



UNIVERSITY *of*  
WEST FLORIDA

## Binary Logistic Regression

Statistics for Data Science II

Suppose we now have outcomes that are binary (only two possible responses).

For example, suppose  $Y_i$  was “the student passes the class,” then:

$$Y_i = \begin{cases} 1 & \text{if student passes} \\ 0 & \text{if student does not pass} \end{cases}$$

Binary variables do not always take on yes/no answers!

e.g., “Do you prefer cats or dogs?”

e.g., “Is the pug fawn or black?”

---

We model binary outcomes using logistic regression.

$$\ln \left( \frac{\pi}{1 - \pi} \right) = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k,$$

where  $\pi = P[Y = 1]$  = the probability of the outcome.

How is this different from linear regression?

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k$$

---

Why can't we use OLS estimation?

1. The residuals are not normally distributed.
  2. The residuals do not have constant variance.
  3. The predicted values (probabilities) do not always fall between 0 and 1, the only possible values for the probability of success.
-

Recall the binary logistic regression model,

$$\ln \left( \frac{\pi}{1 - \pi} \right) = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k,$$

We will specify this in R using the `glm()` function, specifying `family = "binomial"`.

e.g., `glm(outcome ~ predictor1 + predictor2 + ..., data = dataset,  
family = "binomial")`

The binomial distribution is used for 0/1 outcomes, thus, is why we specify it here.

---

## Example:

A researcher is interested in how the GRE, college GPA, and prestige of the undergraduate institution affect admission into graduate school. The response variable, admit/don't admit, is a binary variable. Let's model graduate school admission as a function of GRE, college GPA, and prestige of the undergraduate institution.

```
m1 <- glm(admit ~ gre + gpa + rank, data = data, family = "binomial")
```

## Example:

```
summary(m1) [12]
```

```
## $coefficients
##              Estimate Std. Error  z value    Pr(>|z|)
## (Intercept) -3.44954840 1.132846009 -3.045029 2.326583e-03
## gre          0.00229396 0.001091839  2.101005 3.564052e-02
## gpa          0.77701357 0.327483878  2.372677 1.765968e-02
## rank        -0.56003139 0.127136989 -4.404945 1.058109e-05
```

The resulting model is

$$\ln \left( \frac{\hat{\pi}}{1 - \hat{\pi}} \right) = -3.45 + 0.002 \text{ GRE} + 0.78 \text{ GPA} - 0.56 \text{ rank},$$

where  $\hat{\pi}$  is the probability of admittance to graduate school.

Recall the binary logistic regression model,

$$\ln \left( \frac{\pi}{1 - \pi} \right) = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k,$$

We are modeling the log odds, which are not intuitive with interpretations.

To be able to discuss the odds, we will “undo” the natural log by exponentiation.

i.e., if we want to interpret the slope for  $X_i$ , we will look at  $e^{\hat{\beta}_i}$ .

When interpreting  $\hat{\beta}_i$ , it is an additive effect on the log odds.

When interpreting  $e^{\hat{\beta}_i}$ , it is a multiplicative effect on the odds.

---



Why is it a multiplicative effect?

$$\begin{aligned}\ln\left(\frac{\pi}{1-\pi}\right) &= \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k \\ \exp\left\{\ln\left(\frac{\pi}{1-\pi}\right)\right\} &= \exp\{\beta_0 + \beta_1 X_1 + \dots + \beta_k X_k\} \\ \frac{\pi}{1-\pi} &= e^{\beta_0} e^{\beta_1 X_1} \dots e^{\beta_k X_k}\end{aligned}$$

In linear regression, we interpret  $\hat{\beta}_i$ :

For a 1 [unit of predictor] increase in [predictor name], the [outcome] to [increases or decreases] by  $[|\hat{\beta}_i|]$  [unit of outcome].

In logistic regression, we interpret  $e^{\hat{\beta}_i}$  (the odds ratio):

For a 1 [unit of predictor] increase in [predictor name], the odds of [outcome] to be multiplied by  $[e^{\hat{\beta}_i}]$ .

For a 1 [unit of predictor] increase in [predictor name], the odds of [outcome] are [increased or decreased] by  $[100(e^{\hat{\beta}_i}-1)\% \text{ or } 100(1-e^{\hat{\beta}_i})\%]$ .

