DEPARTMENT OF STATISTICS

Course STATS 330: Advanced Statistical Modelling

Tutorial Sheet 10: October 14, 2010

This tutorial is designed to give you practice in analysing a three-dimensional contingency table. In particular we calculate some odds ratios, and look at the concept of "association reversal", which will be discussed soon in class. Several R tricks useful for manipulating tables are discussed.

In this tutorial you will investigate a data set connected with a sex discrimination case in the US. In 1973 there were 12,763 applications for admission to the graduate school of the University of California at Berkeley. Of the applications by males, 44% were successful; but of the applications by females, only 35% were successful. This difference led to allegations of sex bias in the admissions process.

The data file **`berkeley.txt`** contains 1973 admissions data for the six largest academic departments, designated A, B, C, D, E and F. The variables in the data file are

`Dept`: the department, coded as 1=A, 2=B, 3=C, 4=D, 5=E and 6=F.
`Gender`: the sex of the applicant (M/F)
`Admit:` whether or not the applicant was admitted (0=No, 1=Yes)
`Count`: the number of individuals having the same department, sex and admission status.

Each line of the file also contains an ID number.

Task 1.
Read in the data, and create a data frame **`berkeley.df`**. The data file has variable names and ID numbers, so if you omit **`header = T`** it will read in correctly (see the documentation for **`read.table`**). You will need to turn the numerically coded variables **`Dept`** and **`Admit`** into factors.

Sample code:

```
temp.df<-read.table(file.choose())
berkeley.df<-data.frame(Dept=factor(temp.df[,1],labels=LETTERS[1:6]),
Gender=temp.df[,2],
Admit=factor(temp.df[,3],labels=c("No","Yes")),
Count=temp.df[,4])
```

Task 2.
Make a 3-dimensional contingency table from the data frame.

This can be done by hand. Alternatively, we can treat the table as an "array" in R (this is a generalization of a matrix to more dimensions than 2)

```
> attach(berkeley.df)
> contin.table<-array(Count,c(2,2,6),list(Admit=c("Yes","No"),
Gender=c("M","F"),Dept=LETTERS[1:6]))
> contin.table
, , Dept = A
     Gender
Admit   M  F
  Yes 512 89
  No  313 19

, , Dept = B
      Gender
Admit   M  F
  Yes 353 17
  No  207  8

, , Dept = C
     Gender
Admit   M   F
  Yes 120 202
  No  205 391

, , Dept = D
     Gender
Admit   M   F
  Yes 138 131
  No  279 244

, , Dept = E
     Gender
Admit   M   F
  Yes  53  94
  No  138 299

, , Dept = F
     Gender
Admit   M   F
  Yes  22  24
  No  351 317
```

Note how array works: it fills up the elements in a 2 x 2 x 6 array, column by column. The order of the data in the data frame **`berkeley.df`** has to be taken into account, particularly when making up the labels for the rows, columns and slices: this is the last argument

```
list(Admit=c("Yes","No"), Gender=c("M","F"),Dept=LETTERS[1:6]))
```

which is a list with 3 elements containing the row labels, the column labels and the slice labels respectively. The data frame has to be attached so that R can see the variables. The arrays can

be printed in different ways, by interchanging the roles of rows, columns and slices. Thus can be done by the function aperm, see the documentation for details. For example:

```
> aperm(contin.table)
, , Admit = Yes
    Gender
Dept   M   F
   A 512  89
   B 353  17
   C 120 202
   D 138 131
   E  53  94
   F  22  24

, , Admit = No
    Gender
Dept   M   F
   A 313  19
   B 207   8
   C 205 391
   D 279 244
   E 138 299
   F 351 317
```

Task 3.
Calculate the marginal table of gender by admission. Is there evidence of discrimination?

The marginal tables are calculated using the function **`apply`**, which applies a given function to specified dimensions of the table:

```
> apply(contin.table, c(1,2), sum)
     Gender
Admit    M    F
  Yes 1198  557
  No  1493 1278
```

Note that we specify the dimensions of the array we are **not** adding over as c(1,2).

