Logistic Regression with R: Example One

```
> math = read.table("http://www.utstat.toronto.edu/~brunner/appliedf12/data/mathcat.data")
> math[1:5,]
  hsgpa hsengl hscalc   course passed outcome
1  78.0     80    Yes Mainstrm     No  Failed
2  66.0     75    Yes Mainstrm    Yes  Passed
3  80.2     70    Yes Mainstrm    Yes  Passed
4  81.7     67    Yes Mainstrm    Yes  Passed
5  86.8     80    Yes Mainstrm    Yes  Passed
> attach(math) # Variable names are now available
> length(hsgpa)
[1] 394
>
> # First, some simple examples to illustrate the methods
> # Two continuous explanatory variables
> model1 = glm(passed ~ hsgpa + hsengl, family=binomial)
> summary(model1)

Call:
glm(formula = passed ~ hsgpa + hsengl, family = binomial)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.5577  -0.9833   0.4340   0.9126   2.2883  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept) -14.69568    2.00683  -7.323 2.43e-13 ***
hsgpa         0.22982    0.02955   7.776 7.47e-15 ***
hsengl       -0.04020    0.01709  -2.352   0.0187 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 530.66  on 393  degrees of freedom
Residual deviance: 437.69  on 391  degrees of freedom
AIC: 443.69

Number of Fisher Scoring iterations: 4

> betahat1 =  model1$coefficients; betahat1
 (Intercept)        hsgpa       hsengl 
-14.69567812   0.22982332  -0.04020062 
>
> # For a constant value of mark in HS English, for every one-point increase
> # in HS GPA, estimated odds of passing are multiplied by ...
> exp(betahat1[2])
   hsgpa 
1.258378 
```

Deviance = -2[L_M - L_S] (p. 85)

Where L_M is the maximum log likelihood of the model, and L_S is the maximum log likelihood of an "ideal" model that fits as well as possible. The greater the deviance, the worse the model fits compared to the "best case."

Akaike information criterion: AIC = 2p + Deviance,

where p = number of model parameters

```
>
> # Deviance = -2LL + c
> # Constant will be discussed later.
> # But recall that the likelihood ratio test statistic is the
> # DIFFERENCE between two -2LL values, so
> # G-squared = Deviance(Reduced)-Deviance(Full)
>
> # Test both explanatory variables at once
> # Null deviance is deviance of a model with just the intercept.
> model1$deviance
[1] 437.6855
> model1$null.deviance
[1] 530.6559
> # G-squared = Deviance(Reduced)-Deviance(Full)
> # df = difference in number of betas
> G2 = model1$null.deviance-model1$deviance; G2
[1] 92.97039
> 1-pchisq(G2,df=1)
[1] 0
>
> a1 = anova(model1); a1
Analysis of Deviance Table

Model: binomial, link: logit

Response: passed

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev
NULL                     393     530.66
hsgpa   1   87.221       392     443.43
hsengl  1    5.749       391     437.69
> # a1 is a matrix
> a1[1,4] - a1[2,4]
[1] 87.22114
> anova(model1,test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: passed

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                     393     530.66
hsgpa   1   87.221       392     443.43   <2e-16 ***
hsengl  1    5.749       391     437.69   0.0165 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> # For LR test of hsengl controlling for hagpa
> # Compare Z = -2.352,  p =   0.0187
```

```
>
> # Estimate the probability of passing for a student with
> # HSGPA = 80 and HS English = 75
```