---

## Example:

Convert all  $\hat{\beta}_i$  to odds ratios and provide brief interpretations for the graduate school admissions data.

```
round(exp(coefficients(m1)), 4)
```

## (Intercept)	gre	gpa	rank
## 0.0318	1.0023	2.1750	0.5712

## Example:

##	(Intercept)	gre	gpa	rank
##	0.0318	1.0023	2.1750	0.5712

For a 1 point increase in GRE score, the odds of admission increase by .23%.

For a 1 point increase in GPA, the odds of admission increase by 118%.

For a 1 point increase in rank of undergraduate institution, the odds of admission decrease by 43%.

## Example:

Suppose we turn the rank of undergraduate institution into a factor variable and use that as the only predictor of admission to graduate school. Let a top tier (rank = 1) undergraduate institution be the reference.

```
data <- dummy_cols(data, select_columns = "rank")
m2 <- glm(admit ~ rank_2 + rank_3 + rank_4, data = data, family = "binomial")
round(exp(coefficients(m2)), 4)
```

## (Intercept)	rank_2	rank_3	rank_4
## 1.1786	0.4724	0.2555	0.1851

The odds of someone from a 2nd tier undergraduate institution being admitted to graduate school are 0.47 times that of someone from a top tier undergraduate institution.

## Statistical Test for $\beta_i$

### Hypotheses

$$H_0 : \beta_i = \beta_i^{(0)} \mid \beta_i \geq \beta_i^{(0)} \mid \beta_i \leq \beta_i^{(0)}$$

$$H_1 : \beta_i \neq \beta_i^{(0)} \mid \beta_i < \beta_i^{(0)} \mid \beta_i > \beta_i^{(0)}$$

### Test Statistic

$$z_0 = \frac{\hat{\beta}_i - \beta_i^{(0)}}{SE_{\hat{\beta}_i}}$$

### Rejection Region

Reject  $H_0$  if  $p < \alpha$ .

## Example:

```
summary(m1) [12]
```

```
## $coefficients
```

##	Estimate	Std. Error	z value	Pr(> z )
## (Intercept)	-3.44954840	1.132846009	-3.045029	2.326583e-03
## gre	0.00229396	0.001091839	2.101005	3.564052e-02
## gpa	0.77701357	0.327483878	2.372677	1.765968e-02
## rank	-0.56003139	0.127136989	-4.404945	1.058109e-05

Thus, all are significant predictors of admission to graduate school.

## Confidence Interval for $\beta_i$

$$\hat{\beta}_i \pm z_{1-\alpha/2} SE_{\hat{\beta}_i}$$

### Example:

```
confint(m1)
```

##		2.5 %	97.5 %
##	(Intercept)	-5.7109591680	-1.260314066
##	gre	0.0001715446	0.004461385
##	gpa	0.1415710585	1.428341503
##	rank	-0.8149612229	-0.315479733



We can also find the CI for  $OR_i$  by exponentiating the lower and upper bounds.

**Example:**

```
round(exp(confint(m1)),3)
```

##	2.5 %	97.5 %
## (Intercept)	0.003	0.284
## gre	1.000	1.004
## gpa	1.152	4.172
## rank	0.443	0.729

Recall the logistic regression model,

$$\ln \left( \frac{\pi_i}{1 - \pi_i} \right) = \beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki}$$

We can solve for the probability, which allows us to predict the probability that  $Y_i = 1$  given the specified model:

$$\pi_i = \frac{\exp \{ \beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki} \}}{1 + \exp \{ \beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki} \}}$$

## Example:

```
data$p_hat <- predict(m1, type="response")  
head(data)
```

```
## # A tibble: 6 x 5  
##   admit gre   gpa rank p_hat  
##   <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1     0  380  3.61     3 0.190  
## 2     1  660  3.67     3 0.318  
## 3     1  800    4     1 0.718  
## 4     1  640  3.19     4 0.149  
## 5     0  520  2.93     4 0.0980  
## 6     1  760    3     2 0.379
```

We generally are not worried about residuals in logistic regression.

We can still look at Cook's distance.

Recall that we look for "spikes" on the graph.

This allows us to determine any leverage/influence points.

