Section 8.1.5 to plot the probabilities of the categories of satisfaction.

★ A first step in plotting is to join the data with the predicted values.

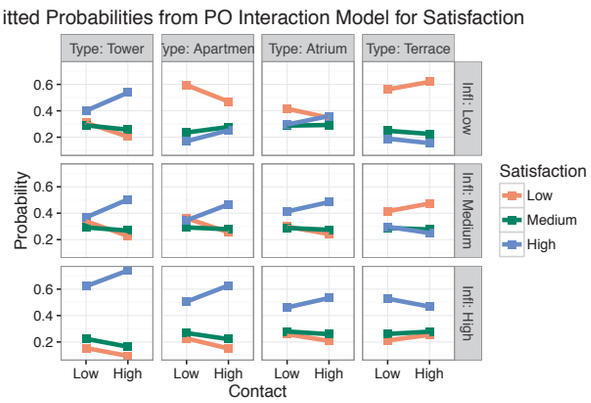
```
> fit.step <- cbind(housing, predict(house.step, type="probs"))
> head(fit.step)

     Sat   Infl  Type Cont Freq     Low  Medium    High
1    Low    Low Tower  Low   21 0.30948 0.28988 0.40064
2 Medium    Low Tower  Low   21 0.30948 0.28988 0.40064
3   High    Low Tower  Low   28 0.30948 0.28988 0.40064
4    Low Medium Tower  Low   34 0.33933 0.29227 0.36841
5 Medium Medium Tower  Low   22 0.33933 0.29227 0.36841
6   High Medium Tower  Low   36 0.33933 0.29227 0.36841
```

Plotting predicted probabilities with ggplot2 is quite flexible, but requires that the data in `fit.step` be reshaped to long format.

```
> library(reshape2)
> plotdat <- melt(fit.step,
+                 id.vars = c("Sat", "Infl", "Type", "Cont"),
+                 measure.vars = c("Low", "Medium", "High"),
+                 variable.name = "Satisfaction",
+                 value.name = "Probability")
```

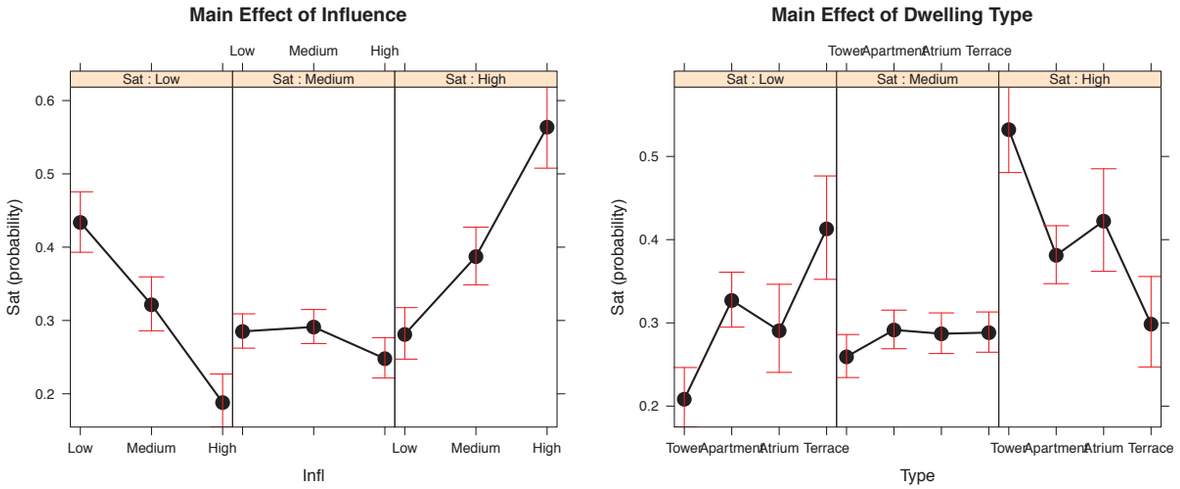
Then, we can plot `y = Probability` against one factor as `x`, and use `facet_grid()` to show panels for the other factors as rows and columns. This gives what we call a "full-model plot."

```
> library(ggplot2)
> ggplot(plotdat, aes(x = as.integer(Cont), y = Probability, color = Satisfaction)) +
+   facet_grid(Infl ~ Type, labeller=label_both) +
+   geom_point(size=2.5, shape=15) +
+   geom_line(size=1.5) +
+   xlab("Contact") +
+   ggtitle("Fitted Probabilities from PO Interaction Model for Satisfaction") +
+   scale_x_discrete(breaks=1:2, labels=c("Low","High")) +
+   theme_bw()
```



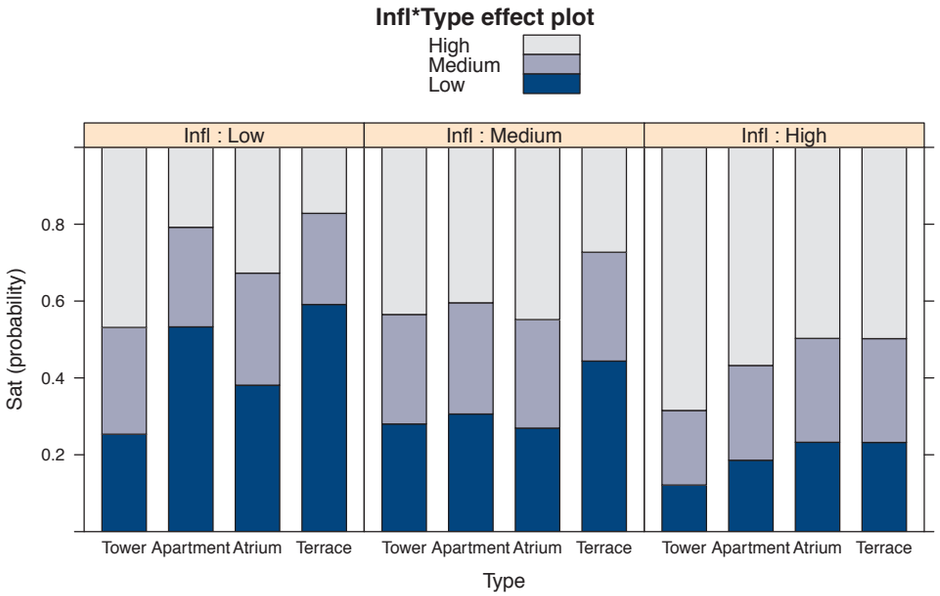
For ease of interpreting the model, effect plots give a more convenient visual summary. The model has 5 terms, and it is useful to plot them all; we only show selected terms here.
Main effects of influence and type:

```
> plot(Effect("Infl", PO.mod2), main="Main Effect of Influence", layout=c(3,1))
> plot(Effect("Type", PO.mod2), main="Main Effect of Dwelling Type", layout=c(3,1))
```



Interaction of influence and type:

```
> plot(Effect(c("Infl","Type"), PO.mod2), style="stacked")
```



(d) Write a brief summary of these analyses, interpreting *how* satisfaction with housing depends on the predictor variables.

★ As Influence increases, high satisfaction also increases, low satisfaction decreases and medium satisfaction remains relatively constant. Dwelling type also has opposite effects on high and low satisfaction, with Tower dwellers being highest on high satisfaction and Terrace dwellers lowest. Again, medium satisfaction seems to be relatively static across dwelling type. High satisfaction also increases with high contact.

Exercise 8.3 The data *TV* on television viewing was analyzed using correspondence analysis in Example 6.4, ignoring the variable `Time`, and extended in Exercise 6.9. Treating `Network` as a three-level response variable, fit a generalized logit model (Section 8.3) to explain the variation in viewing in relation to `Day` and `Time`. The *TV* data is a three-way table, so you will need to convert it to a frequency data frame first.

```
> data("TV", package="vcdExtra")
> TV.df <- as.data.frame.table(TV)
```

(a) Fit the main-effects model, `Network ~ Day + Time`, with `multinom()`. Note that you will have to supply the `weights` argument because each row of `TV.df` represents the number of viewers in the `Freq` variable.

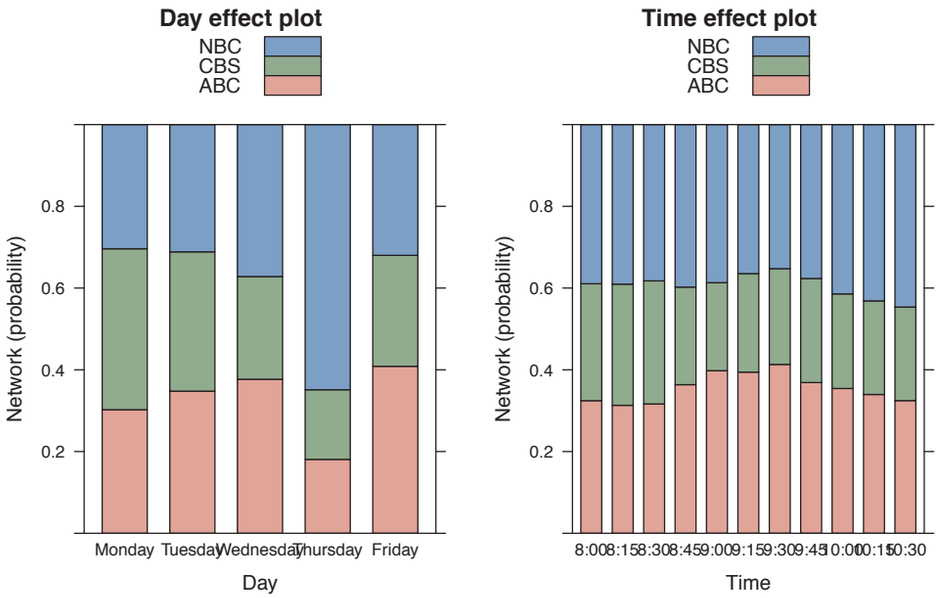
★

```
> library(nnet)
> tv.model = multinom(Network ~ Day + Time, data = TV.df,
+                     weights = Freq, Hess = TRUE)

# weights:  48 (30 variable)
initial  value 41318.808177
iter  10 value 38935.947713
iter  20 value 38818.222728
iter  30 value 38756.956301
final  value 38752.186202
converged

> car::Anova(tv.model)

Analysis of Deviance Table (Type II tests)

Response: Network
     LR Chisq Df Pr(>Chisq)
Day      3400  8     <2e-16 ***
Time      301 20     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both `Day` and `Time` are clearly significant when comparing the Networks.

(b) Prepare an effects plot for the fitted probabilities in this model.
★

```
> library(effects)
> plot(allEffects(tv.model), style = "stacked")
```

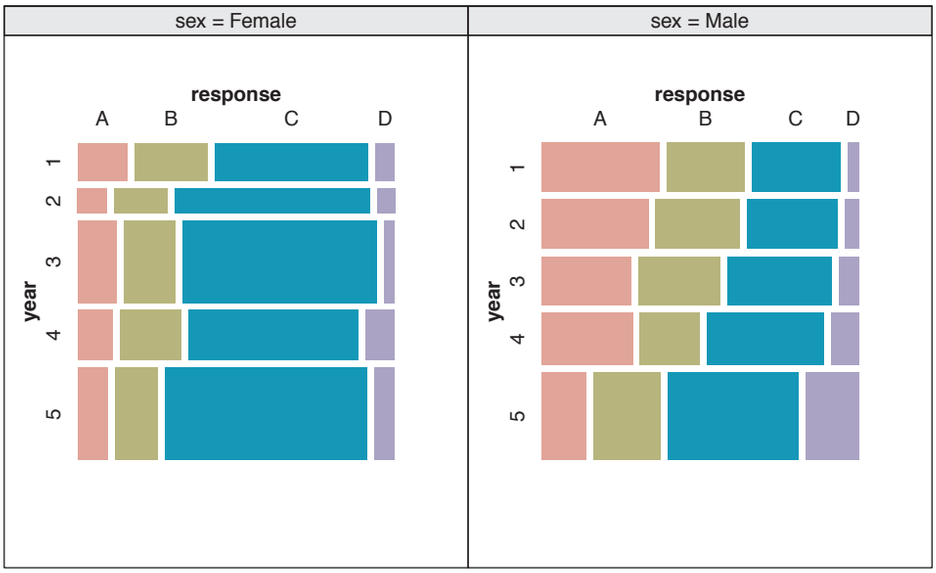


(c) Interpret these results in comparison to the correspondence analysis in Example 6.4.
★ The effect plots for Day clearly show a peak for NBC on Thursday, at the expense of the two other networks. ABC has most viewers on Tuesday and Wednesday, whereas CBS is the most popular Network on Monday. These results are consistent with the results from the correspondence analysis. Regarding time, ABC is most viewed from 8:45 to 9:30—after that, the viewing probability decreases in favor of NBC.

Exercise 8.4 * Refer to Exercise 5.10 for a description of the *Vietnam* data set in vcdExtra. The goal here is to fit models for the polytomous `response` variable in relation to `year` and `sex`.

(a) Fit the proportional odds model to these data, allowing an interaction of `year` and `sex`.
★ The Vietnam data is pretabulated; for a first explorative view, we transform it into tabular format, and use a conditional mosaicplot (with Marimekko shading) to get ideas on the structure.

```
> library(vcdExtra)
> data(Vietnam)
> Vtn_tab = xtabs(Freq ~ ., data = Vietnam)
> cotabplot( ~ year + response | sex, data = Vtn_tab,
+           gp = shading_Marimekko(Vtn_tab[1,,]))
```



Whereas male students have a clear tendency to choose the more peaceful options C and D only with higher study year (the mosaic plot suggests a linear relationship of `year` and `response`), the situation is less clear for female students—there seems to be a general tendency for response C (negotiations), independent from the study year.
The proportional odds model is fitted using `polr()`. Before, we make sure that the response is indeed treated as ordinal:

```
> library(MASS)
> Vietnam$response = ordered(Vietnam$response)
> Vtn_polr = polr(response ~ year * sex, data = Vietnam,
+                 weights = Freq, Hess = TRUE)
> summary(Vtn_polr)

Call:
polr(formula = response ~ year * sex, data = Vietnam, weights = Freq,
    Hess = TRUE)

Coefficients:
             Value Std. Error t value
year          0.101     0.0548    1.84
sexMale      -1.437     0.2230   -6.44
year:sexMale  0.233     0.0603    3.87

Intercepts:
    Value  Std. Error t value
A|B -1.353  0.206     -6.574
B|C -0.254  0.204     -1.244
C|D  2.188  0.210     10.397

Residual Deviance: 7757.06
AIC: 7769.06

> car::Anova(Vtn_polr)

Analysis of Deviance Table (Type II tests)

Response: response
         LR Chisq Df Pr(>Chisq)
year        166.5  1    < 2e-16 ***
sex          58.5  1    2.1e-14 ***
year:sex     15.0  1    0.00011 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Clearly, all terms are significant, including the interaction term with `sex` as expected from the mosaic plots.

(b) Is there evidence that the proportional odds assumption does not hold for this data set? Use the methods described in Section 8.1 to assess this.

★ We refit the model using `vglm()`, and compare it to the non-proportional odds model with `lrtest()`:

```
> library(VGAM)
> Vtn_PO = vglm(response ~ year * sex, data = Vietnam,
+               weights = Freq, family = cumulative(parallel = TRUE))
> Vtn_NPO = vglm(response ~ year * sex, data = Vietnam,
+                weights = Freq, family = cumulative(parallel = FALSE))
> VGAM::lrtest(Vtn_NPO, Vtn_PO)

Likelihood ratio test

Model 1: response ~ year * sex
Model 2: response ~ year * sex
  #Df LogLik Df Chisq Pr(>Chisq)
1 108  -3840
2 114  -3879  6  77.6    1.1e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The unconstrained model fits better than the proportional odds model—again as expected, since for women, the proportional odds assumption seems not be reasonable.

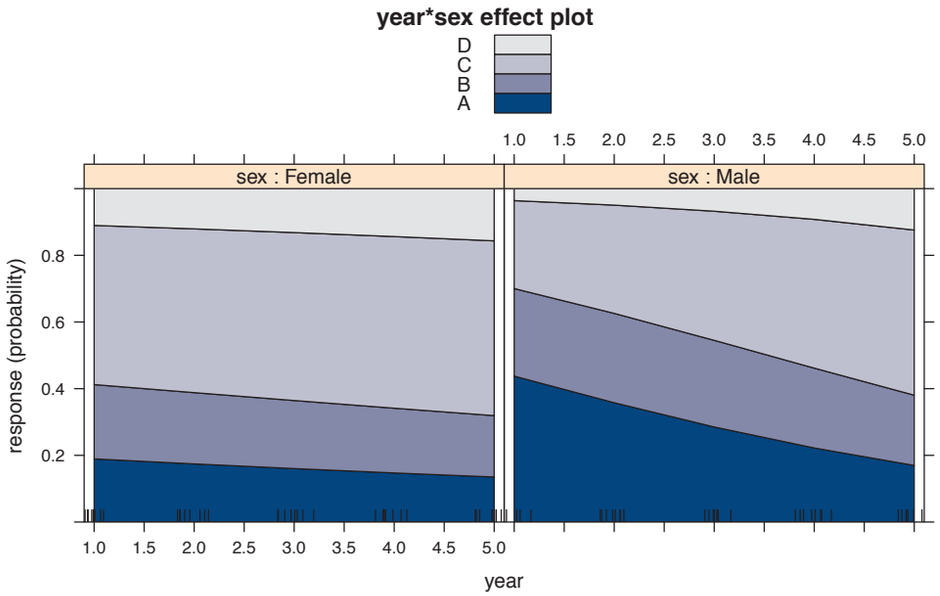
(c) Fit the multinomial logistic model, also allowing an interaction. Use car::`Anova()` to assess the model terms. ★

```
> library(nnet)
> library(car)
> Vtn_mtn = multinom(response ~ year * sex, data = Vietnam, weights = Freq)

# weights:  20 (12 variable)
initial  value 4362.668354
iter  10 value 3927.047674
final  value 3838.854903
converged

> summary(Vtn_mtn)

Call:
multinom(formula = response ~ year * sex, data = Vietnam, weights = Freq)

Coefficients:
  (Intercept)        year  sexMale year:sexMale
B     0.39288 -0.00051719 -0.97197      0.14692
C     1.14525  0.14751329 -1.88983      0.17082
D    -1.31002  0.19089252 -2.00838      0.45507

Std. Errors:
  (Intercept)     year sexMale year:sexMale
B     0.39149 0.110209 0.41055     0.116272
C     0.33686 0.093731 0.35726     0.099851
D     0.56775 0.150979 0.61640     0.162177

Residual Deviance: 7677.7
AIC: 7701.7

> car::Anova(Vtn_mtn)

Analysis of Deviance Table (Type II tests)

Response: response
         LR Chisq Df Pr(>Chisq)
year        175.9  3     <2e-16 ***
sex         137.6  3     <2e-16 ***
year:sex      7.8  3      0.051 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All terms are again significant, alltough weakly so for the interaction term.

