Logistic Reggression on the Berkrley Graduate Admissions Data*

```
> rm(list=ls()); options(scipen=999)
> # UCBAdmissions is a built-in R data set
> UCBAdmissions

, , Dept = A

          Gender
Admit      Male Female
  Admitted  512     89
  Rejected  313     19

, , Dept = B

          Gender
Admit      Male Female
  Admitted  353     17
  Rejected  207      8

, , Dept = C

          Gender
Admit      Male Female
  Admitted  120    202
  Rejected  205    391

, , Dept = D

          Gender
Admit      Male Female
  Admitted  138    131
  Rejected  279    244

, , Dept = E

          Gender
Admit      Male Female
  Admitted   53     94
  Rejected  138    299

, , Dept = F

          Gender
Admit      Male Female
  Admitted   22     24
  Rejected  351    317


# For each department, I want to see the odds rastio and a test of independence
> # Usual odds ratio will be odds of admission for males / odds for females.
> # I'd rather see it the other way.
>
> # Odds ratio function
> oddrat = function(M) M[1,1]*M[2,2] / (M[1,2]*M[2,1])
>
> Bsummary = matrix(NA,6,3)
> colnames(Bsummary) = c('Odds Ratio', 'X-squared', 'p-value')
> rownames(Bsummary) = c('A', 'B', 'C', 'D', 'E', 'F')
>
```

* See last page for copyright information

```
> for(j in 1:6)
+ {
+ subtable = UCBAdmissions[,,j]
+ Bsummary[j,1] = 1/oddrat(subtable) # Making it Females to Males
+ x2test = chisq.test(subtable,correct=FALSE)
+ Bsummary[j,2] = x2test$statistic
+ Bsummary[j,3] = x2test$p.value
+ } # End looping through sub-tables
>
> round(Bsummary,3)
  Odds Ratio X-squared p-value
A      2.864    17.248   0.000
B      1.246     0.254   0.614
C      0.883     0.754   0.385
D      1.085     0.298   0.585
E      0.819     1.001   0.317
F      1.208     0.384   0.535
>
> # There is more than one way to prepare the data for a logistic regression, but
this works.
> UCB = as.data.frame(UCBAdmissions); UCB
      Admit Gender Dept Freq
1  Admitted   Male    A  512
2  Rejected   Male    A  313
3  Admitted Female    A   89
4  Rejected Female    A   19
5  Admitted   Male    B  353
6  Rejected   Male    B  207
7  Admitted Female    B   17
8  Rejected Female    B    8
9  Admitted   Male    C  120
10 Rejected   Male    C  205
11 Admitted Female    C  202
12 Rejected Female    C  391
13 Admitted   Male    D  138
14 Rejected   Male    D  279
15 Admitted Female    D  131
16 Rejected Female    D  244
17 Admitted   Male    E   53
18 Rejected   Male    E  138
19 Admitted Female    E   94
20 Rejected Female    E  299
21 Admitted   Male    F   22
22 Rejected   Male    F  351
23 Admitted Female    F   24
24 Rejected Female    F  317
> nrows = dim(UCB)[1] # 24 rows
> Berkeley = NULL
> for(j in 1:nrows)
+ {
+     oneline = UCB[j,1:3]
+     for(i in 1:UCB[j,4]) Berkeley = rbind(Berkeley,oneline)
+ } # Next j (row of UCB)
>
> head(Berkeley)
     Admit Gender Dept
1 Admitted   Male    A
2 Admitted   Male    A
3 Admitted   Male    A
4 Admitted   Male    A
5 Admitted   Male    A
6 Admitted   Male    A
```

```
> dim(Berkeley)
[1] 4526    3
> sum(UCB$Freq)
[1] 4526
> Berkeley = within(Berkeley,{
+ y = numeric(dim(Berkeley)[1])
+ y[Admit == 'Admitted'] = 1
+ contrasts(Dept) = contr.sum
+ contrasts(Gender) = contr.sum
+ }) # End within Berkeley
> head(Berkeley)
     Admit Gender Dept y
1 Admitted   Male    A 1
2 Admitted   Male    A 1
3 Admitted   Male    A 1
4 Admitted   Male    A 1
5 Admitted   Male    A 1
6 Admitted   Male    A 1
> with(Berkeley, {
+ print( table(Admit,y, useNA = 'ifany') )