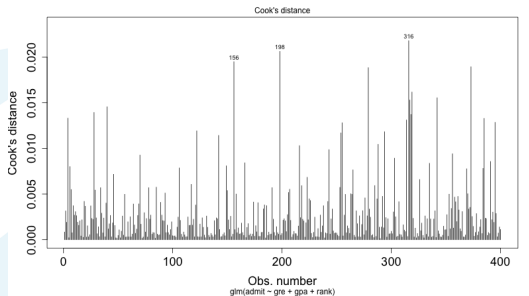
Leverage/influence points are ones that have an effect on the regression model.

If we detect leverage/influence points, we can perform sensitivity analysis to determine how "different" the model is.

---

## Example:

```
plot(m1, which = 4)
```



We can also check for multicollinearity using the VIF.

Recall that  $VIF > 10$  indicates multicollinearity.

**Example:**

```
vif(m1)
```

```
##          gre          gpa          rank  
## 1.135657 1.134708 1.004034
```

## Example:

Let the probability of admittance to graduate school be on the y-axis and GPA be on the x-axis. We will hold the GRE score and rank of undergraduate institution constant by plugging in their median values.

```
c1 <- coefficients(m1)
data$pred_med <- exp(c1[1] + c1[2]*median(data$gre) + c1[3]*data$gpa +
                    c1[4]*median(data$rank))/(1+exp(c1[1] +
                    c1[2]*median(data$gre) + c1[3]*data$gpa +
                    c1[4]*median(data$rank)))
```

## Example:

```
head(data)
```

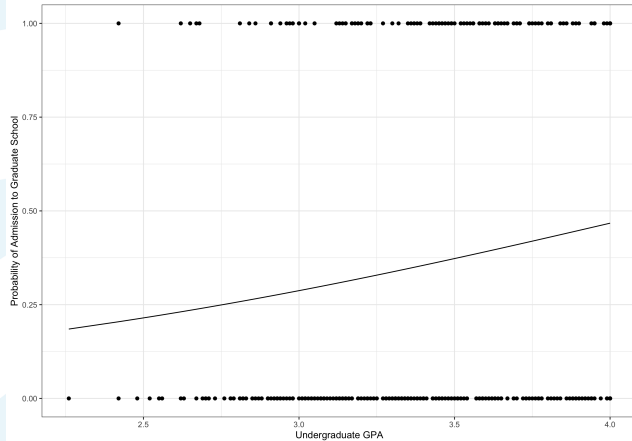
```
## # A tibble: 6 x 6
##   admit   gre   gpa  rank  p_hat pred_med
##   <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1     0   380  3.61     3 0.190   0.393
## 2     1   660  3.67     3 0.318   0.404
## 3     1   800    4     1 0.718   0.467
## 4     1   640  3.19     4 0.149   0.319
## 5     0   520  2.93     4 0.0980  0.276
## 6     1   760    3     2 0.379   0.287
```



## Example:

```
p2 <- data %>% ggplot(aes(x = gpa, y = admit)) +  
  geom_point() +  
  geom_line(aes(y = pred_med))+  
  xlab("Undergraduate GPA") +  
  ylab("Probability of Admission to Graduate School") +  
  theme_bw()
```

## Example:



## Example:

Instead of looking at median values, let's plug in the best possible values for GRE (the maximum) and rank of undergraduate institution (the minimum).

```
data$pred_best <- exp(c1[1] + c1[2]*max(data$gre) + c1[3]*data$gpa +  
                      c1[4]*min(data$rank))/(1+exp(c1[1] +  
                      c1[2]*max(data$gre) + c1[3]*data$gpa +  
                      c1[4]*min(data$rank)))
```

## Example:

```
head(data)
```

```
## # A tibble: 6 x 7
##   admit  gre  gpa  rank  p_hat pred_med pred_best
##   <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>
## 1     0   380  3.61     3 0.190   0.393   0.653
## 2     1   660  3.67     3 0.318   0.404   0.663
## 3     1   800    4     1 0.718   0.467   0.718
## 4     1   640  3.19     4 0.149   0.319   0.575
## 5     0   520  2.93     4 0.0980  0.276   0.526
## 6     1   760    3     2 0.379   0.287   0.539
```

## Example:

```
p3 <- data %>% ggplot(aes(x = gpa, y = admit)) +  
  geom_point() +  
  geom_line(aes(y = pred_best))+  
  xlab("Undergraduate GPA") +  
  ylab("Probability of Admission to Graduate School") +  
  theme_bw()
```

## Example:

