Interpreting logistic models

Frank Edwards

2/27/2023

Quick review

- Remember that logistic regression is a GLM with a logit link function

Quick review

- Remember that logistic regression is a GLM with a logit link function
- A GLM takes the form: $g(y) = \mathrm{X}\beta$

Quick review

- Remember that logistic regression is a GLM with a logit link function
- A GLM takes the form: $g(y) = \mathrm{X}\beta$
- Logistic regression is the special case where g is the logit function: $\mathrm{logit}(y) = \mathrm{X}\beta$

- Remember that logistic regression is a GLM with a logit link function
- A GLM takes the form: $g(y) = \mathrm{X}\beta$
- Logistic regression is the special case where g is the logit function: $\mathrm{logit}(y) = \mathrm{X}\beta$
- A logistic regression model returns $\mathrm{X}\beta$ on the logit scale

Quick review

- Remember that logistic regression is a GLM with a logit link function
- A GLM takes the form: $g(y) = \mathrm{X}\beta$
- Logistic regression is the special case where g is the logit function: $\mathrm{logit}(y) = \mathrm{X}\beta$
- A logistic regression model returns $\mathrm{X}\beta$ on the logit scale
- How can we convert $\mathrm{x}\beta$ to something useful?

Let’s return to the grad school admission example

```r
admits <- read_csv("./data/binary.csv")
summary(admits)
```

```
##      admit             gre             gpa              rank      
##  Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000  
##  Median :0.0000   Median :580.0   Median :3.395   Median :2.000  
##  Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485  
##  3rd Qu.:1.0000   3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000  
##  Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000  
```

Let's look at this as the distribution of the probability of admissions across the data

- First, fit an intercept-only logistic regression model

```
m0 <- glm(admit ~ 1, data = admits, family = "binomial")
m0_est <- tidy(m0)
```

- What does this model tell us?

What does this model tell us?

```r
m0_est$estimate  ## log odds
```

```
## [1] -0.7652847
```

```r
exp(m0_est$estimate)  ## odds
```

```
## [1] 0.4652015
```

```r
exp(m0_est$estimate)/(1 + exp(m0_est$estimate))  ## probability
```

```
## [1] 0.3175
```

```r
mean(admits$admit)  ## mean admission probability
```

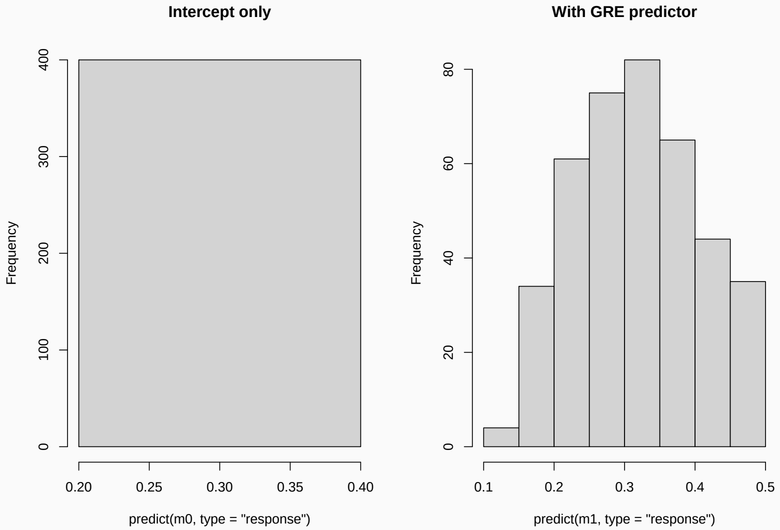
```
## [1] 0.3175
```

Let's add a predictor

```r
m1 <- glm(admit ~ 1 + gre, data = admits, family = "binomial")
tidy(m1)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic    p.value
##   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
## 1 (Intercept) -2.90     0.606        -4.79 0.00000169
## 2 gre          0.00358  0.000986      3.63 0.000280
```

Predictions for all data points in M0 and M1 - what's going on?



Two linear predictors

Why do these generate such different predictions?

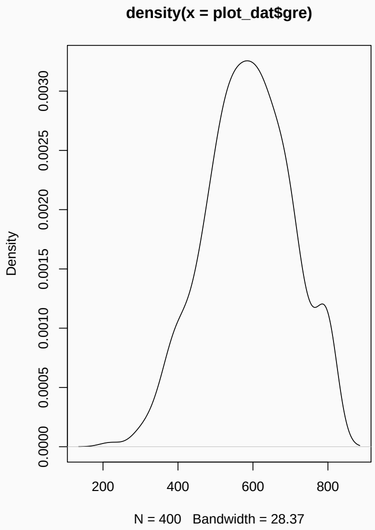
Intercept only model (m0): $p(admit) = \beta_0$