(d) Produce an effect plot for this model and describe the nature of the interaction. ★

```
> library(effects)
> plot(allEffects(Vtn_polr), style = "stacked")
```



The effect of `year` on `response` under the fitted model differentiates male and female students: Whereas the majority of female students prefer the more peaceful options in general (with a slight increase over the years), a majority of male students are in favor of more violent options as freshmen, but change their opinion dramatically during the studies so that the situation is reversed at the end. The model is consistent with the findings from the exploratory mosaic plot.

(e) Fit the simpler multinomial model in which there is no effect of year for females and the effect of year is linear for males (on the logit scale). Test whether this model is significantly worse than the general multinomial model with interaction.

★ To achieve this, we first construct an artificial variable `yearMale`, with non-zero entries only for male students:

```
> Vietnam = within(Vietnam, yearMale <- year * (sex == "Male"))
> Vtn_mtn2 = multinom(response ~ sex + yearMale, data = Vietnam, weights = Freq)

# weights:  16 (9 variable)
initial  value 4362.668354
iter  10 value 3894.170452
final  value 3841.568268
converged

> Anova(Vtn_mtn2)

Analysis of Deviance Table (Type II tests)

Response: response
         LR Chisq Df Pr(>Chisq)
sex           243  3     <2e-16 ***
yearMale      178  3     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(Vtn_mtn, Vtn_mtn2)

Likelihood ratio tests of Multinomial Models

Response: response
          Model Resid. df Resid. Dev   Test    Df LR stat. Pr(Chi)
1 sex + yearMale       111     7683.1
2     year * sex       108     7677.7 1 vs 2     3   5.4267 0.14309
```

The simpler model is not significant worse than the general model with interaction, and should therefore be preferred.

Chapter 9 Loglinear and Logit Models for Contingency Tables

Exercise 9.1 Consider the data set *DaytonSurvey* (described in Example 2.6), giving results of a survey of use of alcohol (A), cigarettes (C), and marijuana (M) among high school seniors. For this exercise, ignore the variables `sex` and `race`, by working with the marginal table `Dayton.ACM`, a $2 \times 2 \times 2$ table in frequency data frame form.

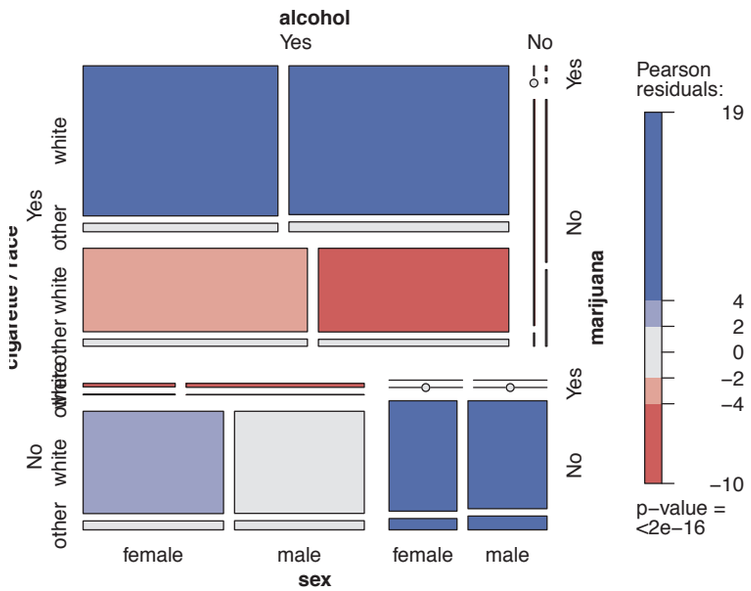
```
> Dayton.ACM <- aggregate(Freq ~ cigarette + alcohol + marijuana,
+                         data=DaytonSurvey, FUN=sum)
```

(a) Use `loglm()` to fit the model of mutual independence, [A][C][M].
★

```
> library(vcdExtra)
> data(DaytonSurvey)
> Dayton.ACM = aggregate(Freq ~ cigarette + alcohol + marijuana,
+                        data = DaytonSurvey, FUN = sum)
> ACM.mutual = loglm(Freq ~ ., data = Dayton.ACM)
> ACM.mutual

Call:
loglm(formula = Freq ~ ., data = Dayton.ACM)

Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 1286.0  4        0
Pearson          1411.4  4        0
```

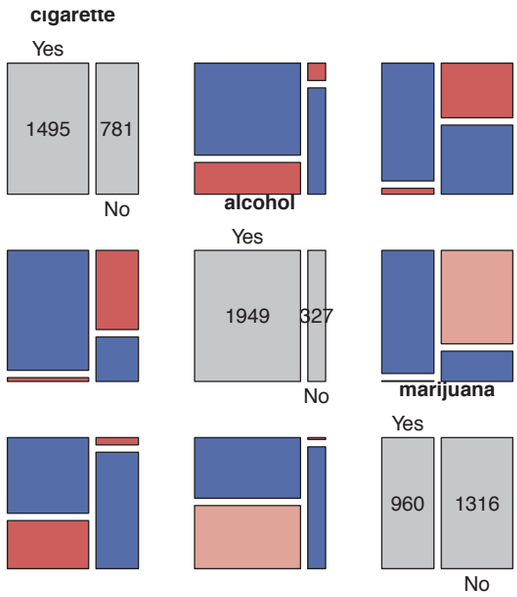
(b) Prepare mosaic display(s) for associations among these variables. Give a verbal description of the association between cigarette and alcohol use.
★ We can start creating a mosaic display for the complete table, using residual-based shading:

```
> mosaic(Freq ~ ., data = DaytonSurvey, shade = TRUE)
```



By default, the expected values (and residuals) are taken from the mutual independence model. As we can see from the structure and the shading, the model fits badly. Alternatively, we can inspect all pairwise (marginal) associations using `pairs()`. For this, we need to transform the data into contingency table form first:

```
> Dayton.tab = xtabs(Freq ~ ., data = Dayton.ACM)
> pairs(Dayton.tab, type = "pairwise", shade = TRUE)
```

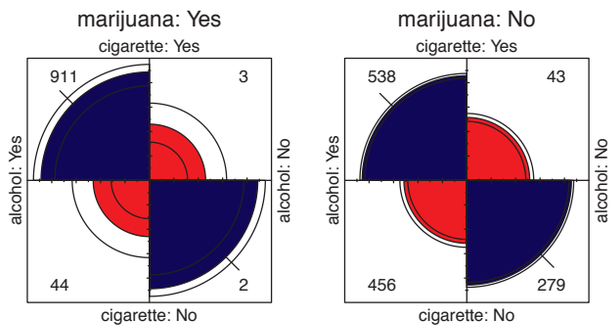


Clearly, the variables are associated mutually. In particular, in the mosaic for `cigarette` and `alcohol`, two many people who drink also smoke (and, conversely, two many people who don't drink also don't smoke) than would be expected under the mutual independence model.

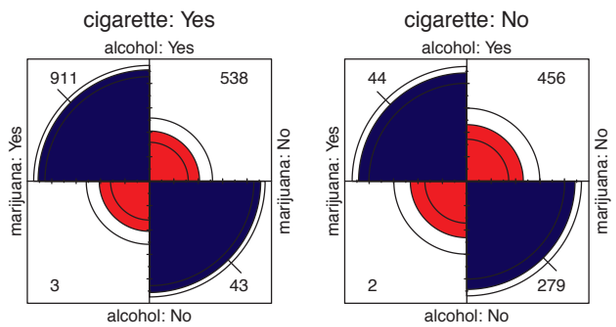
(c) Use `fourfold()` to produce fourfold plots for each pair of variables, AC, AM, and CM, stratified by the remaining one. Describe these associations verbally.

★ A fourfold plot visualizes the log-odds ratios between two binary variables. To plot all pairwise combinations, given the third, of the Dayton data, we permute the table dimensions:

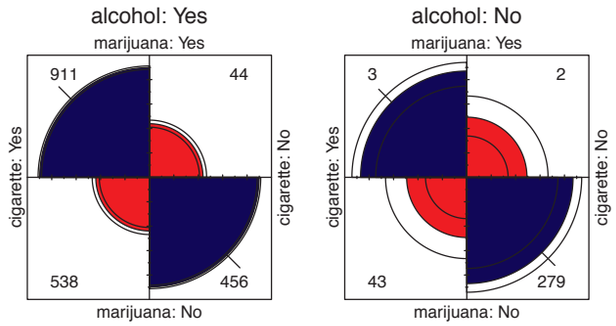
```
> fourfold(aperm(Dayton.tab, c(1, 2, 3)))
```




```
> fourfold(aperm(Dayton.tab, c(2, 3, 1)))
```




```
> fourfold(aperm(Dayton.tab, c(3, 1, 2)))
```



All 6 displays exhibit a positive association between two of the three drugs (more people who drink also smoke than expected under independence, etc.) which illustrates that the hypothesis of mutual independence is not tenable. However, these mutual association patterns are independent from the third, so a reasonable model might be the homogeneous association model (= no three-way association), as investigated in the next exercise.

Exercise 9.2 Continue the analysis of the *DaytonSurvey* data by fitting the following models:

(a) Joint independence, [AC][M]
★

```
> ACM.joint = loglm(Freq ~ alcohol * cigarette + marijuana, data = Dayton.ACM)
> ACM.joint

Call:
loglm(formula = Freq ~ alcohol * cigarette + marijuana, data = Dayton.ACM)
```

```
Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 843.83  3        0
Pearson          704.91  3        0
```

(b) Conditional independence, [AM][CM]
★

```
> ACM.cond = loglm(Freq ~ (alcohol + cigarette) * marijuana, data = Dayton.ACM)
> ACM.cond

Call:
loglm(formula = Freq ~ (alcohol + cigarette) * marijuana, data = Dayton.ACM)

Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 187.75  2        0
Pearson          177.61  2        0
```

(c) Homogeneous association, [AC][AM][CM]
★

```
> ACM.hom = loglm(Freq ~ alcohol * cigarette + alcohol * marijuana +
+                 cigarette * marijuana, data = Dayton.ACM)
> ACM.hom

Call:
loglm(formula = Freq ~ alcohol * cigarette + alcohol * marijuana +
    cigarette * marijuana, data = Dayton.ACM)

Statistics:
                      X^2 df P(> X^2)
Likelihood Ratio 0.37399  1  0.54084
Pearson          0.40110  1  0.52652
```

(d) Prepare a table giving the goodness-of-fit tests for these models, as well as the model of mutual independence, [A][C][M], and the saturated model, [ACM]. *Hint*: `anova()` and `LRstats()` are useful here. Which model appears to give the most reasonable fit?
★

```
> anova(ACM.mutual, ACM.joint, ACM.cond, ACM.hom, test = "Chisq")

LR tests for hierarchical log-linear models

Model 1:
 Freq ~ .
Model 2:
 Freq ~ alcohol * cigarette + marijuana
Model 3:
 Freq ~ (alcohol + cigarette) * marijuana
Model 4:
 Freq ~ alcohol * cigarette + alcohol * marijuana + cigarette * marijuana

            Deviance df Delta(Dev) Delta(df) P(> Delta(Dev)
Model 1   1286.01995  4
Model 2    843.82664  3  442.19331         1        0.00000
Model 3    187.75430  2  656.07234         1        0.00000
Model 4      0.37399  1  187.38032         1        0.00000
Saturated    0.00000  0    0.37399         1        0.54084

> LRstats(ACM.mutual, ACM.joint, ACM.cond, ACM.hom)

Likelihood summary table:
            AIC  BIC LR Chisq Df Pr(>Chisq)
ACM.mutual 1343 1343     1286  4     <2e-16 ***
ACM.joint   903  903      844  3     <2e-16 ***
ACM.cond    249  249      188  2     <2e-16 ***
ACM.hom      63   64        0  1       0.54
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

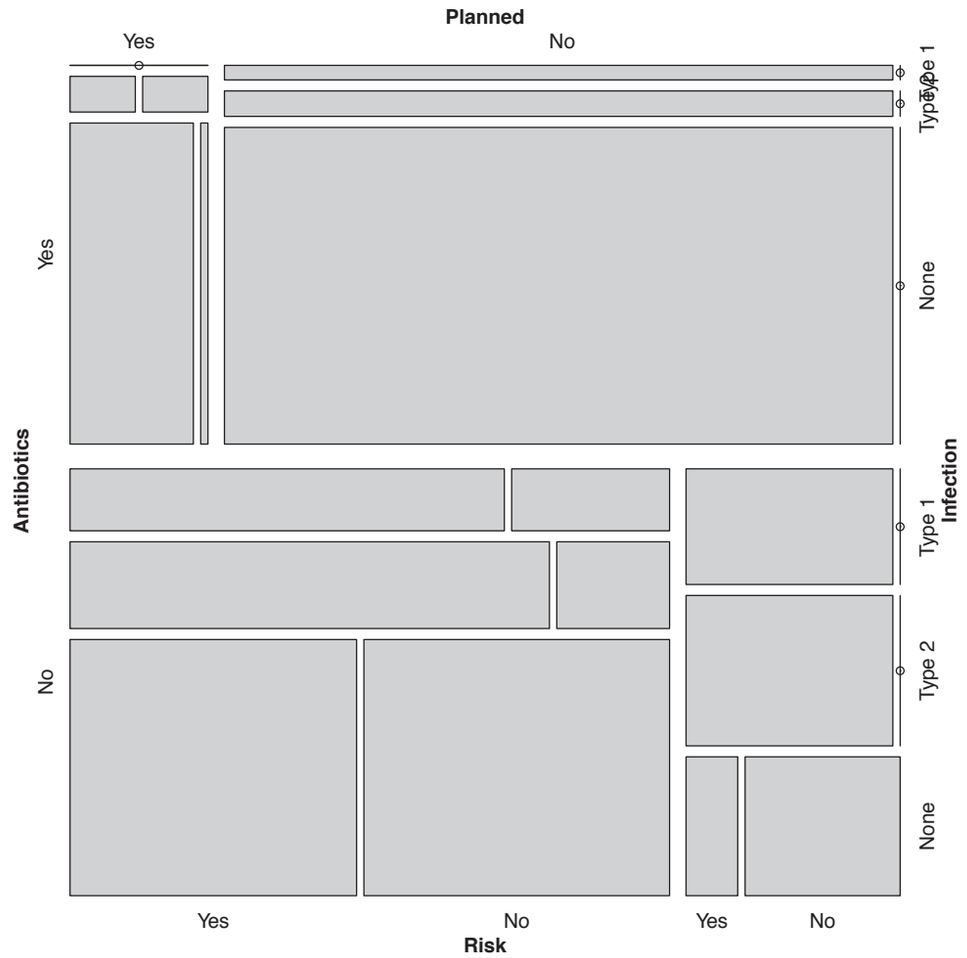
The homogeneous association model is clearly the best model: it is not significant worse than the saturated model, and has the lowest AIC and BIC values.

Exercise 9.3 The data set *Caesar* in vcdExtra gives a 3×2^3 frequency table classifying 251 women who gave birth by Caesarian section by `Infection` (three levels: none, Type 1, Type2) and `Risk`, whether `Antibiotics`

were used, and whether the Caesarian section was `Planned` or not. `Infection` is a natural response variable, but the table has quite a few zeros.

(a) Use `structable()` and `mosaic()` to see the locations of the zero cells in this table. ★

```
> library(vcdExtra)
> data(Caesar)
> mosaic(aperm(Caesar, c(3,4,1,2)))
> structable(aperm(Caesar, c(3,4,1,2)))

                     Planned Yes     No
                     Risk    Yes No Yes No
Antibiotics Infection
Yes         Type 1              0  0   4  0
            Type 2              1  1   7  0
            None               17  1  87  0
No          Type 1             11  4  10  0
            Type 2             17  4  13  0
            None               30 32   3  9
```



Zero cells occur essentially for unplanned sections without risk factors (except for no antibiotics/no infection), and planned sections with risk, antibiotics and Type 1 infection.

(b) Use `loglm()` to fit the baseline model [I][RAP]. Is there any problem due to zero cells indicated in the output? ★

```
> caesar_base = loglm(~ Infection + Planned * Risk * Antibiotics, data = Caesar)
> caesar_base

Call:
loglm(formula = ~Infection + Planned * Risk * Antibiotics, data = Caesar)
```

```
Statistics:
                    X^2 df   P(> X^2)
Likelihood Ratio 85.11 14 3.1566e-12
Pearson           NaN 14        NaN
```

The NaN value for the Pearson Chi-Squared statistics is suspicious. It stems from the fact that some of the expected values (unplanned sections without risk factors and with antibiotics) are 0, yielding NaN values for the corresponding Pearson residuals.

(c) For the purpose of this excercise, treat all the zero cells as *sampling zeros* by adding 0.5 to all cells, e.g., `Caesar1 <- Caesar + 0.5`. Refit the baseline model.
★