The odds ratio is $1198 \times 1278/(557 \times 1493) = 1.84$, so that there seems to be strong evidence of discrimination against females (the odds ratio here is the ratio

$$\text{OR} = (\text{male yes/male no}) / (\text{female yes/female no}),$$

so that a ratio >1 favours males). A formal chi-square test for independence rejects the independence hypothesis – check this using the code at the end of Lecture 28.

Task 4.
Now work out the conditional OR's for the separate departments. What do you see?

We could do this by hand. Alternatively, here is an R trick: Write a small function to compute OR's, then use **`apply`** to apply this function to the separate departments.

```
OR<-function(A){
# calculates OR for a 2 x 2 array A
A[1,1]*A[2,2]/(A[1,2]*A[2,1])
}
> apply(contin.table, 3, OR)
        A         B         C         D         E         F
0.3492120 0.8025007 1.1330596 0.9212838 1.2216312 0.8278727
```

Another way is to fit the saturated model, and compute the conditional OR's by adding the two-factor and three-factor interactions together.

```
> berkeley.glm=glm(Count~Admit*Gender*Dept, family=poisson, data=berkeley.df)
> summary(berkeley.glm)
Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)            2.9444     0.2294  12.835  < 2e-16 ***
AdmitNo                1.5442     0.2527   6.110 9.94e-10 ***
GenderM                2.8018     0.2363  11.858  < 2e-16 ***
DeptB                 -0.8650     0.4215  -2.052  0.04013 *
DeptC                  3.0243     0.2349  12.873  < 2e-16 ***
DeptD                  2.5527     0.2382  10.718  < 2e-16 ***
DeptE                  2.7560     0.2366  11.649  < 2e-16 ***
DeptF                  2.8145     0.2362  11.916  < 2e-16 ***
AdmitNo:GenderM       -1.0521     0.2627  -4.005 6.21e-05 ***
AdmitNo:DeptB         -0.7904     0.4977  -1.588  0.11224
AdmitNo:DeptC         -2.2046     0.2672  -8.252  < 2e-16 ***
AdmitNo:DeptD         -2.1662     0.2750  -7.878 3.32e-15 ***
AdmitNo:DeptE         -2.7013     0.2790  -9.682  < 2e-16 ***
AdmitNo:DeptF         -4.1250     0.3297 -12.512  < 2e-16 ***
GenderM:DeptB          0.4515     0.4309   1.048  0.29469
GenderM:DeptC         -3.4475     0.2515 -13.707  < 2e-16 ***
GenderM:DeptD         -2.6677     0.2520 -10.586  < 2e-16 ***
GenderM:DeptE         -3.5750     0.2577 -13.872  < 2e-16 ***
GenderM:DeptF         -2.6999     0.2487 -10.858  < 2e-16 ***
AdmitNo:GenderM:DeptB  0.8321     0.5104   1.630  0.10306
AdmitNo:GenderM:DeptC  1.1770     0.2996   3.929 8.53e-05 ***
AdmitNo:GenderM:DeptD  0.9701     0.3026   3.206  0.00135 **
AdmitNo:GenderM:DeptE  1.2523     0.3303   3.791  0.00015 ***
AdmitNo:GenderM:DeptF  0.8632     0.4027   2.144  0.03206 *
```

To calculate the conditional OR's we add the Admit:gender interaction to the 3-factor interactions (treating the AdmitNo:GenderM:DeptA interaction as zero, since A is baseline) The following does it:

```
> coefs=coef(berkeley.glm)
> exp(c(coefs[9],  coefs[9] +coefs[20:24]))
      AdmitNo:GenderM AdmitNo:GenderM:DeptB AdmitNo:GenderM:DeptC
            0.3492120             0.8025007             1.1330596
AdmitNo:GenderM:DeptD AdmitNo:GenderM:DeptE AdmitNo:GenderM:DeptF
            0.9212838             1.2216312             0.8278727
```

We see that when the individual departments are taken into account, the OR's are predominantly less than 1, indicating that the *males* are discriminated against! This situation, when the marginal association (the OR in the marginal table) is in a different direction to that in the conditional tables, is called *association reversal*, or *Simpson's paradox*. See Lecture 30.