Basic Course on R: Logistic Regression

Elizabeth Ribble*

28 Oct - 1 Nov 2019

Contents

*emcclel3@msudenver.edu

Most of the following examples use the data "R_data_January2015.csv" which contains variables on mothers whose babies are either intellectually disabled or developmentally normal.

```
babies <- read.csv("R_data_January2015.csv", header = T, row.names = 1)
names(babies)
```

```
##  [1] "Status"                "iodine_deficiency"
##  [3] "BMI"                   "educational_level"
##  [5] "alcohol"               "smoking"
##  [7] "medication"            "birthweight"
##  [9] "pregnancy_length_weeks" "pregnancy_length_days"
## [11] "SAM"                   "SAH"
## [13] "homocysteine"          "cholesterol"
## [15] "HDL"                   "triglycerides"
## [17] "vitaminB12"            "folicacid_serum"
## [19] "folicacid_erys"
```

```
attach(babies)
```

```
## The following objects are masked from babydata (pos = 3):
##
##    alcohol, birthweight, BMI, cholesterol, educational_level,
##    folicacid_erys, folicacid_serum, HDL, homocysteine,
##    iodine_deficiency, medication, pregnancy_length_days,
##    pregnancy_length_weeks, SAH, SAM, smoking, Status,
##    triglycerides, vitaminB12
## The following objects are masked from babies (pos = 4):
##
##    alcohol, birthweight, BMI, cholesterol, educational_level,
##    folicacid_erys, folicacid_serum, HDL, homocysteine,
##    iodine_deficiency, medication, pregnancy_length_days,
##    pregnancy_length_weeks, SAH, SAM, smoking, Status,
##    triglycerides, vitaminB12
## The following objects are masked from babydata (pos = 9):
##
##    alcohol, birthweight, BMI, cholesterol, educational_level,
##    folicacid_erys, folicacid_serum, HDL, homocysteine,
##    iodine_deficiency, medication, pregnancy_length_days,
##    pregnancy_length_weeks, SAH, SAM, smoking, Status,
##    triglycerides, vitaminB12
## The following objects are masked from babies (pos = 11):
##
```

```
##     alcohol, birthweight, BMI, cholesterol, educational_level,
##     folicacid_erys, folicacid_serum, HDL, homocysteine,
##     iodine_deficiency, medication, pregnancy_length_days,
##     pregnancy_length_weeks, SAH, SAM, smoking, Status,
##     triglycerides, vitaminB12
## The following objects are masked from babydata (pos = 12):
##
##     alcohol, birthweight, BMI, cholesterol, educational_level,
##     folicacid_erys, folicacid_serum, HDL, homocysteine,
##     iodine_deficiency, medication, pregnancy_length_days,
##     pregnancy_length_weeks, SAH, SAM, smoking, Status,
##     triglycerides, vitaminB12
```

1 When to Use Logistic Regression

There are many research topics for with the dependent variable y is binary (0/1), e.g.

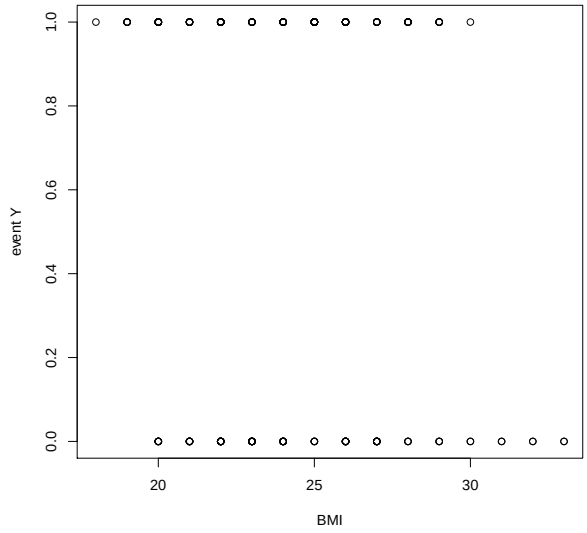
- mortality (dead/alive)
- treatment response (responder/non-responder)
- development of disease (yes/no)

and we want to predict the membership of an individual to one of the two categories based on a set of predictors.

In this situation we have to work with **probabilities**, which are numbers between 0 and 1. A value $P(y)$ that is close to **0** means that y is very **unlikely** to occur, while a value close to **1** means that y is very **likely** to occur.

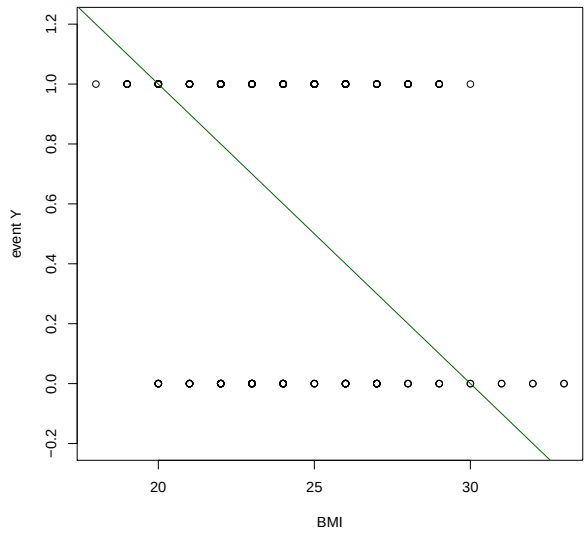
In simple/multiple linear regression a continuous variable y is predicted by continuous/categorical x(s), but if y can only have 2 values (e.g. $y = 0$ or $y = 1$), how do we predict the probability that $y = 1$ given one or more predictors? Could we apply a linear regression model...?

```r
plot(BMI, as.numeric(Status)-1, ylab = "event Y")
```



If we try to fit a **linear** regression model...

```r
plot(BMI, as.numeric(Status)-1, ylab = "event Y", ylim = c(-.2,1.2))
abline(3, -.1, col = "darkgreen")
```



we would try to fit $P(y = 1) = b_0 + b_1 x$, which doesn't work. In linear regression we assume the relationship between x and y is linear, but in the case of binary outcomes this assumption is no longer valid. Results obtained from a linear regression model wouldn't makes sense! Probabilities beyond the interval (0,1) are not interpretable.

Instead we're going to use a logistic curve! First, note that our factor `Status` has levels that are ordered alphabetically:

```
levels(Status)

## [1] "intellectual disability"  "normal brain development"

Status[1:3]

## [1] intellectual disability  normal brain development
## [3] intellectual disability
## Levels: intellectual disability normal brain development

as.numeric(Status)[1:3]

## [1] 1 2 1
```

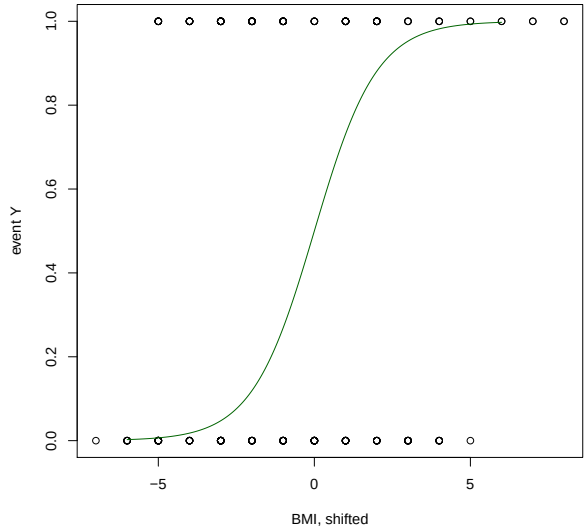
But we would prefer the non-event to be the baseline so we will change `Status` so the event $y = 1$ is intellectual disability and the non-event $y = 0$ is normal brain development. Note that this re-coding does not change the variable in `babies`).

```
newstatus <- 2-as.numeric(Status)
newstatus[1:3]

## [1] 1 0 1
```

So now our visualization should make more sense:

```
plot(BMI-25, newstatus, ylab = "event Y",xlab = "BMI, shifted")
x <- seq(-6, 6, 0.01)
lines(x, exp(x)/(1+exp(x)), type = "l", col = "darkgreen", cex = 2)
```



The forumula we used for the line is `exp(x)/(1+exp(x))`, which is called the logistic function and it is the probability of an event, $P(y = 1)$. Let's see how to derive it...

2 Odds

We're going to start by introducing odds. The odds is the ratio of the probability that the event of interest occurs to the probability that it does not:

$$\text{odds} = \frac{P(\text{event})}{P(\text{no event})} \Leftrightarrow P(\text{event}) = \frac{\text{odds}}{1 + \text{odds}}$$

and note that P(no event)=1-P(event). Describing the probability of event y in terms of odds has a very convenient property:

- as odds increases, $p(y = 1)$ approaches 1
- as odds decreases, $p(y = 1)$ approaches 0

The odds ratio (OR) is a way of comparing whether the probability of a certain event is the same for two groups.

