Logistic Regression

Joshua L. Eubanks (joshua.eubanks@ucf.edu)

2023-11-18

Contents

You'll need these packages installed to run the code: `install.packages(c('AER', 'ggplot2' ,'robust', 'qcc'))`

1 What is Logistic Regression?

Logistic regression models binary (0 or 1), (true or false) outcomes.

Some examples are:

- Will this person pay their bills or default?
- Is this a positive or negative review?
- Is the author a democrat or republican?
- Will I pass or fail the class?

You can even break down some numerical values to binary. Ex: will the company be profitable or at a loss?

2 Why do we need logistic regression?

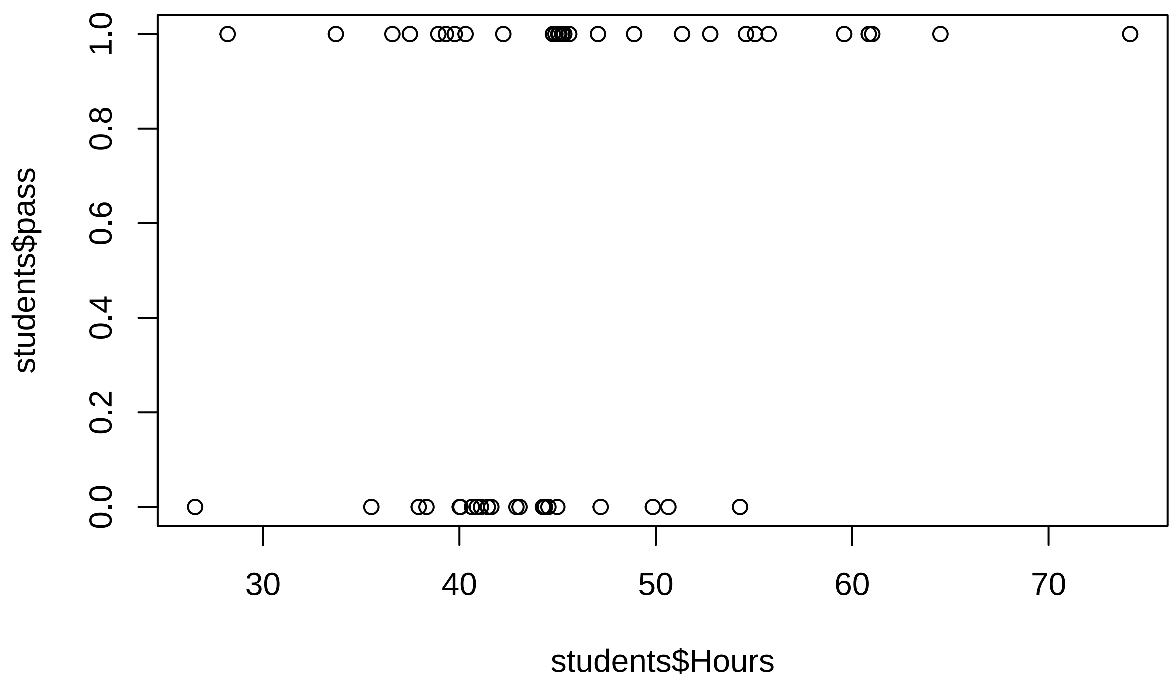
Suppose we use the student's pass or fail example.

```r
# Making some made up data

set.seed(1234)
studyHours <- rnorm(50, mean = 50, sd = 10)
pass <- round(runif(50))

students <- data.frame("pass" = pass,
                       "Hours" = studyHours)
students$pass[students$Hours >= 55] <- 1
```

```r
plot(students$Hours,students$pass)
```