+ cat('\nDummy variables for Department \n')
+ print(contrasts(Dept))
+ cat('\nDummy variables for Gender \n')
+ print(contrasts(Gender))
+ }) # End checking Berkeley
          y
Admit         0    1
  Admitted    0 1755
  Rejected 2771    0

Dummy variables for Department
  [,1] [,2] [,3] [,4] [,5]
A    1    0    0    0    0
B    0    1    0    0    0
C    0    0    1    0    0
D    0    0    0    1    0
E    0    0    0    0    1
F   -1   -1   -1   -1   -1

Dummy variables for Gender
       [,1]
Male      1
Female   -1
>
```

```
> full = glm(y ~ Dept*Gender,family=binomial,data=Berkeley)
> summary(full)

Call:
glm(formula = y ~ Dept * Gender, family = binomial, data = Berkeley)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8642  -0.9127  -0.3821   0.9768   2.3793

Coefficients:
             Estimate Std. Error z value             Pr(>|z|)
(Intercept)  -0.555229   0.055089 -10.079 < 0.0000000000000002 ***
Dept1         1.573389   0.120571  13.049 < 0.0000000000000002 ***
Dept2         1.198990   0.186947   6.414       0.000000000142 ***
Dept3        -0.042750   0.080548  -0.531              0.59560
Dept4        -0.107735   0.082433  -1.307              0.19123
Dept5        -0.501826   0.098578  -5.091       0.000000356829 ***
Gender1      -0.101489   0.055089  -1.842              0.06543 .
Dept1:Gender1 -0.424549   0.120571  -3.521              0.00043 ***
Dept2:Gender1 -0.008522   0.186947  -0.046              0.96364
Dept3:Gender1  0.163950   0.080548   2.035              0.04181 *
Dept4:Gender1  0.060496   0.082433   0.734              0.46302
Dept5:Gender1  0.201583   0.098578   2.045              0.04086 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6044.3  on 4525  degrees of freedom
Residual deviance: 5167.3  on 4514  degrees of freedom
AIC: 5191.3

Number of Fisher Scoring iterations: 5

>
> # Wald tests
> # Wtest = function(L,Tn,Vn,h=0) # H0: L theta = h
> source("http://www.utstat.utoronto.ca/brunner/Rfunctions/Wtest.txt")
>
> # Department (averaging log odds across Gender)
> Ldept = rbind(c(0,1,0,0,0,0,0,0,0,0,0,0),
+               c(0,0,1,0,0,0,0,0,0,0,0,0),
+               c(0,0,0,1,0,0,0,0,0,0,0,0),
+               c(0,0,0,0,1,0,0,0,0,0,0,0),
+               c(0,0,0,0,0,1,0,0,0,0,0,0))
> Wtest(L=Ldept, Tn=coef(full), Vn=vcov(full))
       W       df  p-value
389.4924   5.0000   0.0000
>
> # Gender (averaging log odds across Department)
> Lgender = rbind(c(0,0,0,0,0,0,1,0,0,0,0,0))
> Wtest(L=Lgender, Tn=coef(full), Vn=vcov(full))
        W        df   p-value
3.3940411 1.0000000 0.0654324
> # Compare z-test
>
```

```
> # Interaction
> Linter = rbind(c(0,0,0,0,0,0,0,1,0,0,0,0),
+                c(0,0,0,0,0,0,0,0,1,0,0,0),
+                c(0,0,0,0,0,0,0,0,0,1,0,0),
+                c(0,0,0,0,0,0,0,0,0,0,1,0),
+                c(0,0,0,0,0,0,0,0,0,0,0,1)    )
> Wtest(L=Linter, Tn=coef(full), Vn=vcov(full))
          W          df     p-value
17.90171441  5.00000000  0.00307214
>
> # Test interaction with full versus restricted models
> nointer = glm(y ~ Dept + Gender,family=binomial,data=Berkeley)
> anova(nointer,full,test='Chisq')
Analysis of Deviance Table

Model 1: y ~ Dept + Gender
Model 2: y ~ Dept * Gender
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      4519     5187.5
2      4514     5167.3  5   20.204 0.001144 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
> # Now follow up the interaction. Which odds ratios are different?
> # Modfying Berkeley again ...
> Berkeley = within(Berkeley,{GenderDept = paste(Gender,Dept,sep='')})
> with(Berkeley,{ table(GenderDept,Dept, useNA = 'ifany') })
          Dept
GenderDept   A   B   C   D   E   F
   FemaleA 108   0   0   0   0   0
   FemaleB   0  25   0   0   0   0
   FemaleC   0   0 593   0   0   0
   FemaleD   0   0   0 375   0   0
   FemaleE   0   0   0   0 393   0
   FemaleF   0   0   0   0   0 341
   MaleA   825   0   0   0   0   0
   MaleB     0 560   0   0   0   0
   MaleC     0   0 325   0   0   0
   MaleD     0   0   0 417   0   0
   MaleE     0   0   0   0 191   0
   MaleF     0   0   0   0   0 373
>
> cellmeans = glm(y ~ 0 + GenderDept, family=binomial,data=Berkeley)
> summary(cellmeans)

Call:
glm(formula = y ~ 0 + GenderDept, family = binomial, data = Berkeley)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8642  -0.9127  -0.3821   0.9768   2.3793

Coefficients:
                  Estimate Std. Error z value             Pr(>|z|)
GenderDeptFemaleA  1.54420    0.25272   6.110    0.000000000994422 ***
GenderDeptFemaleB  0.75377    0.42875   1.758               0.0787 .
GenderDeptFemaleC -0.66044    0.08665  -7.622    0.000000000000025 ***
GenderDeptFemaleD -0.62197    0.10831  -5.742    0.000000009340561 ***
GenderDeptFemaleE -1.15715    0.11825  -9.786 < 0.0000000000000002 ***
GenderDeptFemaleF -2.58085    0.21171 -12.190 < 0.0000000000000002 ***
GenderDeptMaleA    0.49212    0.07175   6.859    0.000000000006941 ***
GenderDeptMaleB    0.53375    0.08754   6.097    0.000000001080813 ***
GenderDeptMaleC   -0.53552    0.11494  -4.659    0.000003176262477 ***
GenderDeptMaleD   -0.70396    0.10407  -6.764    0.000000000013399 ***
```

```
GenderDeptMaleE   -0.95696    0.16160  -5.922    0.000000003183932 ***
GenderDeptMaleF   -2.76974    0.21978 -12.602 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6274.4  on 4526  degrees of freedom
Residual deviance: 5167.3  on 4514  degrees of freedom
AIC: 5191.3

Number of Fisher Scoring iterations: 5

>
> # The beta-hats are estimated log odds. Just as a check,
> UCBAdmissions[,,1] # Department A
          Gender
Admit      Male Female
  Admitted  512     89
  Rejected  313     19
> 512/(512+313)
[1] 0.6206061
> # Log odds = 0.49212, so ...
> exp(0.49212)/(1+exp(0.49212))
[1] 0.6206057
>
> # Now all pairwise comparisons of odds ratios
> # First just get estimated odds ratios for each department
>
> #     1  2  3  4  5  6  7  8  9 10 11 12
> a = c(1, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0)
> b = c(0, 1, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0)
> c = c(0, 0, 1, 0, 0, 0, 0, 0,-1, 0, 0, 0)
> d = c(0, 0, 0, 1, 0, 0, 0, 0, 0,-1, 0, 0)
> e = c(0, 0, 0, 0, 1, 0, 0, 0, 0, 0,-1, 0)
> f = c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,-1)
>
> logorHat = coefficients(cellmeans)
> as.matrix(logorHat)
                        [,1]
GenderDeptFemaleA  1.5441974
GenderDeptFemaleB  0.7537718
GenderDeptFemaleC -0.6604399
GenderDeptFemaleD -0.6219709
GenderDeptFemaleE -1.1571488
GenderDeptFemaleF -2.5808479
GenderDeptMaleA    0.4921214
GenderDeptMaleB    0.5337493
GenderDeptMaleC   -0.5355182
GenderDeptMaleD   -0.7039581
GenderDeptMaleE   -0.9569618
GenderDeptMaleF   -2.7697438
