Logistic Regression with R: Example One*

```
> options(scipen=999) # To supress scientific notation
> math = read.table("http://www.utstat.toronto.edu/brunner/data/legal/mathcat.data.txt")
  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
> n = length(hsgpa); n
[1] 394
>
> # First, some simple examples to illustrate the methods
> # Two continuous explanatory variables
> # y values must be numeric
> pass = numeric(n); pass[passed=='Yes'] = 1
> table(passed,pass)
      pass
passed   0   1
   No  158   0
   Yes   0 236
>
> model1 = glm(pass ~ hsgpa + hsengl, family=binomial)
> summary(model1)

Call:
glm(formula = pass ~ 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 0.00000000000024277 ***
hsgpa         0.22982    0.02955   7.776 0.00000000000000747 ***
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
```

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 = 2(k+1) + Deviance,
where k+1 = number of model parameters

```
> 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 = -2LL + c
> # 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=2)
[1] 0

> # Use anova function
> nullmodel = glm(pass ~ 1, family=binomial) # Just the intercept
> anova(nullmodel,model1, test = 'Chisq')
Analysis of Deviance Table

Model 1: pass ~ 1
Model 2: pass ~ hsgpa + hsengl
  Resid. Df Resid. Dev Df Deviance              Pr(>Chi)    
1       393     530.66                                      
2       391     437.69  2    92.97 < 0.00000000000000022 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
> # Wald test of both variables at once
> # Need asymptotic variance-covariance matrix
> 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
> 
> # For Wald tests: Wtest = function(L,Tn,Vn,h=0) # H0: L theta = h
> source("http://www.utstat.utoronto.ca/brunner/Rfunctions/Wtest.txt")
> 
> LL1 = rbind(c(0,1,0),
+             c(0,0,1))
> 
> Wtest(LL1,betahat1,Vhat)
                     W                     df                p-value 
63.73205975488392027730  2.00000000000000000000  0.00000000000000144329 
> round(Wtest(LL1,betahat1,V),3) # Compare G-squared = 92.97
     W      df p-value 
 63.732   2.000   0.000 
```

```
> # Confidence intervals for the beta_j
> sumtable = summary(model1)$coefficients; sumtable  # It's a matrix
               Estimate Std. Error   z value                 Pr(>|z|)
(Intercept) -14.69567812 2.00682690 -7.322843 0.000000000000242771950
hsgpa         0.22982332 0.02955444  7.776269 0.000000000000007469457
hsengl       -0.04020062 0.01709249 -2.351947 0.018675454042679614369
> se1 = sumtable[,2]; se1 # Column 2
(Intercept)       hsgpa      hsengl 
 2.00682690  0.02955444  0.01709249 
> lower95 = betahat1 - 1.96*se1; upper95 = betahat1 + 1.96*se1
> round( cbind(lower95,betahat1,upper95), 3)
            lower95 betahat1 upper95
(Intercept) -18.629  -14.696 -10.762
hsgpa         0.172    0.230   0.288
hsengl       -0.074   -0.040  -0.007
> 
> # Confidence intervals for the odds ratios
> CImat = cbind(lower95,betahat1,upper95) # Same as above
> round( exp(CImat), 3)
            lower95 betahat1 upper95
(Intercept)   0.000    0.000   0.000
hsgpa         1.188    1.258   1.333
hsengl        0.929    0.961   0.993

> # Likelihood ratio tests
> engonly = glm(pass ~ hsengl, family=binomial) # Ignoring GPA, not controlling for it.
> summary(engonly)

Call:
glm(formula = pass ~ 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 

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 530.66  on 393  degrees of freedom
Residual deviance: 522.37  on 392  degrees of freedom
AIC: 526.37

Number of Fisher Scoring iterations: 4

> gpaonly = glm(pass ~ hsgpa, family=binomial)
> 
> # Likelihood ratio test of hsengl controlling for hsgpa
> anova(gpaonly,model1, test = 'Chisq') # Compare Z = -2.352, Z^2 = 5.53
Analysis of Deviance Table

Model 1: pass ~ hsgpa
Model 2: pass ~ hsgpa + hsengl
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)  
1       392     443.43                       
2       391     437.69  1   5.7493   0.0165 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 
```

```
>
> # You can apply anova to a single glm model object, and get useful results
> # Don't do this with lm model objects!
>
> a1 = anova(model1,test="Chisq"); a1
Analysis of Deviance Table

Model: binomial, link: logit

Response: pass

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 <0.0000000000000002 ***
hsengl  1    5.749       391     437.69              0.0165 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> # a1 is a matrix
> a1[1,4] - a1[3,4] # Got 92.97 earlier for H0: beta1=beta2=0
[1] 92.97039

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

$$\pi = \frac{e^{\beta_0+\beta_1 x_1+\ldots+\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\ldots+\beta_k x_k}}$$

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

> # help(predict.glm)
> # predict(model1) would return a vector of n estimated log odds:
```

$$\widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \ldots + \widehat{\beta}_k x_k$$

```
> # For the existing data.
> # Generate estimated probabilities, for a NEW set of x values.
> new1 = data.frame(hsgpa=80, hsengl=75)
> predict(model1,newdata=new1,type="response")
        1
0.6626533

> # Predictions for a batch of new data
> GPA = c(80,80,80,80,80); ENG = c(100,75,50,25,0)
> data.frame(GPA,ENG)
  GPA ENG
1  80 100
2  80  75
3  80  50
4  80  25
5  80   0
```

```
> new2 = data.frame(GPA,ENG); colnames(new2) = c('hsgpa','hsengl')
> predict(model1,newdata=new2,type="response")
        1         2         3         4         5 
0.4182711 0.6626533 0.8429252 0.9361460 0.9756409 

> # There are two ways to get confidence intervals for the probabilities
> # First the direct way (using the multivariate delta method)
> pred1  = predict(model1,newdata=new2,type = "response", se.fit = TRUE)
> lower95 = pred1$fit - 1.96*pred1$se
> upper95 = pred1$fit + 1.96*pred1$se
> OutMat1 = cbind(GPA,ENG,pred1$fit,pred1$se,lower95,upper95)
> colnames(OutMat1)[3] = 'pi-hat'; colnames(OutMat1)[4] = 'se'
> OutMat1
  GPA ENG    pi-hat         se   lower95   upper95
1  80 100 0.4182711 0.10077863 0.2207450 0.6157972
2  80  75 0.6626533 0.02859302 0.6066110 0.7186956
3  80  50 0.8429252 0.06299436 0.7194563 0.9663942
4  80  25 0.9361460 0.05351814 0.8312504 1.0410415
5  80   0 0.9756409 0.03136680 0.9141619 1.0371198

> # The second way is to get an interval for the log odds, and transform it.
> # It works because the probability is an increasing function of the log odds.
> # The default for predict is x'betahat
> pred2  = predict(model1,newdata=new2, se.fit = TRUE)
> pred2
$fit
         1          2          3          4          5 
-0.3298749  0.6751407  1.6801563  2.6851719  3.6901875 

$se.fit
        1         2         3         4         5 
0.4141808 0.1279078 0.4757800 0.8953012 1.3198309 

$residual.scale
[1] 1

> a = pred2$fit - 1.96*pred2$se
> b = pred2$fit + 1.96*pred2$se # These are vectors
> lower95 = exp(a)/(1+exp(a))
> upper95 = exp(b)/(1+exp(b))
> pihat = exp(pred2$fit)/(1+exp(pred2$fit))
> OutMat2 = cbind(GPA,ENG,pihat,lower95,upper95)
> 
> OutMat2
  GPA ENG     pihat   lower95   upper95
1  80 100 0.4182711 0.2420140 0.6182010
2  80  75 0.6626533 0.6045455 0.7162306
3  80  50 0.8429252 0.6786615 0.9316735
4  80  25 0.9361460 0.7171527 0.9883411
5  80   0 0.9756409 0.7508815 0.9981246
> 
> OutMat1 # For comparison
  GPA ENG    pi-hat         se   lower95   upper95
1  80 100 0.4182711 0.10077863 0.2207450 0.6157972
2  80  75 0.6626533 0.02859302 0.6066110 0.7186956
3  80  50 0.8429252 0.06299436 0.7194563 0.9663942
4  80  25 0.9361460 0.05351814 0.8312504 1.0410415
5  80   0 0.9756409 0.03136680 0.9141619 1.0371198
```

```
> ############ Categorical explanatory variables  ############
> # Are represented by dummy variables.
> # First an example from earlier.
>
> 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
          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
>
> # Test independence, first with a Pearson X^2
> cp = chisq.test(coursepassed, correct=FALSE); cp

      Pearson's Chi-squared test

data:  coursepassed
X-squared = 24.674, df = 2, p-value = 0.000004385


> # Now LR test
```

$$G^2 = 2\sum_{i=1}^{I}\sum_{j=1}^{J} n_{ij}\log\left(\frac{n_{ij}}{\widehat{\mu}_{ij}}\right)$$

```
> muhat = cp$expected; nij = coursepassed
> G2 = 2 * sum( nij * log(nij/muhat) ); G2
[1] 24.91574

> # Now with logistic regression and dummy variables
> is.factor(course) # Is course already a factor?
[1] FALSE
> course = factor(course)
> contrasts(course) # Reference cat should be alphabetically first, Elite
         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(pass ~ course, family=binomial); summary(model2)

Call:
glm(formula = pass ~ 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 0.0000123 ***
course1      -1.7142     0.4183  -4.098 0.0000417 ***
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 G^2 = 24.91574 from the LR test of independence.
>
> # 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
```

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

Call:
glm(formula = pass ~ course + hsgpa + hsengl, 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 0.00000000000633 ***
course1      -1.29137    0.45190  -2.858          0.00427 **
course2       0.75847    0.49308   1.538          0.12399
hsgpa         0.21939    0.02988   7.342 0.00000000000021 ***
hsengl       -0.03534    0.01766  -2.001          0.04539 *
---
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: pass

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev              Pr(>Chi)
NULL                     393     530.66
course  2   24.916       391     505.74           0.000003887 ***
hsgpa   1   76.844       390     428.90 < 0.00000000000000022 ***
hsengl  1    4.132       389     424.76               0.04209 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
> # Interpret all the tests
```

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

Call:
glm(formula = pass ~ course + hsgpa + hsengl + 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 0.000000000002419 ***
course1      -0.88042    0.48834  -1.803            0.0714 .
course2       0.79966    0.50023   1.599            0.1099
hsgpa         0.22036    0.03003   7.337 0.000000000000219 ***
hsengl       -0.03619    0.01776  -2.038            0.0416 *
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(pass ~ hsgpa + hsengl + hscalc , family = binomial)
> anova(notcourse,model4,test="Chisq")
Analysis of Deviance Table

Model 1: pass ~ hsgpa + hsengl + hscalc
Model 2: pass ~ course + hsgpa + hsengl + 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$coefficients; betahat3
 (Intercept)      course1      course2        hsgpa       hsengl
-14.18264539  -1.29136575   0.75846785   0.21939002  -0.03533871
> exp(betahat3[3])/exp(betahat3[2])
 course2
7.766609
>
> # What is the estimated probability of passing for a student
> # in the mainstream course with 90% in HS English and a HS GPA of 80%?
>
> x = c(1,0,0,80,90); xb = sum(x*model3$coefficients)
> phat = exp(xb)/(1+exp(xb)); phat
[1] 0.54688
>
> # What if the student had 50% in HS English?
> x = c(1,0,0,80,50); xb = sum(x*model3$coefficients)
> phat = exp(xb)/(1+exp(xb)); phat
[1] 0.8322448
>
> # What if the student had -40 in HS English?
> x = c(1,0,0,80,-40); xb = sum(x*model3$coefficients)
> phat = exp(xb)/(1+exp(xb)); phat
[1] 0.9916913
>
> ######################### Prediction ################################
>
> # First, pseudo-prediction: Bad no no don't do it.
> pihat = predict(model3,type="response")
> prop.table(table(pass))
pass
        0         1
0.4010152 0.5989848
> mean(pihat) # Cool huh?
[1] 0.5989848
>
> predpass = cut(pihat,breaks=c(0,0.5,1),labels = c('No','Yes'))
> # Actually it's half open (0,0.5], but who cares?
> table(predpass)
predpass
 No Yes
137 257
> n = sum(table(predpass)) ; table(predpass)/n
predpass
       No       Yes
0.3477157 0.6522843
>
> prp = table(predpass,pass); prp
        pass
predpass   0   1
     No   98  39
     Yes  60 197
> prprop = prp/n; prprop
        pass
predpass          0          1
     No  0.24873096 0.09898477
     Yes 0.15228426 0.50000000
> # About 75% "accurate"
```

```
> prop.table(prp,margin=1) # Row proportions
        pass
predpass         0         1
     No  0.7153285 0.2846715
     Yes 0.2334630 0.7665370
>
> # But this may be too optimistic. We have a validation data set.
> math2 = read.table("https://www.utstat.toronto.edu/~brunner/data/legal/mathcat-
replic.data.txt")
> pihat2 = predict(model3,newdata=math2,type="response")
> predpass2 = cut(pihat2,breaks=c(0,0.5,1),labels = c('No','Yes'))
> passed2 = math2$passed
> ptable2 = table(predpass2,passed2); ptable2
         passed2
predpass2  No Yes
      No  103  56
      Yes  82 179
> prop.table(ptable2)
         passed2
predpass2        No       Yes
      No  0.2452381 0.1333333
      Yes 0.1952381 0.4261905
> 0.2452381 + 0.4261905 # Compared to about 75% accurate before
[1] 0.6714286
>
> prop.table(ptable2,margin=1) # Row proportions
         passed2
predpass2        No       Yes
      No  0.6477987 0.3522013
      Yes 0.3141762 0.6858238
> prop.table(prp,margin=1) # For comparison
        pass
predpass         0         1
     No  0.7153285 0.2846715
     Yes 0.2334630 0.7665370
>
```

```
> # Finally, will the interesting finding replicate?
>
> n = dim(math2)[1]; n
[1] 420
> detach(math); attach(math2)
The following object is masked _by_ .GlobalEnv:

    course

> course = factor(math2$course) # Unfortunate, but ...
> pass = numeric(n); pass[passed=='Yes'] = 1
>
> model3b = glm(pass ~ course + hsgpa + hsengl, family=binomial)
> summary(model3b)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8427  -1.1238   0.6447   1.0020   4.0476

Coefficients:
                Estimate Std. Error z value    Pr(>|z|)
(Intercept)    -8.346982   1.568100  -5.323 0.000000102 ***
courseElite     1.518775   0.576107   2.636     0.00838 **
courseMainstrm  0.480233   0.346050   1.388     0.16521
hsgpa           0.100440   0.024862   4.040 0.000053467 ***
hsengl          0.002194   0.015501   0.142     0.88745
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 576.28  on 419  degrees of freedom
Residual deviance: 527.64  on 415  degrees of freedom
AIC: 537.64

Number of Fisher Scoring iterations: 5
```