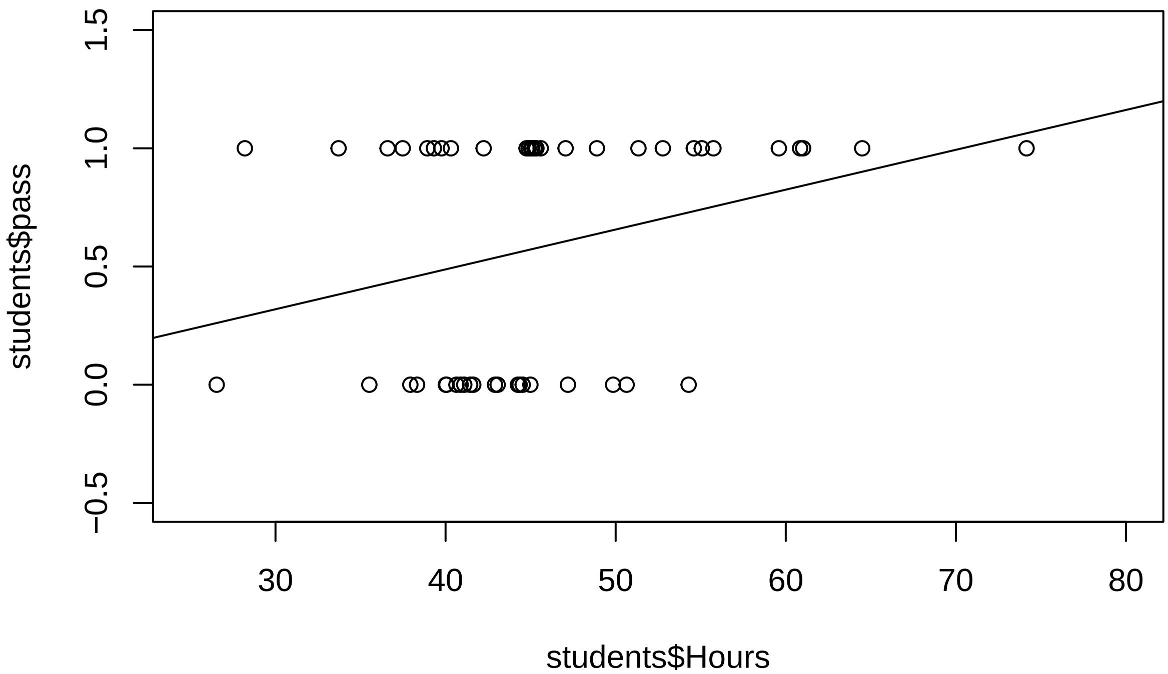
In this case, what would happen if we ran a basic regression?

```r
fit <- lm(pass ~ Hours, data = students )

summary(fit)
```

```
##
## Call:
## lm(formula = pass ~ Hours, data = students)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7287 -0.5054  0.1714  0.4258  0.7111
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.186398   0.359261  -0.519   0.6063
## Hours        0.016855   0.007758   2.173   0.0348 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4807 on 48 degrees of freedom
## Multiple R-squared:  0.08953,    Adjusted R-squared:  0.07056
## F-statistic:  4.72 on 1 and 48 DF,  p-value: 0.03479
```

```r
plot(x=students$Hours,y=students$pass,ylim=c(-.5,1.5),xlim=c(25,80))
abline(fit)
```



Interpreting the coefficents doesn't make sense. Additionally, the fit goes outside the probability limits [0,1].

What we can use is a logit link function:

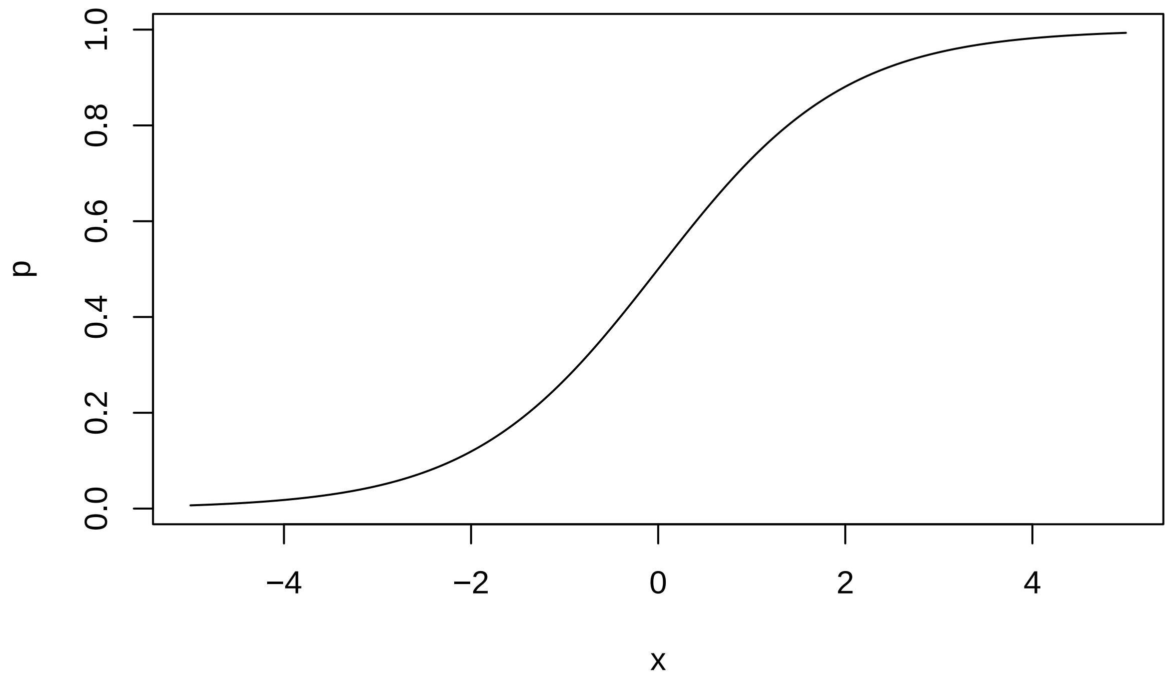
$p(y=1|x_1...x_k) = \frac{\exp[\beta_0+\beta_1 x_1+...+\beta_k x_k]}{1+\exp[\beta_0+\beta_1 x_1+...+\beta_k x_k]}$

As you can see from this plot, we are bound between zero and one:

```r
x <- seq(from =-5, to = 5, by = .001)

p <- exp(x)/(1+exp(x))

plot(x,p,type = "l")
```



3 Interpreting the coefficients

With some algebra, we can write the regression equation as:

$\log \left[\frac{p}{1-p} \right] = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k$

This means that the logistic regression is the **linear model for log odds**

4 Fitting the model

```
fitStudent <- glm(pass ~ Hours, data = students, family = 'binomial')

summary(fitStudent)
```

```
##
## Call:
## glm(formula = pass ~ Hours, family = "binomial", data = students)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6693  -1.1811   0.6066   1.0339   1.6467
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.3983     1.8678  -1.819   0.0688 .
```

```
## Hours         0.0830     0.0417   1.990   0.0466 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 68.029  on 49  degrees of freedom
## Residual deviance: 63.100  on 48  degrees of freedom
## AIC: 67.1
##
## Number of Fisher Scoring iterations: 3
```

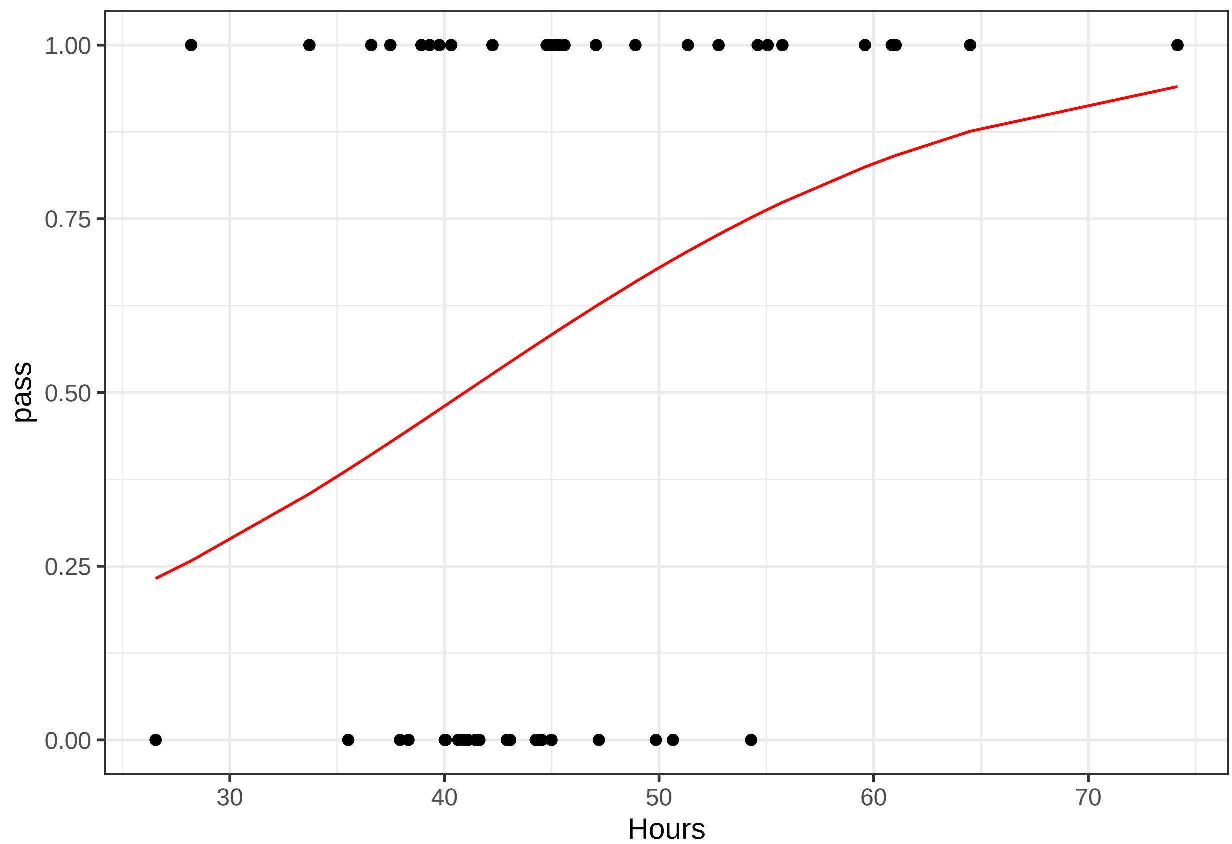
5 Prediction

We can pass parameters into the predict function just as before. One thing to change is `type = "response"`. This will print out the probabilities instead of the log odds.

```r
students$Probability <- predict(fitStudent,
                                newdata = data.frame("Hours" = students$Hours),
                                type = "response")
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```r
ggplot(data = students, aes(Hours,pass))+
  geom_point()+
  geom_line(data = students, aes(Hours,Probability), color = 'red')+
  theme_bw()
```



6 Affairs Example

Let's load a dataset of information about peoples' engagement in extramarital affairs, anonymously collected, of course.

```
data(Affairs, package = "AER")
```

Always start by investigating the properties of the dataset. Calculate the summary statistics.

```
summary(Affairs)
```

```
##     affairs          gender         age         yearsmarried    children
##  Min.   : 0.000   female:315   Min.   :17.50   Min.   : 0.125   no :171
##  1st Qu.: 0.000   male  :286   1st Qu.:27.00   1st Qu.: 4.000   yes:430
##  Median : 0.000                Median :32.00   Median : 7.000
##  Mean   : 1.456                Mean   :32.49   Mean   : 8.178
##  3rd Qu.: 0.000                3rd Qu.:37.00   3rd Qu.:15.000
##  Max.   :12.000                Max.   :57.00   Max.   :15.000
##  religiousness     education       occupation        rating
##  Min.   :1.000   Min.   : 9.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:14.00   1st Qu.:3.000   1st Qu.:3.000
##  Median :3.000   Median :16.00   Median :5.000   Median :4.000
##  Mean   :3.116   Mean   :16.17   Mean   :4.195   Mean   :3.932
##  3rd Qu.:4.000   3rd Qu.:18.00   3rd Qu.:6.000   3rd Qu.:5.000
##  Max.   :5.000   Max.   :20.00   Max.   :7.000   Max.   :5.000
```

```r
table(Affairs$affairs)
```

```
##
##   0   1   2   3   7  12
## 451  34  17  19  42  38
```

Notice that the majority report never having such an affair. Although, several report numbers as high as 12.

To indicate faithfulness, create a binary outcome variable that indicates whether a subject has ever had an affair.

```r
Affairs$ynaffair[Affairs$affairs > 0] <- 1
Affairs$ynaffair[Affairs$affairs == 0] <- 0
# Define this as a factor with two levels.
Affairs$ynaffair <- factor(Affairs$ynaffair,
                           levels = c(0, 1),
                           labels = c("No", "Yes"))
table(Affairs$ynaffair)
```

```
##
##  No Yes
## 451 150
```

Start by fitting the full model, with all available variables.

```r
fit.full <- glm(ynaffair ~ gender + age + yearsmarried +
    children + religiousness + education + occupation + rating,
    data = Affairs, family = binomial())
summary(fit.full)
```

```
##
## Call:
## glm(formula = ynaffair ~ gender + age + yearsmarried + children +
##     religiousness + education + occupation + rating, family = binomial(),
##     data = Affairs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5713  -0.7499  -0.5690  -0.2539   2.5191
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.37726    0.88776   1.551 0.120807
## gendermale     0.28029    0.23909   1.172 0.241083
## age           -0.04426    0.01825  -2.425 0.015301 *
## yearsmarried   0.09477    0.03221   2.942 0.003262 **
## childrenyes    0.39767    0.29151   1.364 0.172508
## religiousness -0.32472    0.08975  -3.618 0.000297 ***
## education      0.02105    0.05051   0.417 0.676851
## occupation     0.03092    0.07178   0.431 0.666630
## rating        -0.46845    0.09091  -5.153 2.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.38  on 600  degrees of freedom
```

```
## Residual deviance: 609.51  on 592  degrees of freedom
## AIC: 627.51
##
## Number of Fisher Scoring iterations: 4
```

Notice that several variables are not statistically significant. Consider removing one or more and fitting a reduced model. Normally, you would consider a sequence of small changes but for this demonstration, we will make one big change by dropping several variables.

```r
fit.reduced <- glm(ynaffair ~ age + yearsmarried +
    religiousness + rating, data = Affairs, family = binomial())
summary(fit.reduced)
```

```
##
## Call:
## glm(formula = ynaffair ~ age + yearsmarried + religiousness +
##     rating, family = binomial(), data = Affairs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6278  -0.7550  -0.5701  -0.2624   2.3998
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.93083    0.61032   3.164 0.001558 **
## age           -0.03527    0.01736  -2.032 0.042127 *
## yearsmarried   0.10062    0.02921   3.445 0.000571 ***
## religiousness -0.32902    0.08945  -3.678 0.000235 ***
## rating        -0.46136    0.08884  -5.193 2.06e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.38  on 600  degrees of freedom
## Residual deviance: 615.36  on 596  degrees of freedom
## AIC: 625.36
##
## Number of Fisher Scoring iterations: 4
```

Now all remaining variables are statistically significant. Compare the two candidate models and test for a statistically significant improvement in fit for the larger model.

```r
anova(fit.reduced, fit.full, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: ynaffair ~ age + yearsmarried + religiousness + rating
## Model 2: ynaffair ~ gender + age + yearsmarried + children + religiousness +
##     education + occupation + rating
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       596     615.36
## 2       592     609.51  4   5.8474   0.2108
```

This jointly tests the exclusion of all the variables dropped in the change above. The high p-value suggests very little is lost by restricting the additional coefficients to zero, which is the same as excluding the variables.