$$\pi = \frac{e^{\beta_0+\beta_1 x_1+\ldots+\beta_{p-1}x_{p-1}}}{1+e^{\beta_0+\beta_1 x_1+\ldots+\beta_{p-1}x_{p-1}}}$$

```
>
> x = c(1,80,75); xb = sum(x*model1$coefficients)
> phat = exp(xb)/(1+exp(xb)); phat
[1] 0.6626533

> # An easier way
> gpa80eng75 = data.frame(hsgpa=80,hsengl=75)
> # Default type is estimated logit; type="response" gives estimated probability.
> predict(model1,newdata=gpa80eng75,type="response")
        1
0.6626533
>
> # Get standard error too
> predict(model1,newdata=gpa80eng75,type="response",se.fit=T)
$fit
        1
0.6626533

$se.fit
         1
0.02859302

$residual.scale
[1] 1

> # How did they calculate that standard error?
> Vhat = vcov(model1); Vhat
             (Intercept)         hsgpa        hsengl
(Intercept)  4.027354203 -0.0492223614 -0.0021256979
hsgpa       -0.049222361  0.0008734652 -0.0002541750
hsengl      -0.002125698 -0.0002541750  0.0002921532
> denom = (1+exp(xb))^2
> gdot = x*exp(xb)/denom; gdot
[1]  0.2235439 17.8835124 16.7657928
> gdot = matrix(gdot,nrow=1,ncol=3)
> sqrt(gdot %*% Vhat %*% t(gdot))
           [,1]
[1,] 0.02859302
```

```
>
> ############ Categorical explanatory variables  ############
> # Are represented by dummy variables.
> # First look at the data.
>
> coursepassed = table(course,passed); coursepassed
          passed
course      No Yes
  Catch-up  27   8
  Elite      7  24
  Mainstrm 124 204
> addmargins(coursepassed,c(1,2)) # See marginal totals too
          passed
course      No Yes Sum
  Catch-up  27   8  35
  Elite      7  24  31
  Mainstrm 124 204 328
  Sum      158 236 394
> prop.table(coursepassed,1) # See proportions of row totals
          passed
course            No       Yes
  Catch-up 0.7714286 0.2285714
  Elite    0.2258065 0.7741935
  Mainstrm 0.3780488 0.6219512

> # Now test with logistic regression and dummy variables
> is.factor(course) # Is course already a factor?
[1] TRUE
> contrasts(course) # Reference cat will be alphabetically first
         Elite Mainstrm
Catch-up     0        0
Elite        1        0
Mainstrm     0        1
> # Want Mainstream to be the reference category
> contrasts(course) = contr.treatment(3,base=3)
> contrasts(course)
         1 2
Catch-up 1 0
Elite    0 1
Mainstrm 0 0
```

```
>
> model2 = glm(passed ~ course, family=binomial); summary(model2)

Call:
glm(formula = passed ~ course, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7251  -1.3948   0.9746   0.9746   1.7181

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.4978     0.1139   4.372 1.23e-05 ***
course1      -1.7142     0.4183  -4.098 4.17e-05 ***
course2       0.7343     0.4444   1.652   0.0985 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 530.66  on 393  degrees of freedom
Residual deviance: 505.74  on 391  degrees of freedom
AIC: 511.74

Number of Fisher Scoring iterations: 4

> anova(model2) # Both dummy variables are entered at once bec. course is a factor.
Analysis of Deviance Table

Model: binomial, link: logit

Response: passed

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev
NULL                     393     530.66
course  2   24.916       391     505.74

> # Compare a Pearson Chi-squared test of independence.
> chisq.test(coursepassed)

     Pearson's Chi-squared test

data:  coursepassed
X-squared = 24.6745, df = 2, p-value = 4.385e-06
```

```
>
> # The estimated odds of passing are __ times as great for students in
> # the catch-up course, compared to students in the mainstream course.
> model2$coefficients
(Intercept)     course1     course2
  0.4978384  -1.7142338   0.7343053
> exp(model2$coefficients[2])
  course1
0.1801017
>
> # Get that number from the contingency table
> addmargins(coursepassed,c(1,2))
          passed
course      No Yes Sum
  Catch-up  27   8  35
  Elite      7  24  31
  Mainstrm 124 204 328
  Sum      158 236 394
> pr = prop.table(coursepassed,1); pr # Estimated conditional probabilities
          passed
course            No       Yes
  Catch-up 0.7714286 0.2285714
  Elite    0.2258065 0.7741935
  Mainstrm 0.3780488 0.6219512

> odds1 = pr[1,2]/(1-pr[1,2]); odds1
[1] 0.2962963
> odds3 = pr[3,2]/(1-pr[3,2]); odds3
[1] 1.645161
> odds1/odds3
[1] 0.1801017
> exp(model2$coefficients[2])
  course1
0.1801017
```

```
> ############### Now a more realistic analysis ####################
>
> model3 = glm(passed ~ hsengl + hsgpa + course, family=binomial)
> summary(model3)

Call:
glm(formula = passed ~ hsengl + hsgpa + course, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5404  -0.9852   0.4110   0.8820   2.2109

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -14.18265    2.06382  -6.872 6.33e-12 ***
hsengl       -0.03534    0.01766  -2.001  0.04539 *
hsgpa         0.21939    0.02988   7.342 2.10e-13 ***
course1      -1.29137    0.45190  -2.858  0.00427 **
course2       0.75847    0.49308   1.538  0.12399
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 530.66  on 393  degrees of freedom
Residual deviance: 424.76  on 389  degrees of freedom
AIC: 434.76

Number of Fisher Scoring iterations: 4

> anova(model3,test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: passed

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     393     530.66
hsengl  1    8.286       392     522.37  0.003994 **