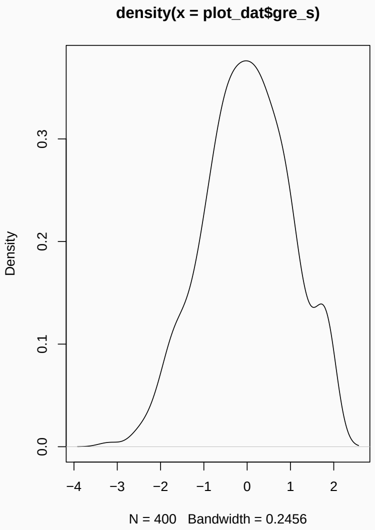
With GRE predictor (m1): $p(admit) = \beta_0 + \beta_1 GRE_i$

To ease interpretation, let's scale GRE

Scale mean-centers and SD scales variables: $\mathrm{scale}(x_i) = \frac{x_i - \bar{x}}{sd(x)}$

Linear transformations of variables: mean-center and SD scale





Re-estimate the model: much nicer to look at

```
admits <- admits %>%
    mutate(gre_s = as.numeric(scale(gre)))
m1 <- glm(admit ~ 1 + gre_s, data = admits, family = "binomial")
m1_est <- tidy(m1)
m1_est
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   -0.796     0.111     -7.20 6.01e-13
## 2 gre_s          0.414     0.114      3.63 2.80e- 4
```

```
m1_alt <- glm(admit ~ scale(gre), data = admits, family = "binomial
```

Interpret the model

```
m1_est
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic  p.value
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)     -0.796     0.111     -7.20 6.01e-13
## 2 gre_s            0.414     0.114      3.63 2.80e- 4
```

Remember: $\mathrm{logit}(y) = \mathrm{X}\beta = \log\left(\frac{y}{1-y}\right)$

So: $y = \mathrm{logit}^{-1}(\mathrm{X}\beta) = \frac{\exp(\mathrm{X}\beta)}{\exp(\mathrm{X}\beta)+1}$

Interpret the model

```
m1_est
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   -0.796     0.111     -7.20 6.01e-13
## 2 gre_s          0.414     0.114      3.63 2.80e- 4
```

Remember: $\mathrm{logit}(y) = \mathrm{X}\beta = \log\left(\frac{y}{1-y}\right)$

So: $y = \mathrm{logit}^{-1}(\mathrm{X}\beta) = \frac{\exp(\mathrm{X}\beta)}{\exp(\mathrm{X}\beta)+1}$

- What is β_0?

Interpret the model

```
m1_est
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   -0.796     0.111     -7.20 6.01e-13
## 2 gre_s          0.414     0.114      3.63 2.80e- 4
```

Remember: $\mathrm{logit}(y) = \mathrm{X}\beta = \log\left(\frac{y}{1-y}\right)$

So: $y = \mathrm{logit}^{-1}(\mathrm{X}\beta) = \frac{\exp(\mathrm{X}\beta)}{\exp(\mathrm{X}\beta)+1}$

- What is β_0?
- What is β_1?

Refresher on exponentials

$$e^{y_1+y_2} = e^{y_1}e^{y_2}$$

Refresher on exponentials

$$e^{y_1+y_2} = e^{y_1}e^{y_2}$$

and

$$e^{y_1-y_2} = \frac{e^{y_1}}{e_2^{y}}$$

Refresher on exponentials

$$e^{y_1+y_2} = e^{y_1}e^{y_2}$$

and

$$e^{y_1-y_2} = \frac{e^{y_1}}{e_2^{y}}$$

so how can we rewrite:

$$\exp(\mathrm{logit}(y)) = \frac{y}{1-y} = e^{\beta_0+\beta_1 x_1}$$

Non-linear relationships

On the log scale, β_0 and β_1 are related to y multiplicatively because

$$e^{\beta_0+\beta_1 x_1} = e^{\beta_0} e^{\beta_1 x_1}$$

Odds

Odds are defined as the probability of the event occurring divided by the probability of probability of the event not occurring. To obtain odds in a logistic regression, we exponentiate both sides:

$$\frac{y}{1-y} = e^{\beta_0+\beta_1 x_1}$$

Odds

Odds are defined as the probability of the event occurring divided by the probability of probability of the event not occurring. To obtain odds in a logistic regression, we exponentiate both sides:

$$\frac{y}{1-y} = e^{\beta_0+\beta_1 x_1}$$

The odds of $y == 1$ are simply $e^{X\beta}$

Odds ratios

The odds ratio is the ratio of two odds - or the proportional change in odds. We can obtain an isolated estimate for the relationship between $\beta_1 x_{1i}$ and y this way:

$$\frac{Odds(y|x_1 = 1)}{Odds(y|x_1 = 0)} = \frac{e^{\beta_0+\beta_1}}{e^{\beta_0}} = \frac{e^{\beta_0} \times e^{\beta_1}}{e^{\beta_0}} = e^{\beta_1}$$

The odds ratio can be interpreted as the change in odds of $y == 1$ for a one-unit change in x_1.