Now that we have settled on a model, consider the interpretation of the coefficients.

```
coef(fit.reduced)
```

```
##   (Intercept)           age  yearsmarried religiousness        rating
##    1.93083017   -0.03527112    0.10062274   -0.32902386   -0.46136144
```

For a logistic regression, the change in estimated probability is approximately proportional, so check the exponential transformation of the coefficients.

```
exp(coef(fit.reduced))
```

```
##   (Intercept)           age  yearsmarried religiousness        rating
##     6.8952321     0.9653437     1.1058594     0.7196258     0.6304248
```

Now analyze the model predictions directly, which is a more reliable way to investigate the predictions of the model. First, generate a dataset of hypothetical values for the predictions. It includes one row for each level of the marital rating variable and the average values of the other variables.

```
testdata <- data.frame(rating = c(1, 2, 3, 4, 5),
    age = mean(Affairs$age), yearsmarried = mean(Affairs$yearsmarried),
    religiousness = mean(Affairs$religiousness))
```

Calculate the probability of extramarital affair by marital ratings.

```
testdata$prob <- predict(fit.reduced, newdata = testdata,
    type = "response")
```

The “response” type returns the predictions in terms of the probability that an affair would occur.

```
testdata
```

```
##   rating      age yearsmarried religiousness      prob
## 1      1 32.48752     8.177696      3.116473 0.5302296
## 2      2 32.48752     8.177696      3.116473 0.4157377
## 3      3 32.48752     8.177696      3.116473 0.3096712
## 4      4 32.48752     8.177696      3.116473 0.2204547
## 5      5 32.48752     8.177696      3.116473 0.1513079
```

For the selected values of the other variables, we can see that the probability of an affair increases as the marital rating declines. Now repeat the calculation for the age variable. The prediction dataset has average values of the other variable but selected levels of the age variable.

```
testdata <- data.frame(rating = mean(Affairs$rating),
    age = seq(17, 57, 10), yearsmarried = mean(Affairs$yearsmarried),
    religiousness = mean(Affairs$religiousness))
```

Calculate probabilities of extramarital affair by age

```
testdata$prob <- predict(fit.reduced, newdata = testdata,
    type = "response")
testdata
```

```
##    rating age yearsmarried religiousness      prob
## 1 3.93178  17     8.177696      3.116473 0.3350834
## 2 3.93178  27     8.177696      3.116473 0.2615373
## 3 3.93178  37     8.177696      3.116473 0.1992953
## 4 3.93178  47     8.177696      3.116473 0.1488796
## 5 3.93178  57     8.177696      3.116473 0.1094738
```

The probability of an affair decreases as people age.

Let's tests the length of the marriage now.

```
testdata <- data.frame(rating = mean(Affairs$rating),
    age = mean(Affairs$age), yearsmarried = 1:5,
    religiousness = mean(Affairs$religiousness))
```

Calculate probabilities of extramarital affair by years married.

```
testdata$prob <- predict(fit.reduced, newdata = testdata,
    type = "response")
testdata
```

```
##    rating      age yearsmarried religiousness      prob
## 1 3.93178 32.48752            1      3.116473 0.1241413
## 2 3.93178 32.48752            2      3.116473 0.1355022
## 3 3.93178 32.48752            3      3.116473 0.1477273
## 4 3.93178 32.48752            4      3.116473 0.1608502
## 5 3.93178 32.48752            5      3.116473 0.1748996
```