Let's go back to our example from earlier:

	event	no event	total
treatment	20	80	100
placebo	50	50	100
total	70	130	200

then

Total Group			
$P(y=1)$	$= x/n$	$= 70/200$	$= 0.35$
$\text{odds}(y=1)$	$= p/(1-p)$	$= 0.35/0.65$	$= 0.54$
Treated Group			
$P_1(y=1)$	$= x_1/n_1$	$= 20/100$	$= 0.20$
$\text{odds}_1(y=1)$	$= p_1/(1-p_1)$	$= 0.20/0.80$	$= 0.25$
Placebo Group			
$P_0(y=1)$	$= x_0/n_0$	$= 50/100$	$= 0.50$
$\text{odds}_0(y=1)$	$= p_0/(1-p_0)$	$= 0.50/0.50$	$= 1.00$

So $\text{OR}(\text{event}) = \text{odds}_1/\text{odds}_0 = 0.25/1.00 = 0.25$. Thus the odds of an event in the treatment group is 0.25 times lower than the odds of an event in the placebo group. Conversely, $1/0.25 = 4$, so we can also say the odds of an event in the placebo group is 4 times higher than the odds of an event in the treatment group. In R:

```
treatment <- c(20, 80)
placebo <- c(50, 50)
mytable <- rbind(treatment, placebo)
mytable[1,1]*mytable[2, 2]/(mytable[1, 2]*mytable[2, 1])

## treatment
##      0.25
```

3 Simple Logistic Regression Model

We are going to use odds in the following way, where $p = P(y = 1)$ is the probability of an event:

$$\ln\left(\frac{p}{1-p}\right) = b_0 + b_1 x$$

The function $\ln(p/(1-p))$ is called a logit of p and it is this function of y that is linear in x instead of y itself.

This model formulation assures that the predicted probability of an event falls between 0 and 1, unlike a linear regression model. Note:

$$\ln\left(\frac{p}{1-p}\right) = b_0 + b_1 x \Leftrightarrow p = \frac{e^{(b_0+b_1 x)}}{1+e^{(b_0+b_1 x)}}$$

so we've completed our derivation of the logistic function formula from the beginning.

Also note that we've made no assumptions about linearity, normality or homoscedasticity!

The values of the intercept and slope are estimated using the maximum likelihood method, which finds the values of the coefficients that make the observed data most likely to occur.

Statistical significance of estimate coefficients is testing with the Wald test, which is based on the χ^2 distribution (though R calls it "z"). The goodness of fit is assessed by deviance, which is based on the differences observed-expected principle. Also, the Akaike information criterion (AIC) gives a measure of the quality of the model.

The coefficients are most usefully interpreted with the following:

$$e^{b_1} = \frac{\text{odds after a unit change in the predictor}}{\text{original odds}}$$

and when x is binary, e^{b_1} is the odds ratio from the 2 by 2 contingency table!

In R:

```
lr1 <- glm(newstatus ~ BMI, family = binomial(logit))
summary(lr1)

##
## Call:
## glm(formula = newstatus ~ BMI, family = binomial(logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2759  -1.0805  -0.9168   1.2680   1.4627
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.40591    1.18095  -2.037   0.0416 *
```

```
## BMI          0.08782    0.04822   1.821   0.0685 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 259.83  on 189  degrees of freedom
## Residual deviance: 256.44  on 188  degrees of freedom
## AIC: 260.44
##
## Number of Fisher Scoring iterations: 4
```

so $\text{logit}(p) = -2.40591 + 0.08782 * \text{BMI}$. Thus the probability of having a baby with an intellectual disability when BMI is large, say 32, is:

```
logit_p1 <- -2.40591+0.08782*32
logit_p1

## [1] 0.40433

## or
predict(lr1, newdata = data.frame(BMI = 32), se.fit = TRUE)

## $fit
##         1
## 0.4043502
##
## $se.fit
## [1] 0.3995839
##
## $residual.scale
## [1] 1

p1 <- exp(logit_p1)/(1 + exp(logit_p1))
p1

## [1] 0.5997275
```

The coefficient $b_1 = 0.08782$ can be exponentiated to obtain the odds ratio:

```
summary(lr1)$coef
```

```
##                Estimate Std. Error   z value   Pr(>|z|)
## (Intercept) -2.40591055 1.18095490 -2.037259 0.04162413
## BMI          0.08782065 0.04821621  1.821393 0.06854720
```

```
b1 <- summary(lr1)$coef[2, 1]
b1
```

```
## [1] 0.08782065
```

```
exp(b1)
```

```
## [1] 1.091792
```

Like multiple linear regression, we can add variables into the model here as well:

```
lr2 <- glm(newstatus ~ BMI + smoking, family = binomial(logit))
summary(lr2)$coef
```

```
##                Estimate Std. Error   z value    Pr(>|z|)
## (Intercept) -2.52420359 1.22600372 -2.058887 0.039505027
## BMI          0.08128428 0.04997051  1.626645 0.103812552
## smokingyes   1.34305855 0.38824022  3.459349 0.000541482
```

Thus the probability of having a baby with an intellectual disability when BMI is large, say 32, and the mother is a smoker is:

```
mynewdata <- data.frame(BMI = 32, smoking = factor("yes"))
logit_p2 <- predict(lr2, newdata = mynewdata)
p2 <- exp(logit_p2)/(1+exp(logit_p2))
p2
```

```
##         1
## 0.8053309
```