hsgpa   1   84.684       391     437.69 < 2.2e-16 ***
course  2   12.921       389     424.76  0.001564 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> # Interpret all the default tests, but watch out!
> summary(glm(passed ~ hsengl, family=binomial))

Call:
glm(formula = passed ~ hsengl, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5895  -1.3039   0.8913   1.0133   1.4060

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.29604    0.95182  -2.412  0.01585 *
hsengl       0.03546    0.01247   2.844  0.00446 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Repeating a little from earlier ...

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -14.18265    2.06382  -6.872 6.33e-12 ***
hsengl       -0.03534    0.01766  -2.001  0.04539 *
hsgpa         0.21939    0.02988   7.342 2.10e-13 ***
course1      -1.29137    0.45190  -2.858  0.00427 **
course2       0.75847    0.49308   1.538  0.12399

      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                    393     530.66
hsengl  1    8.286       392     522.37  0.003994 **
hsgpa   1   84.684       391     437.69 < 2.2e-16 ***
course  2   12.921       389     424.76  0.001564 **
```

```
> # Reproduce the Z-test for hsengl
> betahat3 = model3$coefficients; betahat3
 (Intercept)       hsengl        hsgpa      course1      course2
-14.18264539  -0.03533871   0.21939002  -1.29136575   0.75846785
>
> V3 = vcov(model3)
> Z = betahat3[2]/sqrt(V3[2,2]) ; Z
   hsengl
-2.001046
> # Do some Wald tests
>
> WaldTest = function(L,thetahat,Vn,h=0) # H0: L theta = h
+ # Note Vn is the asymptotic covariance matrix, so it's the
+ # Consistent estimator divided by n. For true Wald tests
+ # based on numerical MLEs, just use the inverse of the Hessian.
+      {
+      WaldTest = numeric(3)
+      names(WaldTest) = c("W","df","p-value")
+      r = dim(L)[1]
+      W = t(L%*%thetahat-h) %*% solve(L%*%Vn%*%t(L)) %*%
+           (L%*%thetahat-h)
+      W = as.numeric(W)
+      pval = 1-pchisq(W,r)
+      WaldTest[1] = W; WaldTest[2] = r; WaldTest[3] = pval
+      WaldTest
+      } # End function WaldTest
>
>
> # Wald chi-squared for hsengl
> L1 = rbind(c(0,1,0,0,0))
>
> WaldTest(L=L1,thetahat=betahat3,Vn=V3)
         W         df    p-value
4.00418656 1.00000000 0.04538739
> Z^2
  hsengl
4.004187
> # Test course controlling for hsengl and hsgpa
> # Compare LR G^2 = 12.921, df=2, p=0.001564
> L2 = rbind(c(0,0,0,1,0),
+            c(0,0,0,0,1) )
> WaldTest(L=L2,thetahat=betahat3,Vn=V3)
          W           df      p-value
11.324864041  2.000000000  0.003474058
```

```
> # How about whether they took HS Calculus?
> model4 = update(model3,~ . + hscalc); summary(model4)

Call:
glm(formula = passed ~ hsengl + hsgpa + course + hscalc, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5517  -0.9811   0.4059   0.8716   2.2061

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -15.42813    2.20154  -7.008 2.42e-12 ***
hsengl       -0.03619    0.01776  -2.038   0.0416 *
hsgpa         0.22036    0.03003   7.337 2.19e-13 ***
course1      -0.88042    0.48834  -1.803   0.0714 .
course2       0.79966    0.50023   1.599   0.1099
hscalcYes     1.25718    0.67282   1.869   0.0617 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 530.66  on 393  degrees of freedom
Residual deviance: 420.90  on 388  degrees of freedom
AIC: 432.9

Number of Fisher Scoring iterations: 4

>
> # Test course controlling for others
> notcourse = glm(passed ~ hsgpa + hsengl + hscalc , family = binomial)
> anova(notcourse,model4,test="Chisq")
Analysis of Deviance Table

Model 1: passed ~ hsgpa + hsengl + hscalc
Model 2: passed ~ hsengl + hsgpa + course + hscalc
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       390     427.75
2       388     420.90  2   6.8575  0.03243 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> # I like Model 3.
```

```
> # I like Model 3. Answer the following questions based on Model 3.
>
> # Controlling for High School english mark and High School GPA,
> # the estimated odds of passing are ___ times as great for students in
> # the Elite course, compared to students in the Catch-up course.
>
> betahat3 = model3$coefficients; betahat3
 (Intercept)       hsengl        hsgpa      course1      course2 
-14.18264539  -0.03533871   0.21939002  -1.29136575   0.75846785 
> exp(betahat3[5])/exp(betahat3[4])
 course2 
7.766609 
>
> # What is the estimated probability of passing for a student
> # in the mainstream course with 90% in HS English and a HS GPA of 80%?
>
> x = c(1,90,80,0,0); xb = sum(x*model3$coefficients)
> phat = exp(xb)/(1+exp(xb)); phat
[1] 0.54688
>
> # What if the student had 50% in HS English?
> x = c(1,50,80,0,0); xb = sum(x*model3$coefficients)
> phat = exp(xb)/(1+exp(xb)); phat
[1] 0.8322448
>
> # What if the student had -40 in HS English?
> x = c(1,-40,80,0,0); xb = sum(x*model3$coefficients)
> phat = exp(xb)/(1+exp(xb)); phat
[1] 0.9916913

> # Could do it with predict
> ez = data.frame(hsengl=c(90,50,-40), hsgpa=c(80,80,80),
+                 course=c("Mainstrm","Mainstrm","Mainstrm"))
> predict(model3,newdata=ez,type="response")
        1         2         3 
0.5468800 0.8322448 0.9916913 
```

A confidence interval would be nice.