```

```
> # Estimated odds ratios for the departments
> exp(sum(logorHat*a)) # A
[1] 2.86359
> exp(sum(logorHat*b)) # B
[1] 1.246105
> exp(sum(logorHat*c)) # C
[1] 0.8825661
> exp(sum(logorHat*d)) # D
[1] 1.085442
> exp(sum(logorHat*e)) # E
[1] 0.8185776
> exp(sum(logorHat*f)) # F
[1] 1.207915
>
> # Compare estimated odds ratios from the table
> Bsummary
  Odds Ratio  X-squared       p-value
A  2.8635896 17.2480134 0.00003280404
B  1.2461048  0.2537215 0.61446676567
C  0.8825661  0.7535389 0.38535809298
D  1.0854419  0.2979776 0.58515307222
E  0.8185776  1.0010686 0.31705206682
F  1.2079151  0.3840933 0.53542068131
>
> # Now calculate L matrices for H0: L beta = 0
> # Need rbind to make them matrices
> AvsB = rbind(a-b)
> AvsC = rbind(a-c)
> AvsD = rbind(a-d)
> AvsE = rbind(a-e)
> AvsF = rbind(a-f)
> BvsC = rbind(b-c)
> BvsD = rbind(b-d)
> BvsE = rbind(b-e)
> BvsF = rbind(b-f)
> CvsD = rbind(c-d)
> CvsE = rbind(c-e)
> CvsF = rbind(c-f)
> DvsE = rbind(d-e)
> DvsF = rbind(d-f)
> EvsF = rbind(e-f)
>
```

```
> # Humm, 15 tests
> Wtest(L=AvsB, Tn=coef(cellmeans), Vn=vcov(cellmeans))
       W       df  p-value
2.657602 1.000000 0.103056
> Wtest(L=AvsC, Tn=coef(cellmeans), Vn=vcov(cellmeans))
             W             df        p-value
15.43794267248  1.00000000000  0.00008525915
> Wtest(L=AvsD, Tn=coef(cellmeans), Vn=vcov(cellmeans))
            W           df      p-value
10.276170683  1.000000000  0.001347593
> Wtest(L=AvsE, Tn=coef(cellmeans), Vn=vcov(cellmeans))
             W            df       p-value
14.3719621528  1.0000000000  0.0001500196
> Wtest(L=AvsF, Tn=coef(cellmeans), Vn=vcov(cellmeans))
          W         df    p-value
4.59531468 1.00000000 0.03205946
> Wtest(L=BvsC, Tn=coef(cellmeans), Vn=vcov(cellmeans))
         W         df   p-value
0.5607102 1.0000000 0.4539742
> Wtest(L=BvsD, Tn=coef(cellmeans), Vn=vcov(cellmeans))
         W         df   p-value
0.0890155 1.0000000 0.7654325
> Wtest(L=BvsE, Tn=coef(cellmeans), Vn=vcov(cellmeans))
         W         df   p-value
0.7624697 1.0000000 0.3825567
> Wtest(L=BvsF, Tn=coef(cellmeans), Vn=vcov(cellmeans))
         W         df   p-value
0.0034042 1.0000000 0.9534734
> Wtest(L=CvsD, Tn=coef(cellmeans), Vn=vcov(cellmeans))
         W         df   p-value
0.9891244 1.0000000 0.3199565
> Wtest(L=CvsE, Tn=coef(cellmeans), Vn=vcov(cellmeans))
          W         df    p-value
0.09314708 1.00000000 0.76021379
> Wtest(L=CvsF, Tn=coef(cellmeans), Vn=vcov(cellmeans))
         W         df   p-value
0.8650638 1.0000000 0.3523255
> Wtest(L=DvsE, Tn=coef(cellmeans), Vn=vcov(cellmeans))
         W         df   p-value
1.2707110 1.0000000 0.2596333
> Wtest(L=DvsF, Tn=coef(cellmeans), Vn=vcov(cellmeans))
          W         df    p-value
0.09879718 1.00000000 0.75327787
> Wtest(L=EvsF, Tn=coef(cellmeans), Vn=vcov(cellmeans))
         W         df   p-value
1.1363519 1.0000000 0.2864245
> Bsummary # Look again
  Odds Ratio  X-squared       p-value
A  2.8635896 17.2480134 0.00003280404
B  1.2461048  0.2537215 0.61446676567
C  0.8825661  0.7535389 0.38535809298
D  1.0854419  0.2979776 0.58515307222
E  0.8185776  1.0010686 0.31705206682
F  1.2079151  0.3840933 0.53542068131
```