```
> Caesar1 = Caesar + 0.5
> caesar_sampling = loglm(~ Infection + Planned * Risk * Antibiotics, data = Caesar1)
> caesar_sampling

Call:
loglm(formula = ~Infection + Planned * Risk * Antibiotics, data = Caesar1)

Statistics:
                    X^2 df   P(> X^2)
Likelihood Ratio 77.865 14 7.0291e-11
Pearson          77.814 14 7.1805e-11
```

(d) Now fit a "main effects" model [IR][IA][IP][RAP] that allows associations of `Infection` with each of the predictors.
★

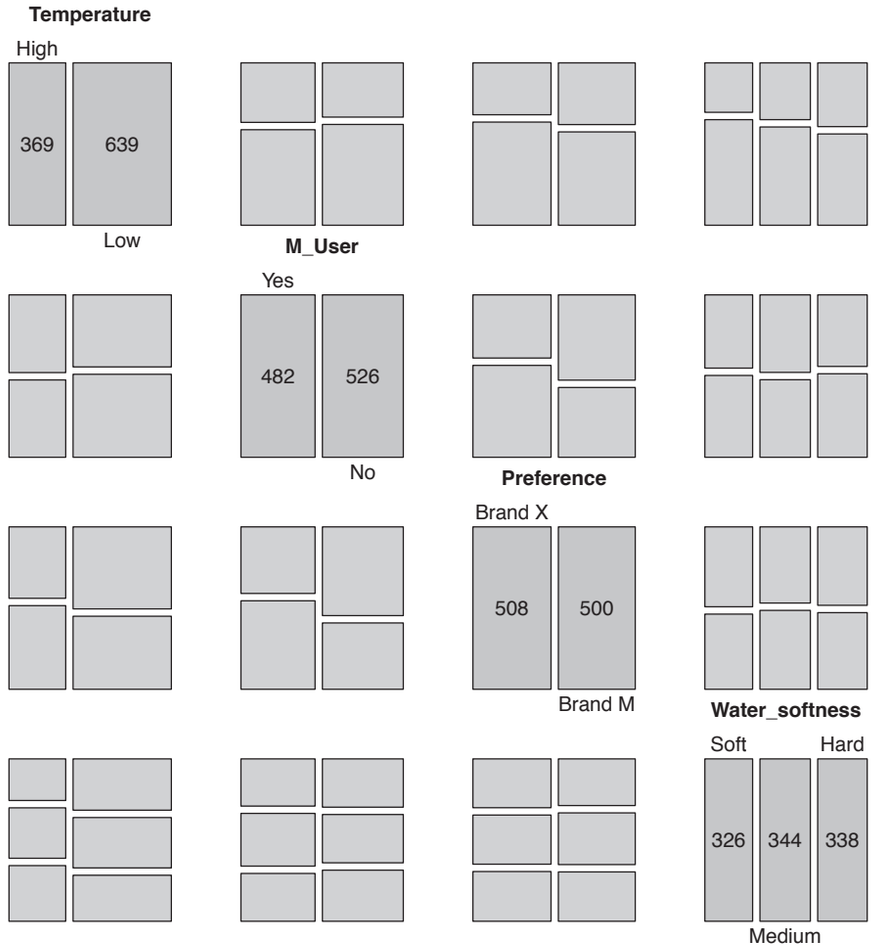
```
> caesar_main = loglm(~ Infection * (Risk + Antibiotics + Planned) +
+ (Risk * Antibiotics * Planned), data = Caesar1)
> caesar_main

Call:
loglm(formula = ~Infection * (Risk + Antibiotics + Planned) +
    (Risk * Antibiotics * Planned), data = Caesar1)

Statistics:
                    X^2 df   P(> X^2)
Likelihood Ratio 25.539  8 1.2592e-03
Pearson          73.682  8 9.0539e-13

> anova(caesar_sampling, caesar_main, test = "Chisq")

LR tests for hierarchical log-linear models

Model 1:
 ~Infection + Planned * Risk * Antibiotics
Model 2:
 ~Infection * (Risk + Antibiotics + Planned) + (Risk * Antibiotics * Planned)

          Deviance df Delta(Dev) Delta(df) P(> Delta(Dev)
Model 1     77.865 14
Model 2     25.539  8     52.326         6        0.00000
Saturated    0.000  0     25.539         8        0.00126
```

The model fits better than the base model, but nevertheless shows a lack of fit.

Exercise 9.4 The *Detergent* in vcdExtra gives a $2^3 \times 3$ table classifying a sample of $1,008$ consumers according to (a) expressed `Preference` for Brand "X" or Brand "M" in a blind trial, (b) `Temperature` of laundry water used, (c) previous use (`M_user`) of detergent Brand "M," and (d) the softness (`Water_softness`) of the laundry water used.

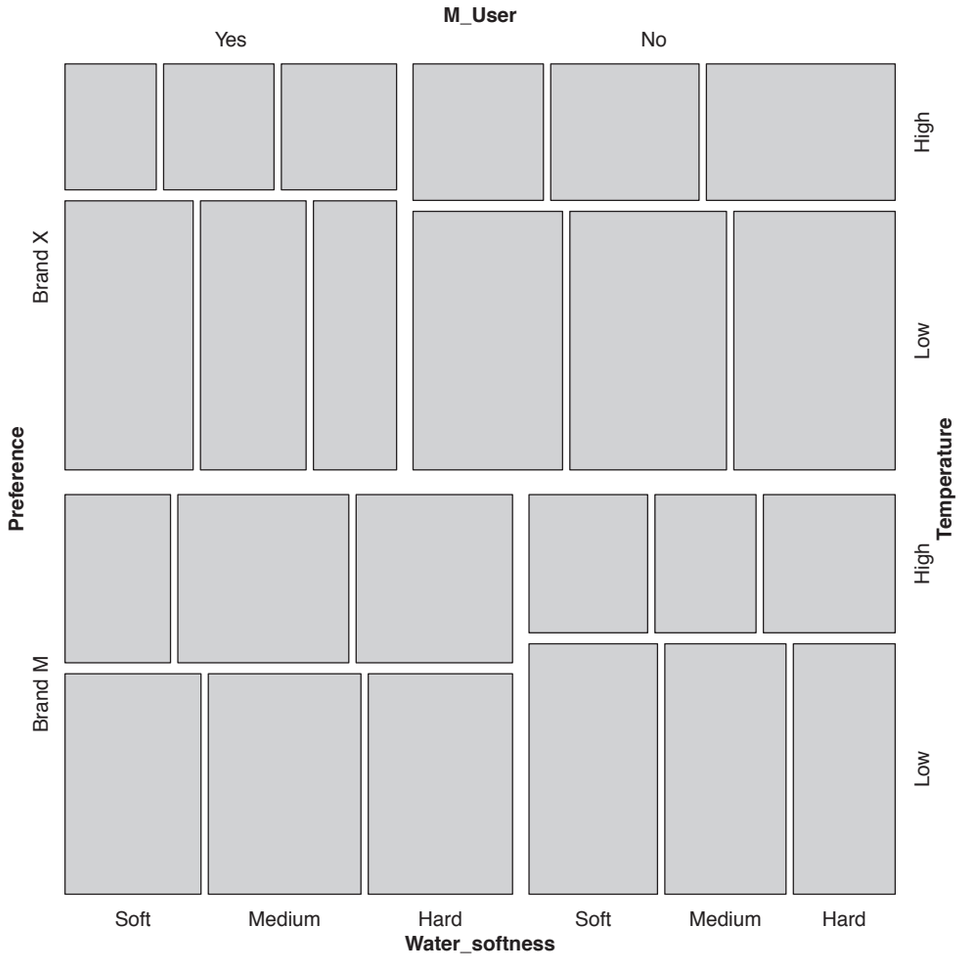
(a) Make some mosaic displays to visualize the associations among the table variables. Try using different orderings of the table variables to make associations related to `Preference` more apparent.
★ We can start with an exploratory pairs plot, showing all pairwise associations:

```
> library(vcdExtra)
> pairs(Detergent, type = "pairwise")
```



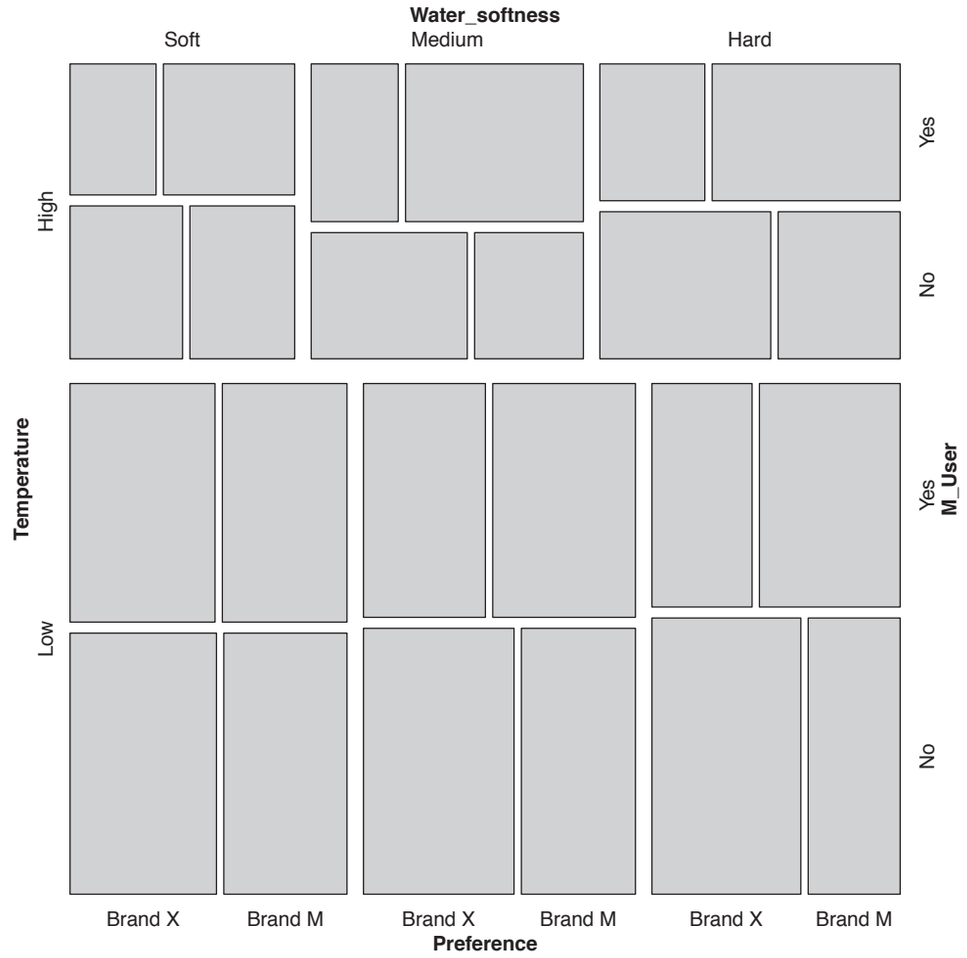
already suggesting a pairwise association between `M_User` and `Preference` as well as between `Temperature` and `Water_softness`, and (pairwise) independence of the other factors. In the next plot, we try a single mosaic display, splitting first by `M_User` and `Preference`, to see what remains *given* this effect:

```
> mosaic(~ Preference + M_User + Temperature + Water_softness, data
+ = Detergent)
```



This shows, for each combination of `M_User` and `Preference`, the relationship of `temperature` and `Water_softness`. Since the tiles do not align in all four cases, we might again suspect some association between these two variables.
Another approach in case of a response variable is to put it *last* in the mosaic, showing conditional (in)dependence structures:

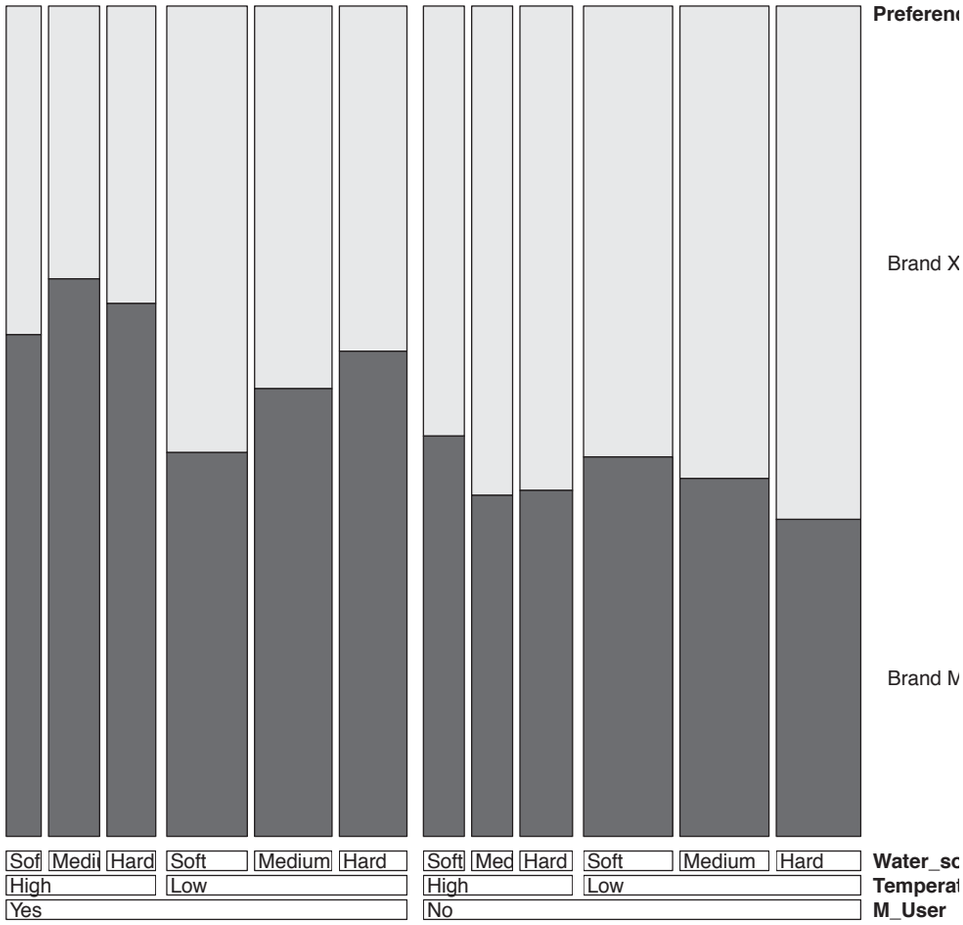
```
> mosaic(~ Temperature + Water_softness + M_User + Preference, data
+ = Detergent)
```



Looking at the mosaic "outside-in", we first notice the association between `Temperature` and `Water_softness`. Next, we inspect the relationship between `M_User` and `Preference`, *given* all combinations of the other two variables. They seem associated in all cases, except for low temperature and soft water. So finally, both mosaics lead to the same conclusion.

(b) Use a `doubledecker()` plot to visualize how `Preference` relates to the other factors. ★

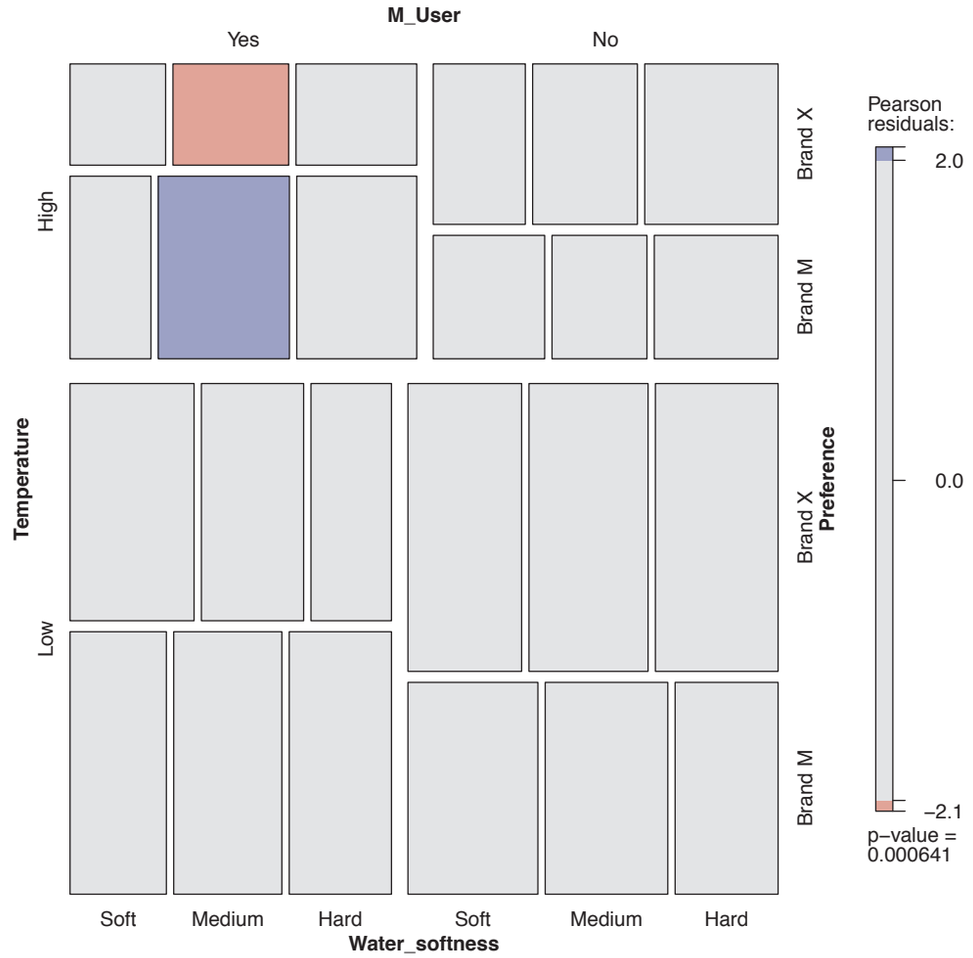
```
> doubledecker(Preference ~ M_User + Temperature + Water_softness, data = Detergent)
```



This shows an influence from `M_User` on `preference`, since all bars are higher for `Yes` than for `No`, and some additional influence of `Temperature` since the structure is different for `High` and `Low` (at least for respondents who previously used M).

(c) Use `loglm()` to fit the baseline model [P][TMW] for `Preference` as the response variable. Use a mosaic display to visualize the lack of fit for this model.
★

```
> detergent_base = loglm(~ Preference + Temperature * M_User *
+                        Water_softness, data = Detergent)
> detergent_base

Call:
loglm(formula = ~Preference + Temperature * M_User * Water_softness,
    data = Detergent)

Statistics:
                     X^2 df   P(> X^2)
Likelihood Ratio 32.826 11 0.00056150
Pearson          32.469 11 0.00064103

> mosaic(detergent_base)
```



The shaded tiles indicate too many M-Users using M for high temperatures and medium water softness than would be expected under the base model. A model that takes into account the findings from the exploratory analysis would be:

```
> detergent_model2 = loglm(~ Preference * M_User + Temperature *
+                          Water_softness, data = Detergent)
> detergent_model2

Call:
loglm(formula = ~Preference * M_User + Temperature * Water_softness,
    data = Detergent)

Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 16.248 15  0.36576
Pearson          16.727 15  0.33547
```

Although a mosaic plot with residual-based shading would still show a colored tile for this model, it is not significantly worse than the saturated model.

Chapter 10 Extending Loglinear Models

Exercise 10.1 Example 10.5 presented an analysis of the data on visual acuity for the subset of women in the *VisualAcuity* data. Carry out a parallel analysis of the models fit there for the men in this data set, given by:

```
> data("VisualAcuity", package="vcd")
> men <- subset(VisualAcuity, gender=="male", select=-gender)
```

★ Fit the independence, quasi-independence, symmetry, and quasi-symmetry models for the data men. These require the gnm (Turner and Firth, 2014) package. It is convenient to use update() to add terms to a model.

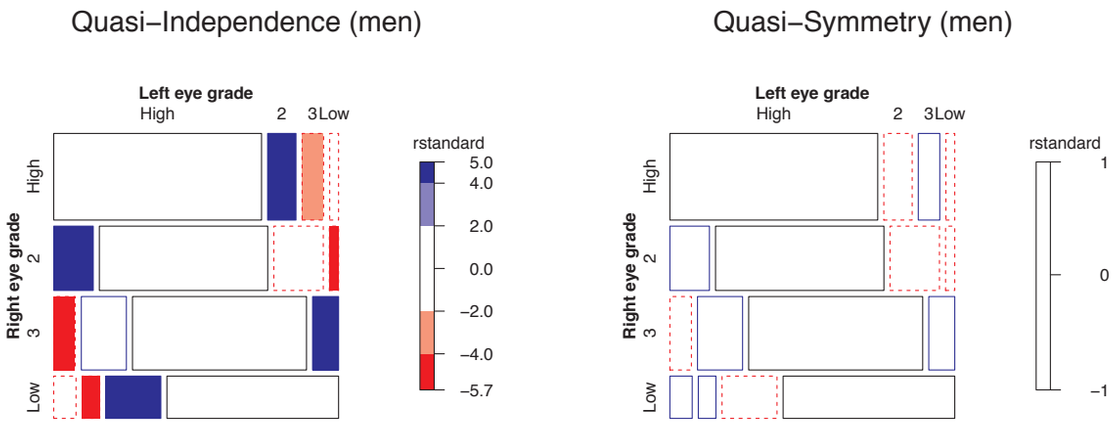
```
> library(gnm)
> indep <- glm(Freq ~ right + left,  data = men, family=poisson)
> quasi <- update(indep, . ~ . + Diag(right, left))
> symm <- glm(Freq ~ Symm(right, left),
+         data = men, family = poisson)
> qsymm <- update(symm, . ~ right + left + .)
```

Nested models can be compared using anova() and all models can be compared using LRstats(). Note that unlike the data for women, the symmetry model is preferred over quasi-symmetry by AIC and BIC.

```
> anova(indep, quasi, qsymm, test="Chisq")

Analysis of Deviance Table

Model 1: Freq ~ right + left
Model 2: Freq ~ right + left + Diag(right, left)
Model 3: Freq ~ right + left + Symm(right, left)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         9       2781
2         5         81  4     2701   <2e-16 ***
3         3          1  2       79   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

>   # test of marginal homogeneity
> anova(symm, qsymm, test="Chisq")

Analysis of Deviance Table

Model 1: Freq ~ Symm(right, left)
Model 2: Freq ~ right + left + Symm(right, left)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         6       4.77
2         3       1.09  3     3.68      0.3

>   # model summaries, with AIC and BIC
> models <- glmlist(indep, quasi, symm, qsymm)
> vcdExtra::LRstats(models)

Likelihood summary table:
       AIC  BIC LR Chisq Df Pr(>Chisq)
indep 2901 2906     2781  9    < 2e-16 ***
quasi  208  217       81  5    6.6e-16 ***
symm   131  138        5  6       0.57
qsymm  133  143        1  3       0.78
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As in the text, it is useful to display some of these using mosaic displays like Figure 10.8.

```
> labs <- c("High", "2", "3", "Low")
> largs <- list(set_varnames = c(right="Right eye grade", left="Left eye grade"),
+               set_labels=list(right=labs, left=labs))
> mosaic(quasi, ~right + left, residuals_type="rstandard", gp=shading_Friendly,
+        labeling_args=largs,
+        main="Quasi-Independence (men)"  )
> mosaic(qsymm, ~right + left, residuals_type="rstandard", gp=shading_Friendly,
+        labeling_args=largs,
+        main="Quasi-Symmetry (men)")
```



Exercise 10.2 Table 10.1 gives a 4×4 table of opinions about premarital sex and whether methods of birth control should be made available to teenagers aged 14–16, from the 1991 General Social Survey (Agresti, 2013, Table 10.3). Both variables are ordinal, and their grades are represented by the case of the row and column labels.

Table 10.1: Opinions about premarital sex and availability of teenage birth control. *Source*: Agresti (2013, Table 10.3).

Premarital sex	Birth control			
	DISAGREE	disagree	agree	AGREE
WRONG	81	68	60	38
Wrong	24	26	29	14
wrong	18	41	74	42
OK	36	57	161	157

(a) Fit the independence model to these data using `loglm()` or `glm()`.
★ First, enter the data into a matrix and assign row and column labels (`dimnames()`).

```
> birthcontrol <- matrix(c(
+  81, 68, 60, 38,
+  24, 26, 29, 14,
+  18, 41, 74, 42,
+  36, 57, 161, 157), 4, 4, byrow=TRUE)
>
> dimnames(birthcontrol) <-
+         list("presex" = c("WRONG", "Wrong", "wrong", "OK"),
+              "birthcontrol" = c("DISAGREE", "disagree", "agree", "AGREE"))
> birthcontrol

       birthcontrol
presex  DISAGREE disagree agree AGREE
  WRONG       81       68    60    38
  Wrong       24       26    29    14
  wrong       18       41    74    42
  OK          36       57   161   157
```

`loglm()` can handle the data in matrix form.

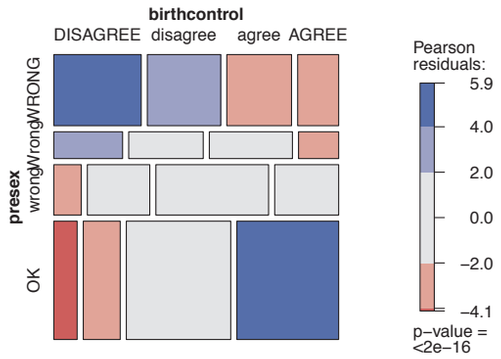
```
> loglm(~presex + birthcontrol, data=birthcontrol)

Call:
loglm(formula = ~presex + birthcontrol, data = birthcontrol)

Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 127.65  9        0
Pearson          128.68  9        0
```

`glm()` requires that the data be converted to a data frame, but is much more capable in the results it gives.

```
> birthcontrol.df <- as.data.frame(as.table(birthcontrol))
> birth.indep <- glm(Freq ~ presex + birthcontrol,
+                    data = birthcontrol.df, family = poisson)
> summary(birth.indep)

Call:
glm(formula = Freq ~ presex + birthcontrol, family = poisson,
    data = birthcontrol.df)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-4.547  -2.582   0.067   1.652   5.258

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)          3.7474     0.0962   38.95  < 2e-16 ***
presexWrong         -0.9768     0.1217   -8.03  9.8e-16 ***
presexwrong         -0.3446     0.0988   -3.49  0.00049 ***
presexOK             0.5092     0.0805    6.32  2.5e-10 ***
birthcontroldisagree 0.1886     0.1072    1.76  0.07861 .
birthcontrolagree    0.7118     0.0968    7.35  2.0e-13 ***
birthcontrolAGREE    0.4565     0.1014    4.50  6.7e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 431.08  on 15  degrees of freedom
Residual deviance: 127.65  on  9  degrees of freedom
AIC: 232.3

Number of Fisher Scoring iterations: 5

> anova(birth.indep)

Analysis of Deviance Table

Model: poisson, link: log

Response: Freq

Terms added sequentially (first to last)


             Df Deviance Resid. Df Resid. Dev
NULL                           15        431
presex        3    236.3        12        195
birthcontrol  3     67.1         9        128
```

(b) Make a mosaic display showing departure from independence and describe verbally the pattern of association. ★ Those who believe that premarital sex is wrong disagree that birth control should be made available to teenagers, in relation to the strength of their opinion about premarital sex. Only those who believe that premarital sex is OK support birth control for teenagers.

```
> mosaic(birthcontrol, shade=TRUE)
```



(c) Treating the categories as equally spaced, fit the $L \times L$ model of uniform association, as in Section 10.1. Test the difference against the independence model with a likelihood-ratio test.
★ The $L \times L$ model fits appreciably better than the independence model, for a "cost" of only one extra parameter in the model.

```
> library(gnm)
> linlin <- gnm(Freq ~ presex + birthcontrol +
+                      as.numeric(presex) * as.numeric(birthcontrol),
+               data=birthcontrol.df, family=poisson)
> anova(birth.indep, linlin, test="Chisq")

Analysis of Deviance Table

Model 1: Freq ~ presex + birthcontrol
Model 2: Freq ~ presex + birthcontrol + as.numeric(presex) + as.numeric(birthcontrol) +
    as.numeric(presex):as.numeric(birthcontrol)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         9      127.7
2         8       11.5  1      116   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(d) Fit the RC(1) model with `gnm()`, and test the difference of this against the model of uniform association.
★ The $L \times L$ model and the RC(1) model both fit well, but the former, with only one parameter for association, is deemed best by AIC and BIC.

```
> RC <- gnm(Freq ~ presex + birthcontrol + Mult(presex, birthcontrol),
+           data=birthcontrol.df, family=poisson, verbose=FALSE)
> anova(linlin, RC, test="Chisq")

Analysis of Deviance Table

Model 1: Freq ~ presex + birthcontrol + as.numeric(presex) + as.numeric(birthcontrol) +
    as.numeric(presex):as.numeric(birthcontrol)
Model 2: Freq ~ presex + birthcontrol + Mult(presex, birthcontrol)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         8      11.53
2         4       7.14  4     4.39     0.36

>   # compare all models
> LRstats(birth.indep, linlin, RC)

Likelihood summary table:
           AIC BIC LR Chisq Df Pr(>Chisq)
birth.indep 232 238    127.7  9     <2e-16 ***
linlin      118 124     11.5  8       0.17
RC          122 131      7.1  4       0.13
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

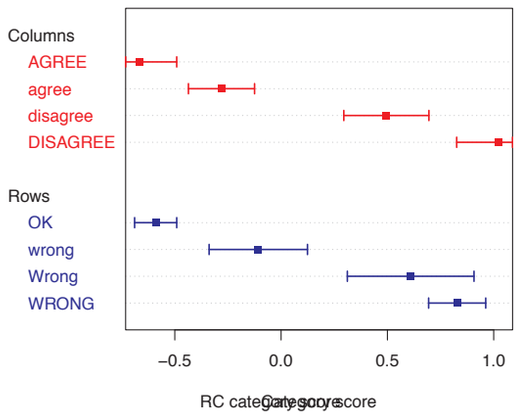
(e) Write a brief summary of these results, including plots useful for explaining the relationships in this data set.
★ Essentially, the association of attitudes toward premarital sex and birth control is that of two ordinal variables: the more one disagrees with premarital sex, the less one is likely to support birth control

for teenagers. The uniform association model, `linlin`, is the most parsimonious, assuming that the categories of `presex` and `birthcontrol` are equally spaced. From the coefficient for the $L \times L$, one can say that each step toward viewing premarital sex as OK multiplies the odds of agreement with birth control by 1.33, an increase of 33%.

```
> # odds ratio interpretation
> exp(coef(linlin)[10])

as.numeric(presex):as.numeric(birthcontrol)
                                      1.3309
```

The logmult package also fits the RC(1) model and provides a useful plot of the category scores together with confidence intervals via its `plot()` method. From this it can be seen that there is little reason to question that the category scores are equally spaced, except possibly for the WRONG and Wrong categories of `presex`.

```
> library(logmult)
> rc1 <- rc(birthcontrol, verbose=FALSE, weighting="marginal", se="jackknife")

Computing jackknife standard errors...

> plot(rc1, pch=15, conf=.95)
> title(xlab="RC category score")
```



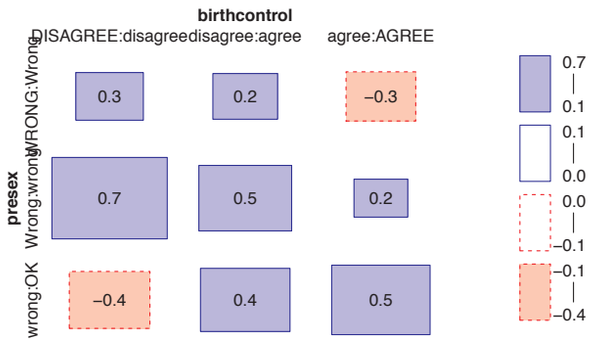
Exercise 10.3 For the data on attitudes toward birth control in Table 10.1,

(a) Calculate and plot the observed local log odds ratios.
★ The (log) odds ratios give the log odds of a 1-step change in attitude toward birth control for a 1-step change in the attitude toward premarital sex. Note that the mean of these values is similar to the common log odds ratio fit by the model of uniform association.

```
> (LOR <- loddsratio(birthcontrol))

log odds ratios for presex and birthcontrol

             birthcontrol
presex        DISAGREE:disagree disagree:agree agree:AGREE
  WRONG:Wrong           0.25498        0.23436    -0.27148
  Wrong:wrong           0.74316        0.48129     0.16184
  wrong:OK             -0.36367        0.44786     0.54124

> exp(mean(as.matrix(LOR)))

[1] 1.2811
```

A simple plot of these is a `tile()` plot, showing how the odds rations change systematically in each row, decreasing in the first two rows, but increasing in the last row. This is similar to the `corrplot()` shown in Figure 10.2 in the text.

```
> tile(LOR)
```



(b) Also fit the R, C, and R+C models.

★

```
> Rscore <- as.numeric(birthcontrol.df$presex)
> Cscore <- as.numeric(birthcontrol.df$birthcontrol)
>
> # row effects model (mental)
> roweff <- glm(Freq ~ presex + birthcontrol +
+                      presex:Cscore,
+                 family = poisson, data = birthcontrol.df)
>
> coleff <- glm(Freq ~ presex + birthcontrol +
+                      Rscore:birthcontrol,
+                 family = poisson, data = birthcontrol.df)
>
> RplusC <- glm(Freq ~ presex + birthcontrol +
+                      Rscore:birthcontrol + presex:Cscore,
+                 family = poisson, data = birthcontrol.df)
>
> LRstats(birth.indep, roweff, coleff, RplusC, linlin)

Likelihood summary table:
           AIC BIC LR Chisq Df Pr(>Chisq)
birth.indep 232 238    127.7  9     <2e-16 ***
roweff      120 128      9.5  6       0.15
coleff      120 128      9.1  6       0.17
RplusC      122 131      6.9  4       0.14
linlin      118 124     11.5  8       0.17
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

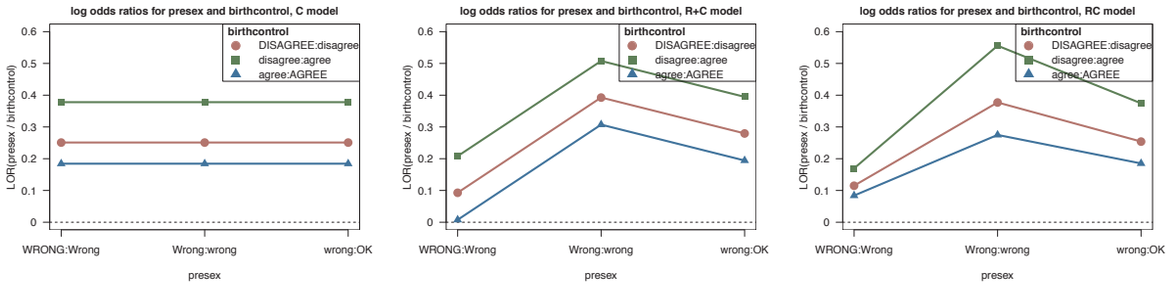
(c) Use the method described in Section 10.1.2 to visualize the structure of fitted local log odds ratios implied by each of these models, together with the RC(1) model.

★ Here is a simple function to extract the fitted values from a given model, calculate the log odds ratios, and plot them.

```
> plot_LOR_fit <- function(model, data, ...) {
+         dim <- dim(data)
+         fit <- matrix(fitted(model), nrow=dim[1], ncol=dim[2], dimnames=dimnames(data))
+         plot(LOR <- loddsratio(fit), ...)
+         invisible(LOR)
+ }
```

Using this, we plot the C, R+C, and RC(1) models below in the style shown in Figure 10.4 in the text. These plots show quite clearly how these models differ in the structure of their fitted log odds ratios.

```
> plot_LOR_fit(coleff, birthcontrol, confidence=FALSE, ylim=c(0, .6),
+         main="log odds ratios for presex and birthcontrol, C model")
>
> plot_LOR_fit(RplusC, birthcontrol, confidence=FALSE, ylim=c(0, .6),
+         main="log odds ratios for presex and birthcontrol, R+C model")
>
> plot_LOR_fit(RC, birthcontrol, confidence=FALSE, ylim=c(0, .6),
+         main="log odds ratios for presex and birthcontrol, RC model")
```



Exercise 10.4 The data set *gss8590* in logmult gives a $4 \times 5 \times 4$ table of education levels and occupational categories for the four combinations of gender and race from the General Social Surveys, 1985–1990, as reported by Wong (2001, Table 2). Wong (2010, Table 2.3B) later used the subset pertaining to women to illustrate RC(2) models. This data is created below as `Women.tab`, correcting an inconsistency to conform with the 2010 table.

```
> data("gss8590", package="logmult")
> Women.tab <- margin.table(gss8590[,,c("White Women", "Black Women")], 1:2)
> Women.tab[2,4] <- 49
> colnames(Women.tab)[5] <- "Farm"
```

(a) Fit the independence model, and also the RC(1) and RC(2) models using `rc()` with marginal weights, as illustrated in Example 10.4. Summarize these statistical tests in a table.

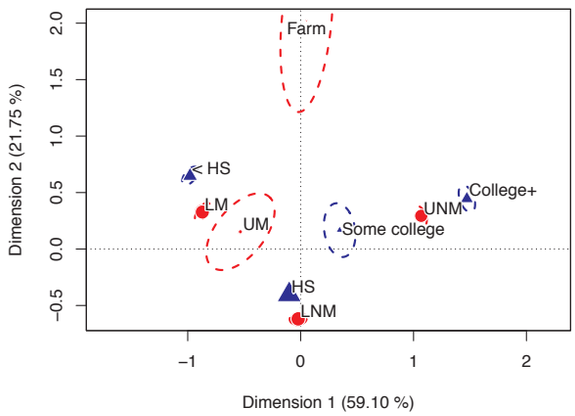
★ Only the RC2 model fits here.

```
> Women.df <- as.data.frame(as.table(Women.tab))
>
> indep <- glm(Freq ~ Education + Occupation,
+              data = Women.df, family = poisson)
> library(logmult)
> RC1 <- rc(Women.tab, verbose=FALSE, weighting="marginal", se="jackknife")

Computing jackknife standard errors...

> RC2 <- rc(Women.tab, verbose=FALSE, weighting="marginal", se="jackknife", nd=2)

Computing jackknife standard errors...

> LRstats(indep, RC1, RC2)

Likelihood summary table:
       AIC  BIC LR Chisq Df Pr(>Chisq)
indep 1506 1514     1373 12     <2e-16 ***
RC1    269  283      125  6     <2e-16 ***
RC2    153  171        1  2       0.74
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(b) Plot the solution for the RC(2) model with 68% confidence ellipses. What verbal labels would you use for the two dimensions?

★ Dimension 1 most clearly corresponds to Education, ordered from low (`< HS`) to high (`College`). Dimension 2 largely reflects the Occupation categories, with `Farm` being most different from the others.

```
> plot(RC2, conf=0.68, cex=1.5)
```



(c) Is there any indication that a simpler model, using integer scores for the row (Education) or column (Occupation) categories, or both, might suffice? If so, fit the analogous column effects, row effects, or $L \times L$ model, and compare with the models fit in part (a).

★ Given that the RC(1) model does not fit very well here, there is little chance that a simpler model using integer scores would be adequate.

Chapter 11 Generalized Linear Models for Count Data

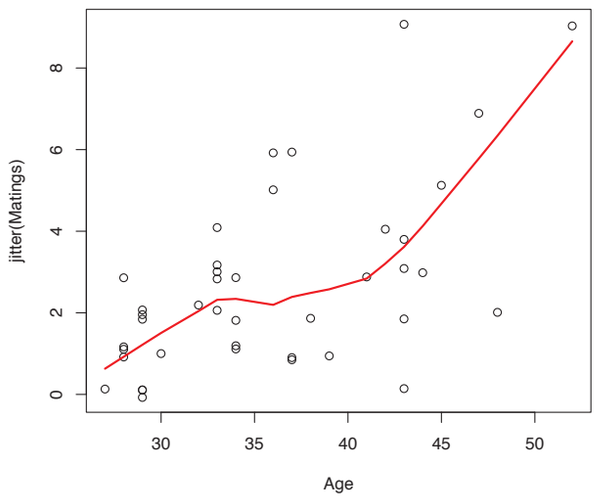
Exercise 11.1 Poole (1989) studied the mating behavior of elephants over 8 years in Amboseli National Park, Kenya. A focal aspect of the study concerned the mating success of males in relation to age, since larger males tend to be more successful in mating. Her data were used by Ramsey and Schafer (2002, Chapter 22) as a case study, and are contained in the Sleuth2 (Ramsey et al., 2012) package (Ramsey et al., 2012) as *case2201*.

For convenience, rename this to `elephants`, and study the relation between `Age` (at the beginning of the study) and number of successful `Matings` for the 41 adult male elephants observed over the course of this study, ranging in age from 27–52.

```
> data("case2201", package="Sleuth2")
> elephants <- case2201
> str(elephants)

'data.frame': 41 obs. of  2 variables:
 $ Age    : num  27 28 28 28 28 29 29 29 29 29 ...
 $ Matings: num  0 1 1 1 3 0 0 0 2 2 ...
```

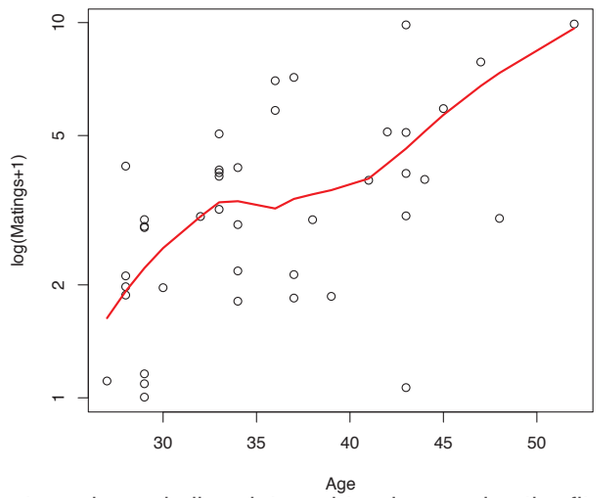
(a) Create some exploratory plots of `Matings` against `Age` in the styles illustrated in this chapter. To do this successfully, you will have to account for the fact that `Matings` has a range of only 0–9, and use some smoothing methods to show the trend.
★ The simplest plot just jitters the number of matings, and draws a smoothed lowess curve.

```
> with(elephants, {
+         plot(jitter(Matings) ~ Age)
+         lines(lowess(Age, Matings), lwd=2, col="red")}
+         )
```



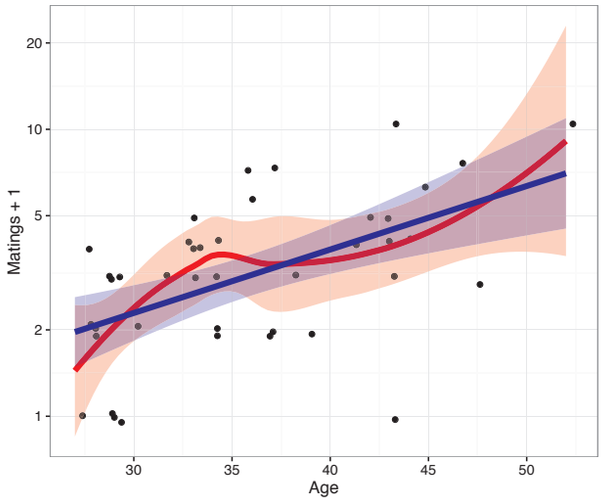
(b) Repeat (a) above, but now plotting `log(Matings+1)` against `Age` to approximate a Poisson regression with a log link and avoid problems with the zero counts.
★ As noted in the text, you can use `log="y"`, but then the Y variable should be used as Y+1 to avoid problems with 0 counts.

```
> with(elephants, {
+         plot(jitter(Matings+1) ~ Age, log="y", ylab="log(Matings+1)")
+         lines(lowess(Age, Matings+1), lwd=2, col="red")}
+         )
```



ggplot2 makes it easy to make a similar plot, and to also overlay the fit of a linear model. This also shows error bands for the loess and linear fits.

```
> library(ggplot2)
> ggplot(elephants, aes(x=Age, y=Matings+1)) +
+   geom_jitter() +
+         geom_smooth(method="loess", color="red", size=2, fill="red", alpha=0.2) +
+         geom_smooth(method="lm", color="blue", size=2, fill="blue", alpha=0.2) +
+         scale_y_log10(breaks=c(1,2,5,10, 20)) +
+   theme_bw()
```



(c) Fit a linear Poisson regression model for `Matings` against `Age`. Interpret the fitted model *verbally* from a graph of predicted number of matings and/or from the model coefficients. (*Hint*: Using `Age-27` will make the intercept directly interpretable.)

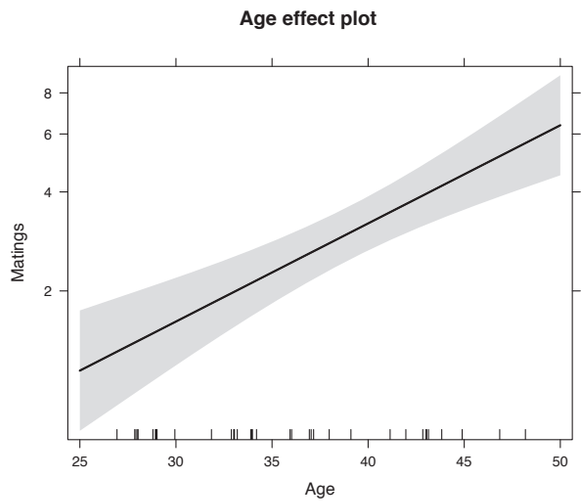
★ `Age-27` shifts the value of intercept to the minimum value in the data. In a model formula, use `I(Age-27)`. The coefficient for `Age` then relates to increments in log(Matings); expontiating the coefficient give the multiple of Matings for a unit change in Age.

```
> elephants.mod1 <- glm(Matings ~ I(Age-27), data=elephants, family=poisson)
> coef(elephants.mod1)

(Intercept) I(Age - 27)
   0.272698    0.068693

> exp(coef(elephants.mod1))

(Intercept) I(Age - 27)
     1.3135      1.0711
```

Thus, at Age 27, these elephants have a predicted 1.31 matings. For each additional year, matings

are expected to be multiplied by 1.07, an increase of 7%. Perhaps the best way to visualize the model predictions are through an effect plot.

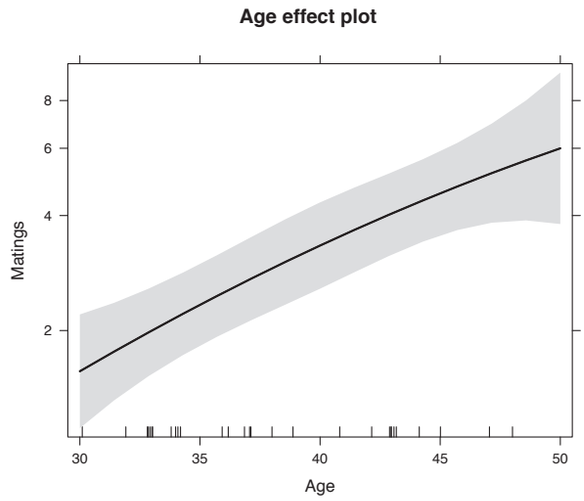
```
> plot(Effect("Age", elephants.mod1,
+              xlevels=list(Age=seq(25,50,5))))
```



(d) Check for nonlinearity in the relationship by using the term `poly(Age,2)` in a new model. What do you conclude?

★ There is little evidence for nonlinearity, as evidenced by a test of the quadratic model in Age, or by a plot of the predicted values under this model.

```
> elephants.mod1q <- glm(Matings ~ poly(Age,2), data=elephants, family=poisson)
> anova(elephants.mod1, elephants.mod1q, test="Chisq")

Analysis of Deviance Table

Model 1: Matings ~ I(Age - 27)
Model 2: Matings ~ poly(Age, 2)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        39       51.0
2        38       50.8  1    0.185     0.67

> plot(allEffects(elephants.mod1q))
```



(e) Assess whether there is any evidence of overdispersion in these data by fitting analogous quasi-Poisson and negative-binomial models.

★ The simple way to assess overdispersion is via `dispersiontest()`, which is not significant here. A more careful answer would fit and compare the overdispersed models suggested.

```
> dispersiontest(elephants.mod1)

Overdispersion test

data:  elephants.mod1
z = 0.496, p-value = 0.31
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
     1.108
```

Exercise 11.2 The data set *quine* in MASS gives data on absenteeism from schools in rural New South Wales, Australia. 146 children were classified by ethnic background (`Eth`), age (`Age`, a factor), `Sex`, and Learner status (`Lrn`), and the number of days absent (`Days`) from school in a particular school year was recorded.

(a) Fit the all main-effects model in the Poisson family and examine the tests of these effects using `summary()` and car::`Anova()`. Are there any terms that should be dropped according to these tests?

★ For such a factorial design, a useful first step is to examine the sample sizes in the cells of this 4-way classification. The table below shows that the design is very unbalanced, and that there are no slow learners (SL) in age F3.

```
> quine.tab <- xtabs(~ Lrn + Age + Sex + Eth, data=quine)
> ftable(Age + Sex ~ Lrn + Eth, data=quine.tab)

        Age F0    F1    F2    F3
        Sex  F  M  F  M  F  M  F  M
Lrn Eth
AL  A        4  5  5  2  1  7  9  7
    N        4  6  6  2  1  7 10  7
SL  A        1  3 10  3  8  4  0  0
    N        1  3 11  7  9  3  0  0
```

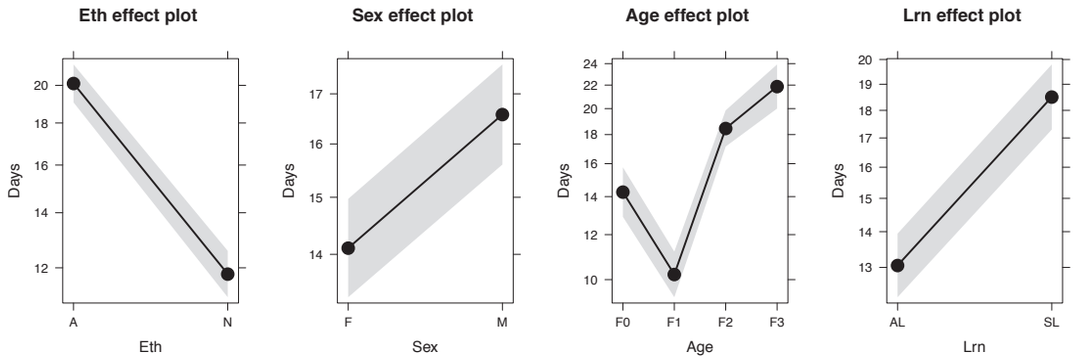
`Age` here corresponds to "Form", or grade in school, arguably an ordered factor, so we make it so. The Poisson model `Anova()` shows that all terms are significant. `summary()` shows that the effects of `Age` have significant linear, quadratic and cubic trends.

```
> data("quine", package="MASS")
> quine$Age <- ordered(quine$Age)
> quine.mod1 <- glm(Days ~ ., data=quine, family=poisson)
> Anova(quine.mod1)

Analysis of Deviance Table (Type II tests)

Response: Days
    LR Chisq Df Pr(>Chisq)
Eth    166.8  1    < 2e-16 ***
Sex     14.4  1    0.00015 ***
Age    168.3  3    < 2e-16 ***
Lrn     45.8  1    1.3e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary(quine.mod1)

Call:
glm(formula = Days ~ ., family = poisson, data = quine)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
 -6.81   -3.06   -1.12    1.82    9.91

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.8033     0.0434   64.55  < 2e-16 ***
EthN         -0.5336     0.0419  -12.74  < 2e-16 ***
SexM          0.1616     0.0425    3.80  0.00015 ***
Age.L         0.4192     0.0471    8.90  < 2e-16 ***
Age.Q         0.2519     0.0497    5.06  4.1e-07 ***
Age.C        -0.3013     0.0410   -7.34  2.1e-13 ***
LrnSL         0.3489     0.0520    6.70  2.0e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2073.5  on 145  degrees of freedom
```

```
Residual deviance: 1696.7  on 139  degrees of freedom
AIC: 2299

Number of Fisher Scoring iterations: 5
```

Although not asked in the question, a careful analysis will try to understand the fitted model, e.g., via an effect plot.

```
> plot(allEffects(quine.mod1), rows=1, cols=4, ci.style='bands')
```



(b) Re-fit this model as a quasi-Poisson model. Is there evidence of overdispersion? Test for overdispersion formally, using `dispersiontest()` from AER (Kleiber and Zeileis, 2015).

★ `dispersiontest()` reveals very substantial overdispersion, with variance approximately $\widehat{\phi} = 13.2$ times the mean under this model.

```
> library(AER)
> dispersiontest(quine.mod1)

Overdispersion test

data:  quine.mod1
z = 5.47, p-value = 2.3e-08
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
     12.53

> quine.mod1q <- glm(Days ~ ., data=quine, family=quasipoisson)
```

(c) Carry out the same significance tests and explain why the results differ from those for the Poisson model.

★ The `Anova()` tests for the quasi-Poisson model `quine.mod1q` show that only ethnicity and age have significant effects, learner status nearly so. Relative to the standard Poisson model, the coefficients are the same, but the standard errors have been adjusted by a factor of $\widehat{\phi}^{-1/2} = 0.28$

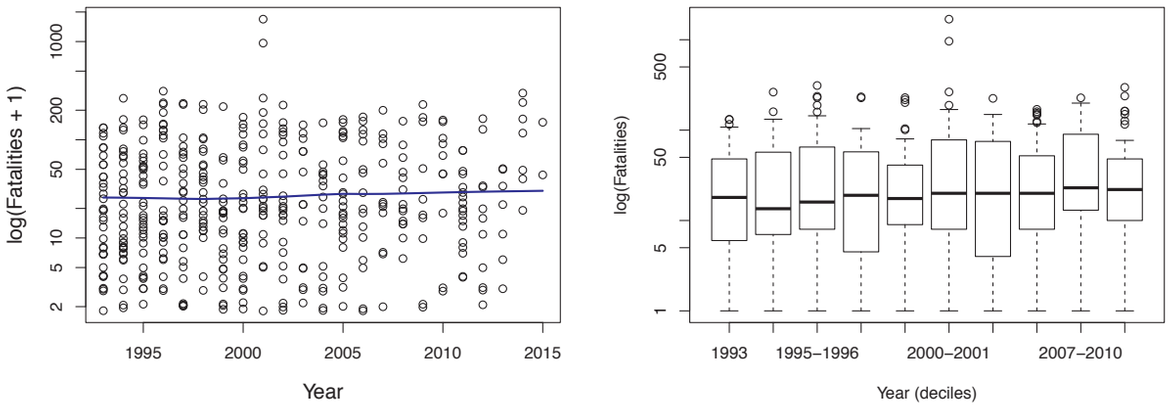
```
> Anova(quine.mod1q)
Analysis of Deviance Table (Type II tests)

Response: Days
    LR Chisq Df Pr(>Chisq)
Eth    12.67  1    0.00037 ***
Sex     1.09  1    0.29560
Age    12.78  3    0.00513 **
Lrn     3.48  1    0.06218 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> coeftest(quine.mod1q)

z test of coefficients:

            Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.803      0.158   17.79  < 2e-16 ***
EthN          -0.534      0.152   -3.51  0.00045 ***
SexM           0.162      0.154    1.05  0.29510
Age.L          0.419      0.171    2.45  0.01418 *
Age.Q          0.252      0.181    1.40  0.16288
Age.C         -0.301      0.149   -2.02  0.04297 *
LrnSL          0.349      0.189    1.85  0.06463 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exercise 11.3 The data set *AirCrash* in vcdExtra was analyzed in Exercise 5.2 and Exercise 6.3 in relation to the Phase of the flight and Cause of the crash. Additional variables include the number of Fatalities and Year. How does Fatalities depend on the other variables?

(a) Use the methods of this chapter to make some exploratory plots relating fatalities to each of the predictors.

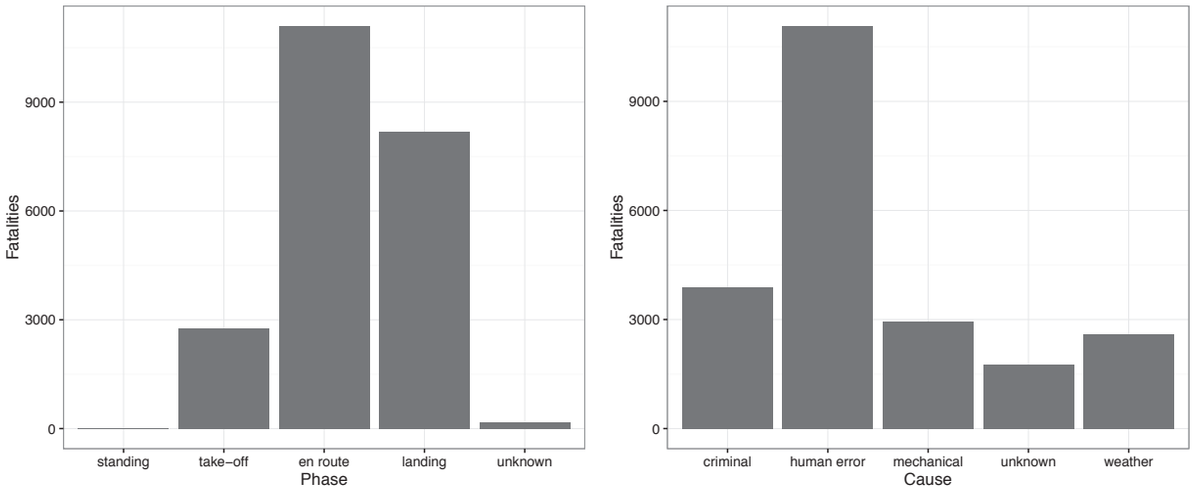
★ Both Phase and Cause are unordered factors, whose levels are ordered lexically. For graphics and analysis it is useful to reorder Phase in temporal order. Plots of the total number fatalities over Year, in either jittered scatterplot or boxplot form show no overall trend. (Other plots, not shown here, indicate that the number of crashes has decreased over time.) As in the text, we plot these on a log scale.

```
> AirCrash$Phase <- factor(AirCrash$Phase,
+                  levels=c("standing", "take-off", "en route", "landing", "unknown"))
> with(AirCrash, {
+      plot(jitter(Fatalities+1) ~ Year, log="y",
+           ylab = "log(Fatalities + 1)", cex.lab=1.25)
+      lines(lowess(Year, Fatalities+1), col="blue", lwd=2)
+                  })
>
> plot(Fatalities ~ cutfac(Year), data=AirCrash, log="y",
+      ylab = "log(Fatalities)", xlab="Year (deciles)")
```



Plots of fatalities against the factors can be done in a variety of forms. Here we use ggplot2 for bar charts. Note the use of geom_bar(aes(weight=Fatalities)), so that the height of each bar is proportional to the number of fatalities; otherwise, it would just show the number of crashes.

```
> ggplot(AirCrash, aes(Phase)) + geom_bar(aes(weight=Fatalities)) +
+       ylab("Fatalities") + theme_bw()
> ggplot(AirCrash, aes(Cause)) + geom_bar(aes(weight=Fatalities)) +
+       ylab("Fatalities") + theme_bw()
```

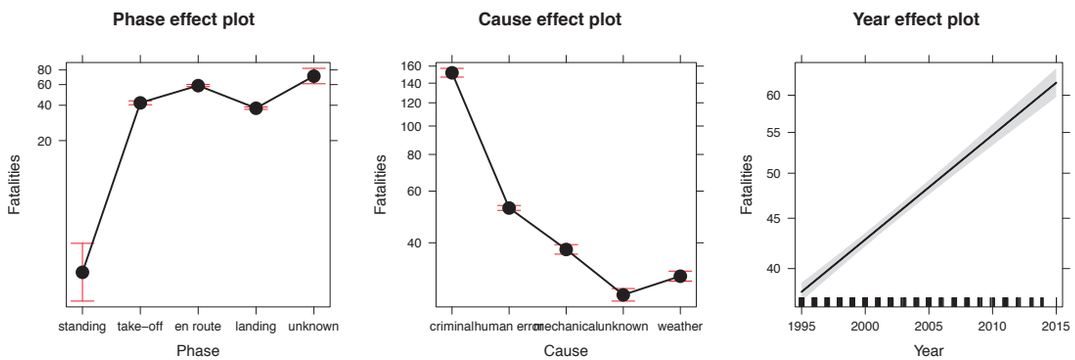


(b) Fit a main effects poisson regression model for `Fatalities`, and make effects plots to visualize the model. Which phases and causes result in the largest number of fatalities?
★

```
> crash.mod1 <- glm(Fatalities ~ Phase + Cause + Year,
+                   data=AirCrash, family=poisson)
> Anova(crash.mod1)

Analysis of Deviance Table (Type II tests)

Response: Fatalities
      LR Chisq Df Pr(>Chisq)
Phase     1586  4     <2e-16 ***
Cause     5004  4     <2e-16 ***
Year       436  1     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the marginal plots for part (a), it appears that most fatalities occur en route or on landing, and the primary cause is human error. However, an effects plot for the main effects model, where other factors are controled in a given panel, tells a different story: Unknown phase rises to the top, and the greatest cause is criminal activity. The plot for `Year` is slightly misleading, because the range is relatively small compared to the other panels. This can be avoided by setting `ylim=log(c(1,180))` in the plot.

```
> plot(allEffects(crash.mod1), rows=1, cols=3)
```



(c) A linear effect of `Year` might not be appropriate for these data. Try using a natural spline term, `ns(Year, df)` to achieve a better, more adequate model.
★ A careful analysis would investigate the tradeoff between goodness-of-fit and parsimony in the choice of degrees of freedom. Here we just illustrate the use of `ns(Year,3)`, allowing 3 df for the `Year` effect. This fits significantly better than the linear model in `Year`.

```
> library(splines)
> crash.mod2 <- glm(Fatalities ~ Phase + Cause + ns(Year,3),
+                   data=AirCrash, family=poisson)
```

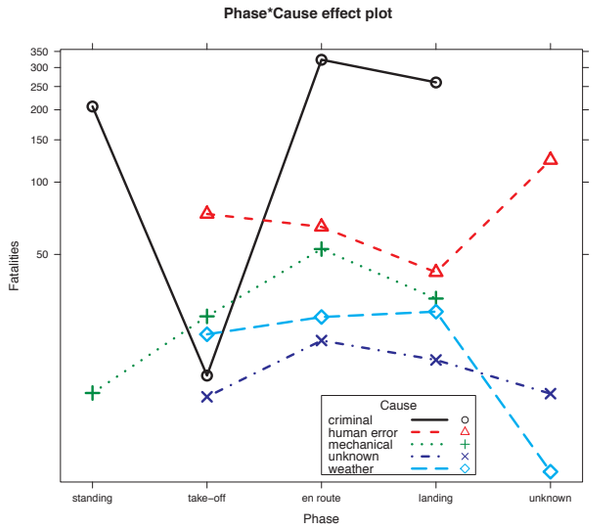
```
> anova(crash.mod1, crash.mod2, test="Chisq")

Analysis of Deviance Table

Model 1: Fatalities ~ Phase + Cause + Year
Model 2: Fatalities ~ Phase + Cause + ns(Year, 3)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       429      30833
2       427      30724  2      109   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(d) Use a model-building tool like `add1()` or MASS::`stepAIC()` to investigate whether there are important two-way interactions among the factors and your chosen effect for `Year`.
★ We don't show a complete analysis here. `add1()` just looks at each of the two-way interaction terms and reports that adding each one, separately, would decrease the deviance and AIC, resulting in a better-fitting model. `stepAIC()` eventually includes all three two-way terms.

```
> add1(crash.mod2, scope= ~.^2, test="Chisq")

Single term additions

Model:
Fatalities ~ Phase + Cause + ns(Year, 3)
                Df Deviance   AIC  LRT Pr(>Chi)
<none>               30724 32862
Phase:Cause     11    29959 32119  765   <2e-16 ***
Phase:ns(Year, 3) 11    30395 32554  329   <2e-16 ***
Cause:ns(Year, 3) 12    28626 30788 2098   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> # refit the stepAIC final model
> crash.modAIC <- update(crash.mod2, .~. + Phase:Cause + Phase:ns(Year, 3) + Cause:ns(Year, 3))
> Anova(crash.modAIC)

Analysis of Deviance Table (Type II tests)

Response: Fatalities
                LR Chisq Df Pr(>Chisq)
Phase                703  4     <2e-16 ***
Cause               4654  4     <2e-16 ***
ns(Year, 3)          480  3     <2e-16 ***
Phase:Cause          426  8     <2e-16 ***
Phase:ns(Year, 3)    194  8     <2e-16 ***
Cause:ns(Year, 3)   2049 12     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(e) Visualize and interpret your final model and write a brief summary to answer the question posed.
★ The number of fatalities in air crashes depends in a complex way on the combinations of `Phase`, `Cause` and `Year`. We illustrate this below with an effect plot for the interaction of `Phase` and `Cause`, the interpretation of which just entails trying to describe the patterns in the plot.

```
> plot(Effect(c("Phase", "Cause"), crash.modAIC),
+      multiline=TRUE, lwd=3, key.args=list(x=0.5, y=0.2))
```



Exercise 11.4 Male double-crested cormorants use advertising behavior to attract females for breeding. The `Cormorants` data set in vcdExtra gives some results from a study by Meagan Mc Rae (2015) on counts of advertising males observed two or three times a week at six stations in a tree-nesting colony for an entire breeding season. The number of advertising birds was counted and these observations were classified by characteristics of the trees and nests. The goal was to determine how this behavior varies temporally over the season and spatially over observation stations, as well as with characteristics of nesting sites. The response variable is `count` and other predictors are shown below. See `help(Cormorants, package="vcdExtra")` for further details.

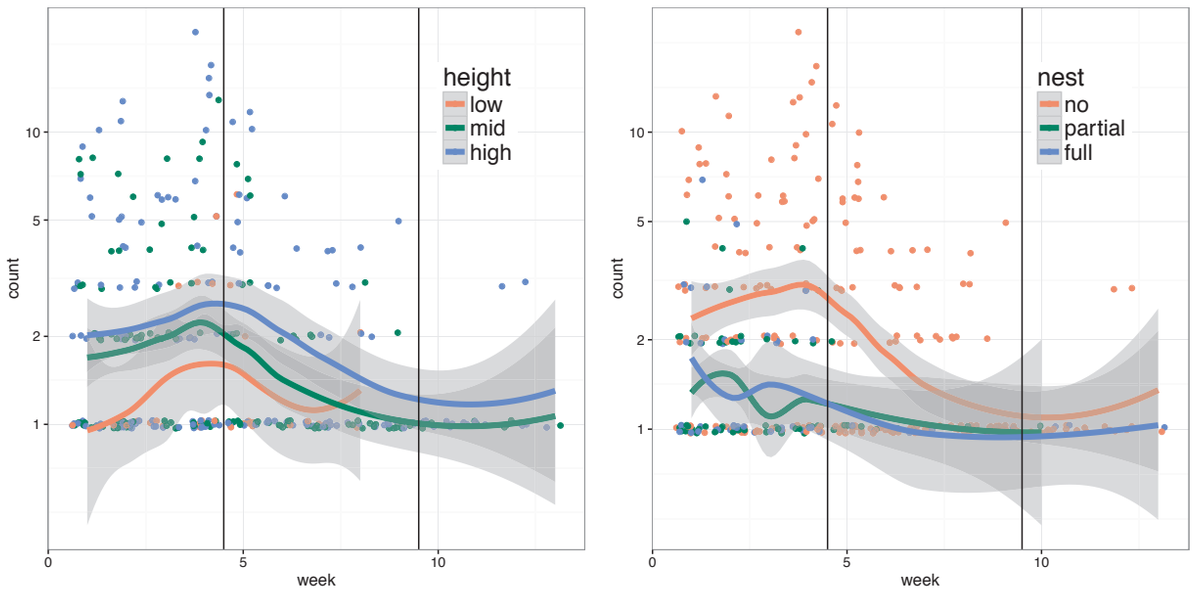
```
> data("Cormorants", package = "vcdExtra")
> car::some(Cormorants)

      category week station    nest height  density tree_health count
4          Pre    1      C1 partial   high      few     healthy     2
34         Pre    1      B1      no   high      few        dead     1
143        Pre    3      B2      no    mid      few        dead     1
155 Incubation    4      C2      no    mid      few        dead    13
160 Incubation    4      C2 partial   high      few     healthy     1
175 Incubation    4      C4 partial   high      few        dead     3
240 Incubation    6      C2      no    mid moderate        dead     1
250 Incubation    6      C4      no   high      few        dead     1
314 Incubation    9      B2      no    mid      few     healthy     2
340       <NA>   12      B1      no   high      few        dead     1
```

(a) Using the methods illustrated in this chapter, make some exploratory plots of the number of advertising birds against week in the breeding season, perhaps stratified by another predictor, like tree `height`, nest condition, or observation `station`. To see anything reasonable, you should plot `count` on a log (or square root) scale, jitter the points, and add smoothed curves. The variable `category` breaks the weeks into portions of the breeding season, so adding vertical lines separating those will be helpful for interpretation.

★ Here we use ggplot2 to plot `count` against `week`, stratified by `height` and `nest`. The plot for `height` shows that counts of advertising birds increase with height in the tree and rise over week at the end of the pre-breeding portion of the season.

```
> my_theme <- theme_bw() +
+             theme(legend.position = c(0.8, 0.8),
+                   legend.title = element_text(size=18),
+                   legend.text = element_text(size=16))
>
> ggplot(Cormorants, aes(week, count, color=height)) +
+   geom_jitter() +
+         stat_smooth(method="loess", size=2) +
+         scale_y_log10(breaks=c(1,2,5,10)) +
+         geom_vline(xintercept=c(4.5, 9.5)) +
+         my_theme
>
> ggplot(Cormorants, aes(week, count, color=nest)) +
+   geom_jitter() +
+   stat_smooth(method="loess", size=2) +
```

```
+    scale_y_log10(breaks=c(1,2,5,10)) +
+    geom_vline(xintercept=c(4.5, 9.5)) +
+    my_theme
```

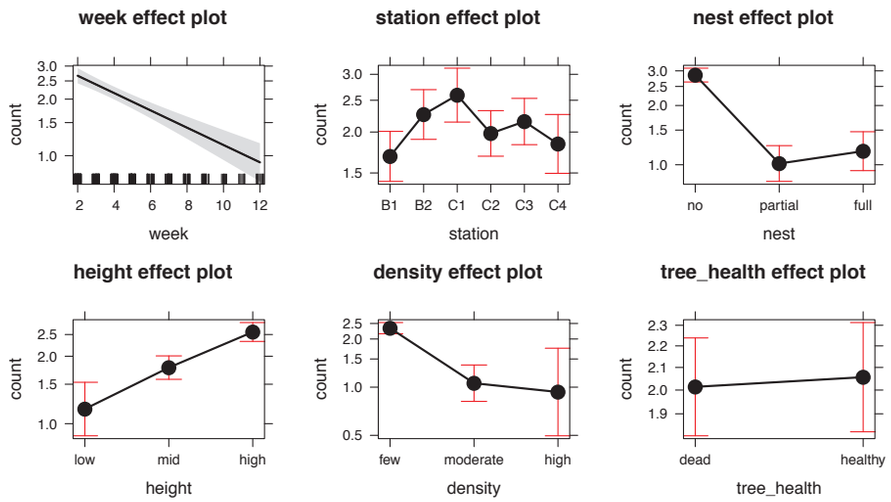


(b) Fit a main-effects Poisson GLM to these data and test the terms using `Anova()` from the car package.
★ All terms except `tree_health` show significant main effects.

```
> fit1 <-glm(count ~ week + station + nest + height + density + tree_health, data=Cormorants,
+     family =  poisson)
> library(car)
> Anova(fit1)

Analysis of Deviance Table (Type II tests)

Response: count
            LR Chisq Df Pr(>Chisq)
week            62.8  1    2.3e-15 ***
station         14.1  5      0.015 *
nest           137.8  2    < 2e-16 ***
height          48.0  2    3.8e-11 ***
density         47.3  2    5.5e-11 ***
tree_health      0.1  1      0.792
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(c) Interpret this model using an effects plot.
★ According to the linear model, counts (a) decrease over weeks in the seasons; (b) are greatest with no nests; (c) increase with height in the tree and (d) are greatest when the density is low.

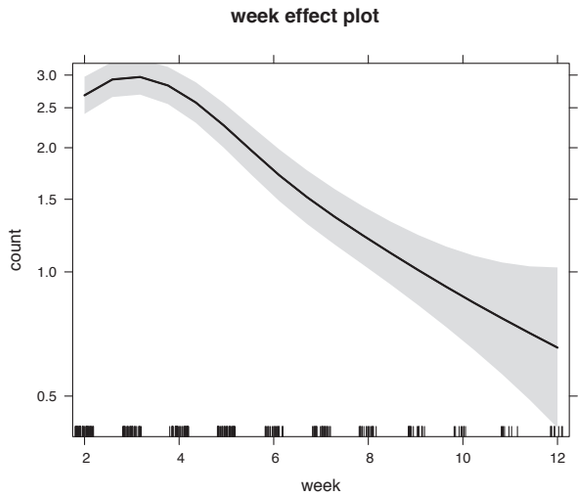
```
> library(effects)
> plot(allEffects(fit1))
```



(d) Investigate whether the effect of `week` should be treated as linear in the model. You could try using a polynomial term like `poly(week, degree)` or perhaps better, using a natural spline term like `ns(week, df)` from the splines package.

★ Here we drop `tree_health` and use a natural spline with 3 degrees of freedom for `week` to allow a moderate departure from linearity, roughly comparable to a cubic term `poly(week, 3)` in complexity. The effect plot for week shows a pronounced rise over the early weeks, followed by a steady decline over the rest of the season.

```
> library(splines)
> fit2 <-glm(count ~ ns(week,3) + station + nest + height + density , data=Cormorants,
+     family = poisson)
> Anova(fit2)

Analysis of Deviance Table (Type II tests)

Response: count
            LR Chisq Df Pr(>Chisq)
ns(week, 3)     98.8  3    < 2e-16 ***
station         17.5  5     0.0036 **
nest           129.3  2    < 2e-16 ***
height          59.2  2    1.4e-13 ***
density         48.7  2    2.6e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> plot(Effect("week", fit2))
```



A more thorough analysis would investigate whether there are important interactions among the model terms. The model below checks whether there are interactions of week with any of the other factors.

The conclusion is no.

```
> fit3 <- update(fit2, . ~ . + week * (station + nest + height + density))
> Anova(fit3)

Analysis of Deviance Table (Type II tests)

Response: count
              LR Chisq Df Pr(>Chisq)
ns(week, 3)       34.3  2    3.6e-08 ***
station           16.7  5     0.0051 **
nest             129.9  2    < 2e-16 ***
height            59.1  2    1.5e-13 ***
density           48.4  2    3.0e-11 ***
week                    0
station:week       1.2  5     0.9449
nest:week          0.0  2     0.9771
height:week        2.7  2     0.2647
density:week       3.5  2     0.1742
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(e) Test this model for overdispersion, using either a `quasipoisson` family or `dispersiontest()` in AER.
★ Using `dispersiontest()` on the `fit1` and `fit2` models shows significant overdispersion for the former, but not for the latter. This illustrates that the test for overdispersion assumes that the model is correctly specified.

```
> require(AER)
> dispersiontest(fit1)

Overdispersion test

data:  fit1
z = 1.79, p-value = 0.036
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
    1.3713

> dispersiontest(fit2)

Overdispersion test

data:  fit2
z = 1.19, p-value = 0.12
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
    1.1782
```

Exercise 11.5 For the *CodParasites* data, recode the `area` variable as an ordered factor as suggested in footnote 13. Test the hypotheses that prevalence and intensity of cod parasites is linearly related to area.

★ We illustrate this only for fitting the negative binomial model to the `intensity` variable, the count of parasites. The four areas A1–A4 are ordered from East to West, so the easiest way to do this test of linearity is to create a numeric equivalent.

```
> data("CodParasites", package = "countreg")
>
> ## omit NAs in response
> CodParasites <- subset(CodParasites, !is.na(intensity))
>
> # make a numeric variable from area
> CodParasites$areaEW <- as.numeric(CodParasites$area)
```

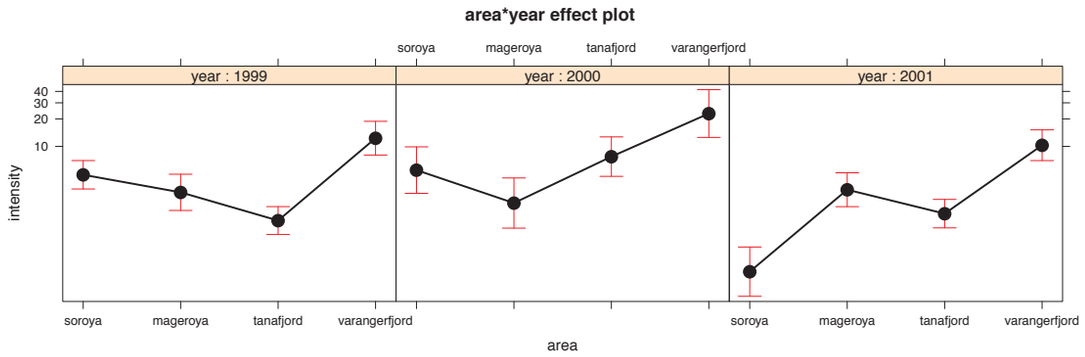
Then, we fit negative binomial models with `area` as a factor, and `areaEW` as a numeric variable.

```
> library(MASS)
> cp_nb     <- glm.nb(intensity ~ length + area * year, data = CodParasites)
> cp_nb_lin <- glm.nb(intensity ~ length + areaEW * year, data = CodParasites)
> Anova(cp_nb_lin)

Analysis of Deviance Table (Type II tests)

Response: intensity
```

```
            LR Chisq Df Pr(>Chisq)
length          23.2  1    1.5e-06 ***
areaEW          44.1  1    3.2e-11 ***
year            25.2  2    3.3e-06 ***
areaEW:year     17.2  2    0.00019 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(cp_nb_lin, cp_nb, test="Chisq")

Likelihood ratio tests of Negative Binomial Models

Response: intensity
                    Model   theta Resid. df    2 x log-lik.   Test    df LR stat.
1 length + areaEW * year 0.19085      1184         -5088.5
2   length + area * year 0.21531      1178         -5002.7 1 vs 2     6   85.852
     Pr(Chi)
1
2 2.2204e-16
```

The model with area as a factor is significantly better than the model with only a linear effect. Nonetheless, an effect plot shows that the intensity of parasites is greatest in the most eastern area, Varangerfjord (A4) in all three years, lending some credance to the "Russian hypothesis."

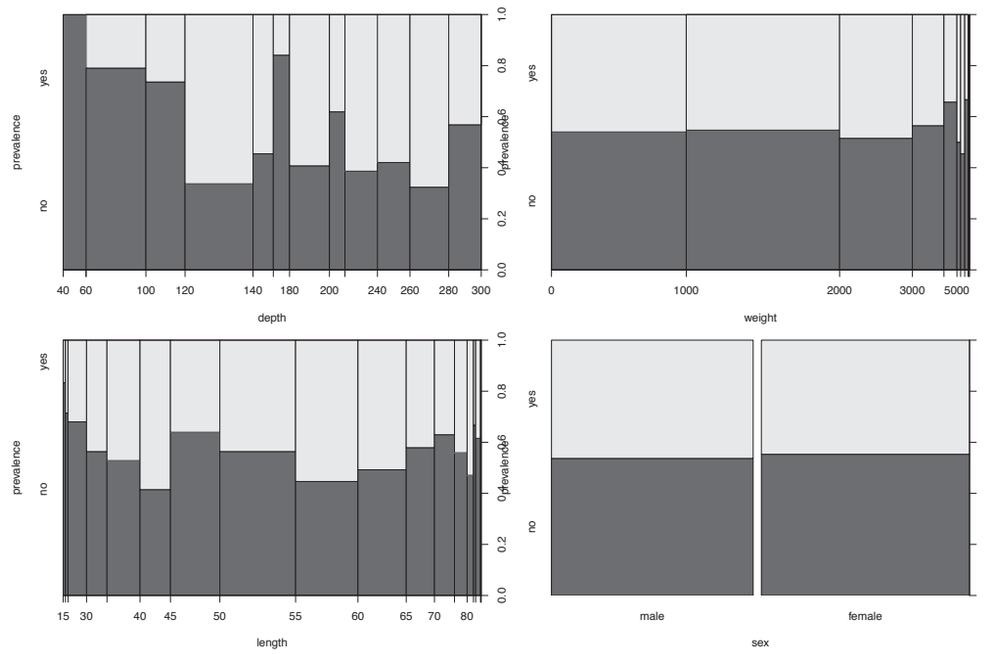
```
> plot(Effect(c("area", "year"), cp_nb), layout=c(3,1))
```



Exercise 11.6 In Example 11.10, we ignored other potential predictors in the *CodParasites* data: `depth`, `weight`, `length`, `sex`, `stage`, and `age`. Use some of the graphical methods shown in this case study to assess whether any of these are related to prevalence and intensity.

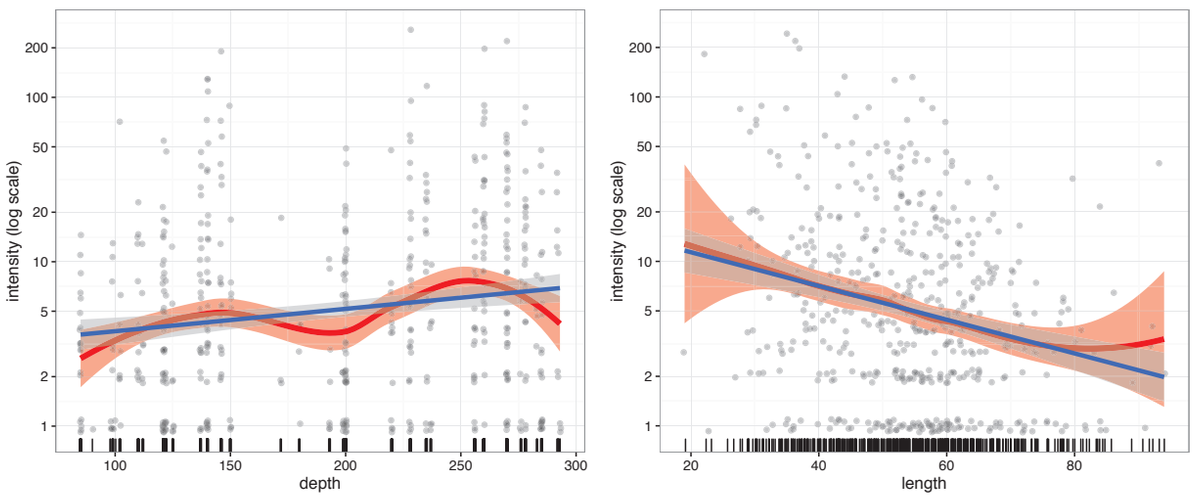
★ Prevalence, a factor corresponding to `intensity>0`, can be studied by simple plots, that are essentially spineplots. In these plots prevalence seems to vary with `depth` and `length`, but not with `weight` and `sex`.

```
> CodParasites <- subset(CodParasites, sex !=0)
> CodParasites$sex <- factor(CodParasites$sex, labels=c("male", "female"))
>
> op <-par(mfrow = c(2, 2), mar=c(4,4,1,1)+.1)
> plot(prevalence ~ depth, data = CodParasites)
> plot(prevalence ~ weight, data = CodParasites)
> plot(prevalence ~ length, data = CodParasites)
> plot(prevalence ~ sex, data = CodParasites)
> par(op)
```



For intensity, the most informative plots use ggplot2, in the style of Figure 11.20. That is, plot jittered points, use a log scale for intensity, and overlay smoothed and linear fits. Intensity increases slightly with depth and decreases with the length of the fish.

```
> CPpos <- subset(CodParasites, intensity>0 & !is.na(length))
> ggplot(CPpos, aes(x=depth, y=intensity)) +
+         geom_jitter(position=position_jitter(height=.1), alpha=0.25) +
+         geom_rug(position='jitter', sides='b') +
+         scale_y_log10(breaks=c(1,2,5,10,20,50,100, 200)) +
+         stat_smooth(method="loess", color="red", fill="red", size=2) +
+         stat_smooth(method="lm", size=1.5) + theme_bw() +
+         labs(y='intensity (log scale)')
>
> ggplot(CPpos, aes(x=length, y=intensity)) +
+         geom_jitter(position=position_jitter(height=.1), alpha=0.25) +
+         geom_rug(position='jitter', sides='b') +
+         scale_y_log10(breaks=c(1,2,5,10,20,50,100, 200)) +
+         stat_smooth(method="loess", color="red", fill="red", size=2) +
+         stat_smooth(method="lm", size=1.5) + theme_bw() +
+         labs(y='intensity (log scale)')
```



Exercise 11.7 The analysis of the *PhdPubs* data in the examples in this chapter were purposely left incomplete,

going only as far as the negative binomial model.

(a) Fit the zero-inflated and hurdle models to this data set, considering whether the count component should be Poisson or negative-binomial, and whether the zero model should use all predictors or only a subset. Describe your conclusions from this analysis in a few sentences.

★ Here we fit the hurdle and zero-inflated models, both poisson and negative-binomial, using all predictors. By AIC, model `phd.znb` is best, but by BIC, the regular negative-binomial fares best.

```
> library(MASS)
> library(countreg)
> data("PhdPubs", package="vcdExtra")
>
> phd.nbin  <- glm.nb(articles ~ ., data=PhdPubs)
>   # hurdle and zero-inflated
> phd.hp    <- hurdle(articles ~ ., data=PhdPubs, dist = "poisson")
> phd.hnb   <- hurdle(articles ~ ., data=PhdPubs, dist = "negbin")
> phd.zp    <- zeroinfl(articles ~ ., data=PhdPubs, dist = "poisson")
> phd.znb   <- zeroinfl(articles ~ ., data=PhdPubs, dist = "negbin")
>
> LRstats(phd.nbin, phd.hp, phd.hnb, phd.zp, phd.znb,  sortby="BIC")

Likelihood summary table:
          AIC  BIC LR Chisq  Df Pr(>Chisq)
phd.hp   3235 3292     3211 903     <2e-16 ***
phd.zp   3234 3291     3210 903     <2e-16 ***
phd.hnb  3131 3194     3105 902     <2e-16 ***
phd.znb  3126 3188     3100 902     <2e-16 ***
phd.nbin 3135 3169     3121 909     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Examine the zero-inflated negative-binomial model:

```
> summary(phd.znb)

Call:
zeroinfl(formula = articles ~ ., data = PhdPubs, dist = "negbin")

Pearson residuals:
   Min     1Q Median     3Q    Max
-1.281 -0.760 -0.286  0.444  6.507

Count model coefficients (negbin with log link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.3655     0.1399    2.61   0.0090 **
female       -0.1946     0.0758   -2.57   0.0102 *
married       0.1015     0.0843    1.20   0.2288
kid5         -0.1516     0.0542   -2.80   0.0052 **
phdprestige   0.0156     0.0345    0.45   0.6510
mentor        0.0244     0.0035    6.97  3.1e-12 ***
Log(theta)    0.9766     0.1354    7.21  5.4e-13 ***

Zero-inflation model coefficients (binomial with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.2943     1.2827   -0.23   0.8185
female        0.6580     0.8557    0.77   0.4419
married      -1.4741     0.9176   -1.61   0.1082
kid5          0.6241     0.4410    1.42   0.1570
phdprestige  -0.0113     0.2888   -0.04   0.9689
mentor       -0.8765     0.3152   -2.78   0.0054 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Theta = 2.655
Number of iterations in BFGS optimization: 42
Log-likelihood: -1.55e+03 on 13 Df
```

This suggests a simpler model in which `mentor` is the only variable affecting the zero counts. This model is preferable to the full model by both AIC and BIC.

```
> phd.znb1   <- zeroinfl(articles ~ .| mentor, data=PhdPubs, dist = "negbin")
> LRstats(phd.znb, phd.znb1)

Likelihood summary table:
          AIC  BIC LR Chisq  Df Pr(>Chisq)
phd.znb  3126 3188     3100 902     <2e-16 ***
phd.znb1 3124 3168     3106 906     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

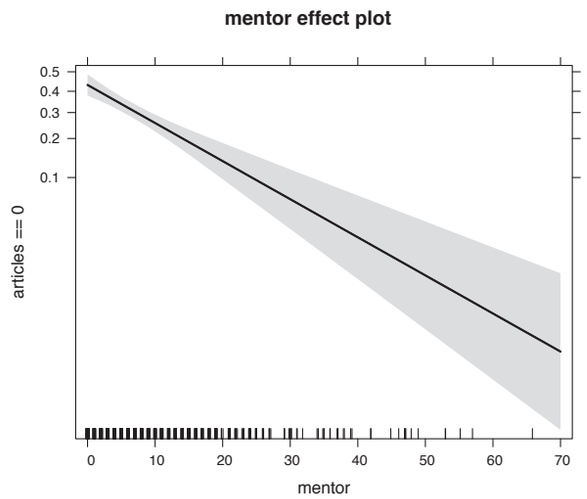
(b) Using the methods illustrated in this chapter, create some graphs summarizing the predicted counts and probabilities of zero counts for one of these models.

★ As described in the text, effect plots for analogs of hurdle and zero-inflated models can only be made by fitting and graphing separately the models for the zero and positive counts. We use a logistic regression for the zero counts.

```
> phd.zero <-  glm(articles==0 ~., data=PhdPubs, family = binomial)
> Anova(phd.zero)

Analysis of Deviance Table (Type II tests)

Response: articles == 0
           LR Chisq Df Pr(>Chisq)
female          2.5  1      0.115
married         3.3  1      0.069 .
kid5            6.6  1      0.010 *
phdprestige     0.3  1      0.572
mentor         51.2  1    8.4e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> phd.nzero <- glm.nb(articles ~ ., data=PhdPubs, subset = articles > 0)
> Anova(phd.nzero)

Analysis of Deviance Table (Type II tests)

Response: articles
           LR Chisq Df Pr(>Chisq)
female          7.2  1     0.0071 **
married         1.1  1     0.3043
kid5            5.0  1     0.0256 *
phdprestige     0.0  1     0.8953
mentor         35.8  1    2.2e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

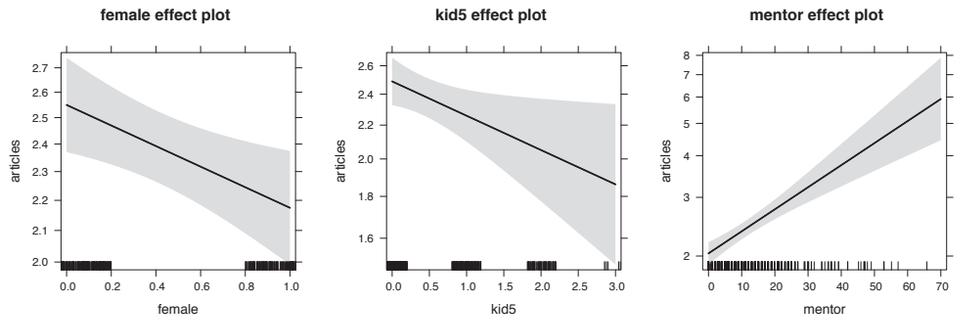
The effect of `mentor` is dramatic: as the number of publications by the mentor increases, the probability that a student will publish zero articles declines markedly.

```
> plot(Effect("mentor", phd.zero))
```



For the positive count model, we plot only the significant effects of `female`, `kids5`, and `mentor`. All have clear interpretations: the number of articles published is lower for females, decreases with the number of young children, and increases with the number of articles published by the mentor.

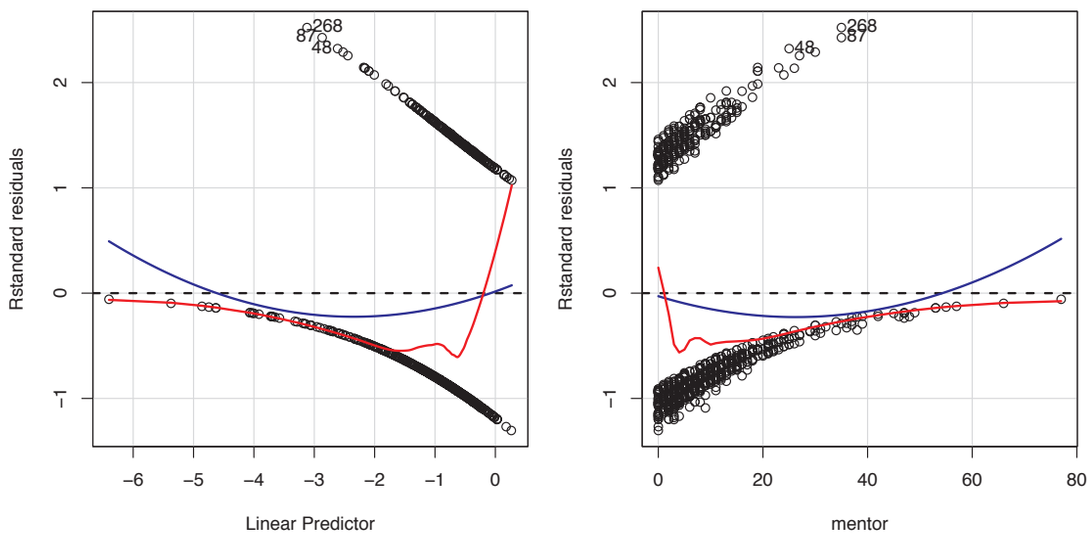
```
> plot(allEffects(phd.nzero)[c(1,3,5)], rows=1, cols=3)
```



(c) For your chosen model, use some of the diagnostic plots of residuals and other measures shown in Section 11.6 to determine if your model solves any of the problems noted in Example 11.17 and Example 11.18, and whether there are any problems that remain.

★ Unfortunately, the best graphic methods, from the car package are unavailable for hurdle and zero-inflated models. A suitable alternative is to use diagnostic plots for separate models for the zero (`phd.zero`) and positive counts (`phd.nzero`). Plots for the zero count component are not particularly remarkable for a logistic regression model.

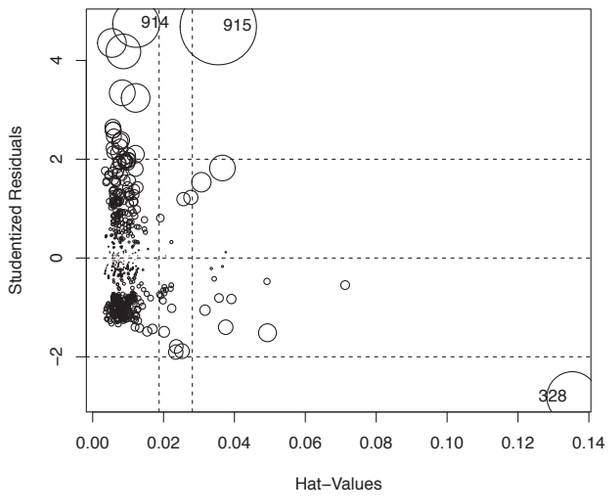
```
> op <- par(mfrow=c(1,2), mar=c(4,4,1,1)+.1)
> residualPlot(phd.zero, type="rstandard",
+              quadratic=TRUE, col.smooth="red", col.quad="blue",  id.n=3)
>
> residualPlot(phd.zero, "mentor", type="rstandard",
+              quadratic=TRUE, col.smooth="red", col.quad="blue",  id.n=3)
> par(op)
```



Similar plots for the positive count model (`phd.n zero`) would be done similarly, and are not particularly remarkable, except that they nominate some large positive residuals as unusual.

An influence plot suggests that three cases should be given further scrutiny.

```
> influencePlot(phd.nzero)

    StudRes      Hat    CookD
328 -2.8149 0.135304 0.099324
914  4.7390 0.012247 0.084424
915  4.6849 0.035444 0.224707
```



Exercise 11.8 In Example 11.19 we used a simple analysis of $\log(\boldsymbol{y} + 1)$ for the multivariate responses in the *NMES1988* data using a classical MLM (Eqn. (11.16)) as a rough approximation of a multivariate Poisson model. The HE plot in Figure 11.40 was given as a visual summary, but did not show the data. Examine why the MLM is not appropriate statistically for these data, as follows:

(a) Calculate residuals for the model `nmes.mlm` using

```
> resids <- residuals(nmes.mlm, type="deviance")
```

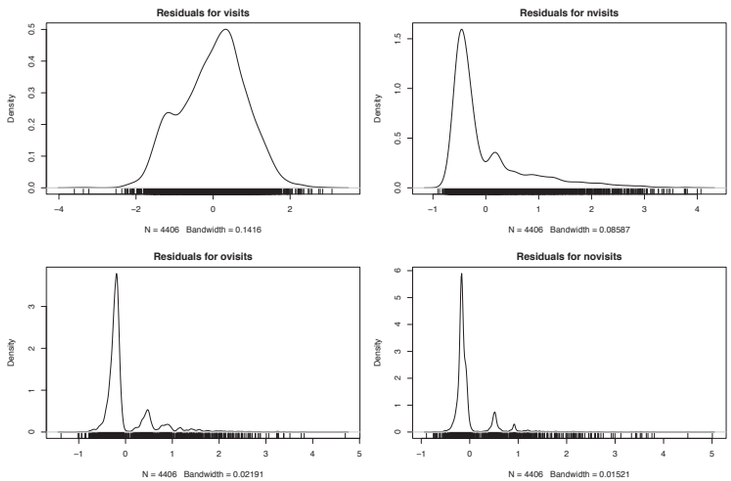
★ This just copies the code from the text to fit the model and calculate the residuals:

```
> data("NMES1988", package="AER")
> nmes2 <- NMES1988[, c(1:4, 6:8, 13, 15, 18)]
>
> clog <- function(x) log(x+1)
> nmes.mlm <- lm(clog(cbind(visits, nvisits, ovisits, novisits)) ~ ., data=nmes2)
> resids <- residuals(nmes.mlm, type="deviance")
```

(b) Make univariate density plots of these residuals to show their univariate distributions. These should be approximately normal under the MLM. What do you conclude?

★ There are a variety of ways to produce density plots in R, but the simplest for this question is just to use `density()` with default settings. The residuals for `visits` are reasonably symmetric. All the others are quite positively skewed.

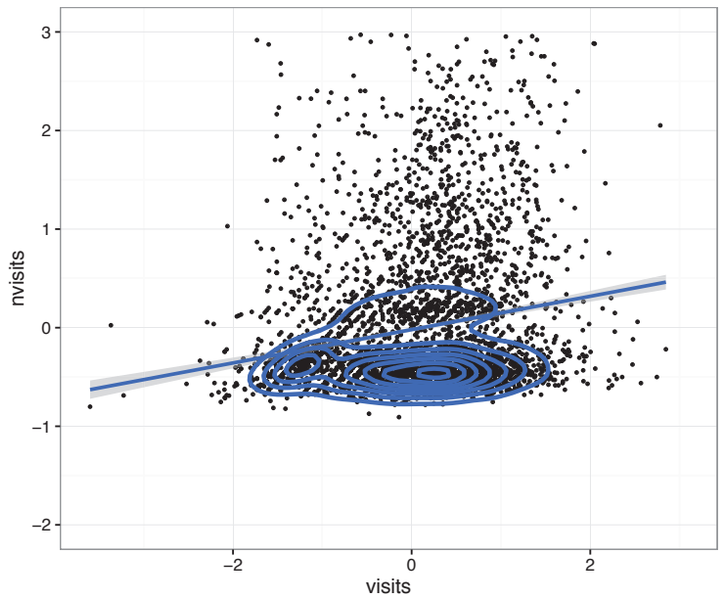
```
> op <- par(mfrow=c(2,2), mar=c(5,4,2,1)+0.1)
> plot(density(resids[,1]), main = "Residuals for visits")
> rug(resids[,1])
>
> plot(density(resids[,2]), main = "Residuals for nvisits")
> rug(resids[,2])
>
> plot(density(resids[,3]), main = "Residuals for ovisits")
> rug(resids[,3])
>
> plot(density(resids[,4]), main = "Residuals for novisits")
> rug(resids[,4])
> par(op)
```



(c) Make some bivariate plots of these residuals. Under the MLM, each should be bivariate normal with elliptical contours and linear regressions. Add 2D density contours (`kde2d()`, or `geom_density2d()` in ggplot2) and some smoothed curve. What do you conclude?

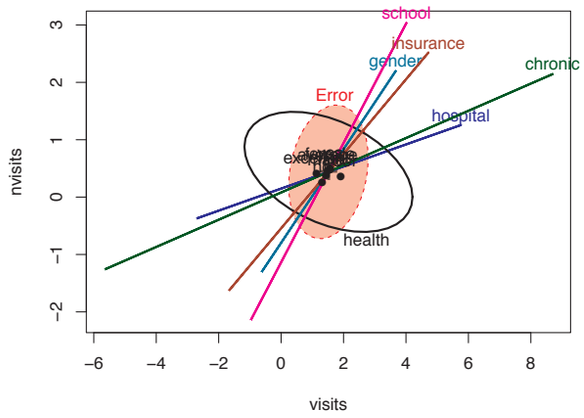
★ Here is just one example for the first two variables, `visits`, and `nvisits`. The bivariate distribution is very far from bivariate normal, largely because `nvisits` is so positively skewed. All other pairs of the residuals would be worse.

```
> ggplot(as.data.frame(resids), aes(x=visits, y=nvisits)) +
+         geom_point(size=0.6) + ylim(c(-2, 3)) +
+         stat_smooth(method="lm") +
+         geom_density2d(size=1.25) + theme_bw()
```



The overall conclusion is that, while Figure 11.40 in the text gives a useful overall summary of the relationship among the multivariate resonses, it should only be considered as a rough approximation. For comparison, the plot above corresponds to the error ellipse in the (2,1) panel of Figure 11.40, shown in red in the plot below.

```
> library(heplots)
> heplot(nmes.mlm, variables=1:2, fill=c(TRUE, FALSE))
```



References

Agresti, A. (2013). *Categorical Data Analysis*. Wiley Series in Probability and Statistics. New York: Wiley-Interscience [John Wiley & Sons], 3rd edn.

Agresti, A. and Winner, L. (1997). Evaluating agreement and disagreement among movie reviewers. *Chance*, 10(2), 10–14.

Bertin, J. (1983). *Semiology of Graphics*. Madison, WI: University of Wisconsin Press. (trans. W. Berg).

Bouchet-Valat, M. (2015). *logmult: Log-Multiplicative Models, Including Association Models*. R package version 0.6.1.

Fisher, R. A. (1940). The precision of discriminant functions. *Annals of Eugenics*, 10, 422–429.

Fox, J. and Weisberg, S. (2015). *car: Companion to Applied Regression*. R package version 2.0-25/r421.

Friendly, M. (2014a). *HistData: Data sets from the history of statistics and data visualization*. R package version 0.7-5.

Friendly, M. (2014b). *Lahman: Sean Lahman's Baseball Database*. R package version 3.0-1.

Friendly, M. (2015). *vcdExtra: vcd Extensions and Additions*. R package version 0.6-7.

Friendly, M. and Kwan, E. (2003). Effect ordering for data displays. *Computational Statistics and Data Analysis*, 43(4), 509–539.

Geissler, A. (1889). Beitrage zur frage des geschlechts verhaltnisses der geborenen. *Z. K. Sachsischen Statistischen Bureaus*, 35(1), n.p.

Grayson, D. K. (1990). Donner party deaths: A demographic assessment. *Journal of Anthropological Research*, 46(3), 223–242.

Harrell, Jr., F. E. (2015). *rms: Regression Modeling Strategies*. R package version 4.3-0.

Jansen, J. (1990). On the statistical analysis of ordinal data when extravariation is present. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 39(1), 75–84.

Kleiber, C. and Zeileis, A. (2015). *AER: Applied Econometrics with R*. R package version 1.2-3.

Mc Rae, M. (2015). *Spatial, Habitat and Frequency Changes in Double-crested Cormorant Advertising Display in a Tree-nesting Colony*. Masters project, environmental studies, York University.

Meyer, D., Zeileis, A., and Hornik, K. (2015). *vcd: Visualizing Categorical Data*. R package version 1.3-3.

Mosteller, F. and Wallace, D. L. (1963). Inference in an authorship problem. *Journal of the American Statistical Association*, 58(302), 275–309.

Poole, J. H. (1989). Mate guarding, reproductive success and female choice in African elephants. *Animal Behavior*, 37, 842–849.

Ramsey, F. L. and Schafer, D. W. (2002). *The Statistical Sleuth: A Course in Methods of Data Analysis*. Belmont, CA: Duxbury, 2nd edn.

Ramsey, F. L., Schafer, D. W., Sifneos, J., and Turlach, B. A. (2012). *Sleuth2: Data sets from Ramsey and Schafer's* Statistical Sleuth *(2nd ed)*. R package version 1.0-7.

Ripley, B. (2015). *MASS: Support Functions and Datasets for Venables and Ripley's MASS*. R package version 7.3-40.

Turner, H. and Firth, D. (2014). *gnm: Generalized Nonlinear Models*. R package version 1.0-7.

Wickham, H. and Chang, W. (2015). *ggplot2: An Implementation of the Grammar of Graphics*. R package version 1.0.1.

Wong, R. S.-K. (2001). Multidimensional association models: A multilinear approach. *Sociological Methods and Research*, 30(2), 197–240.

Wong, R. S.-K. (2010). *Association Models*. Quantitative Applications in the Social Sciences. Los Angeles: SAGE Publications.

Wright, K. (2015). *agridat: Agricultural Datasets*. R package version 1.11.

Yee, T. W. (2015). *VGAM: Vector Generalized Linear and Additive Models*. R package version 0.9-7.

Zeileis, A. and Kleiber, C. (2014). *countreg: Count Data Regression*. R package version 0.1-2/r88.

Index