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
```

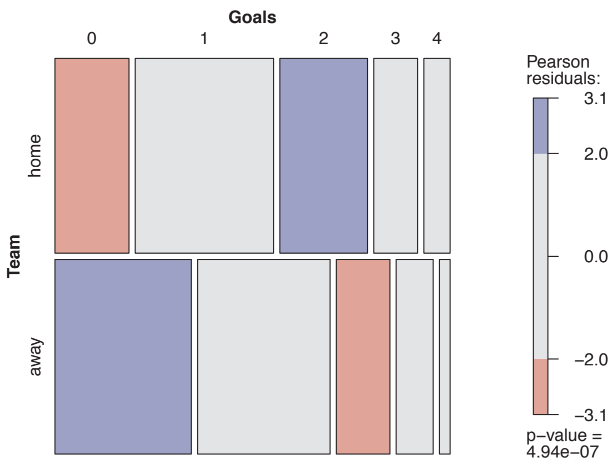
(d) Comment on the distribution of the numbers of home-team and away-team goals. Is there any evidence that home teams score more goals on average?
★ You could find the mean number of goals, weighted by their marginal frequencies. On average, home teams score about 0.4 more goals.

```
> 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.
★

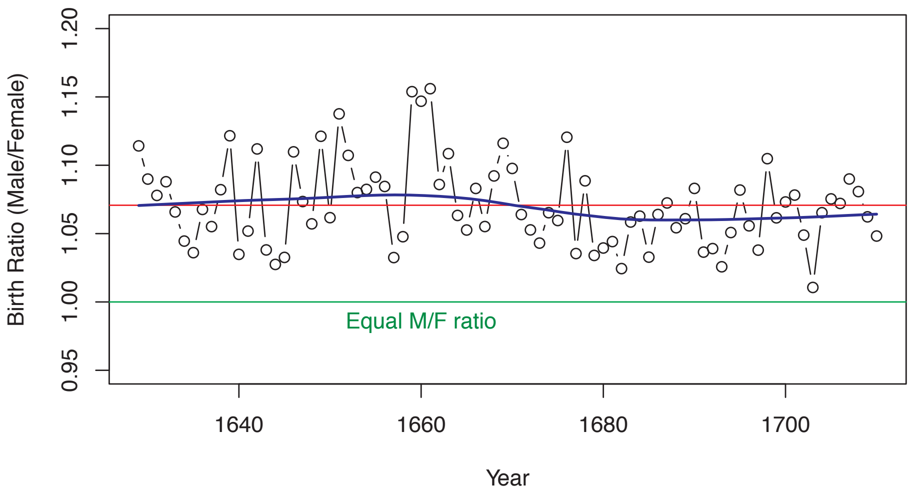
```
> library(xtable)
> va.xtab <- xtable(va.tabm)
> print(va.xtab, type="html")
```

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)
+ })
```