Interpreting odds ratios

- Odds ratios appear convenient - e^{β_1} is a percent change in y for a one-unit change in x_1

Interpreting odds ratios

- Odds ratios appear convenient - e^{β_1} is a percent change in y for a one-unit change in x_1

How do they work?

In our example: what do these figures mean?

```
new_dat <- c(1, 0)  # for scale(gre) == 0, mean score
odds_0 <- exp(new_dat %*% m1_est$estimate)
odds_0
```

```
##           [,1]
## [1,] 0.4510945
```

```
new_dat1 <- c(1, 1)
odds_1 <- exp(new_dat1 %*% m1_est$estimate)
odds_1
```

```
##           [,1]
## [1,] 0.6823082
```

```
odds_1/odds_0  # odds ratio
```

```
##          [,1]
## [1,] 1.512562
```

```
exp(m1_est$estimate[2])  # exp(beta_1)
```

```
## [1] 1.512562
```

The odds of admission are 1.51 times higher for a student with a GRE score one standard deviation above the mean than they are for a student with a mean GRE score.

Interpreting the odds ratio

The odds of admission are 1.51 times higher for a student with a GRE score one standard deviation above the mean than they are for a student with a mean GRE score.

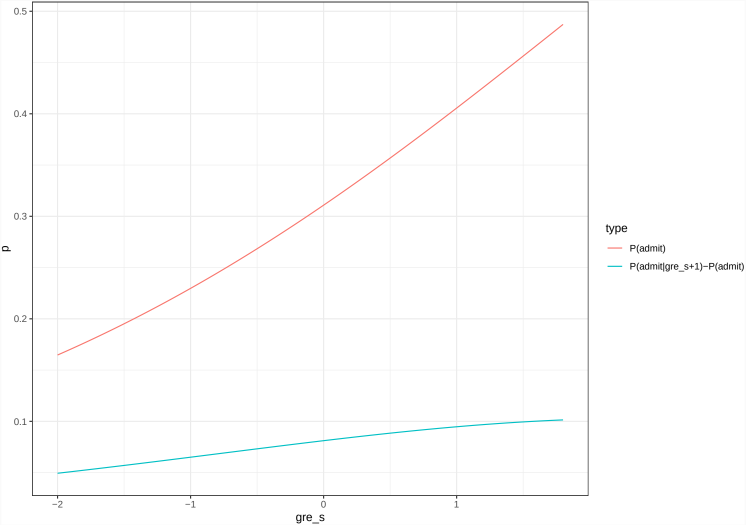
Any trouble you can anticipate here?

Interpreting the odds ratio

The odds of admission are 1.51 times higher for a student with a GRE score one standard deviation above the mean than they are for a student with a mean GRE score.

Any trouble you can anticipate here?

A visual example: the “effect” of 1 SD increase in GRE scores on Pr(admit==1)



It is easy enough to work on the probability scale

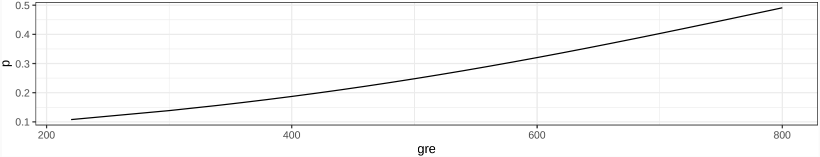
To obtain predicted probabilities of the observed:

- p_hat<-predict(m1, type = “response”)

On the probability scale

```r
preds <- predict(m1, type = "response")
p_hat <- data.frame(gre = admits$gre, p = preds)

ggplot(p_hat, aes(x = gre, y = p)) + geom_line()
```



The basic logic of prediction

1. Choose scenarios of theoretical interest
2. Define these in terms of “counterfactual” (fake) data
3. Plug these fake data into the linear predictor (regression equation)
4. Visualize!

The basic logic of prediction

Reminder: our model is

$$logit(p(admit_i)) = \beta_0 + \beta_1 GRE_i$$

$$admit_i \sim Binomial(1, p)$$

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

1. Choose scenarios of theoretical interest

Low GRE, average GRE, high GRE

Define these scenarios in R

2. Define these in terms of "counterfactual" (fake) data

```
## Look at the distribution of the data to think about scenarios
mean(admits$gre)
```

```
## [1] 587.7
```

```
sd(admits$gre)
```

```
## [1] 115.5165
```

Define these scenarios in R

2. Define these in terms of “counterfactual” (fake) data

```
## Look at the distribution of the data to think about scenarios
mean(admits$gre)
```

```
## [1] 587.7
```

```
sd(admits$gre)
```

```
## [1] 115.5165
```

Let’s define scenarios at the mean, 1 SD below the mean, and 1 SD above the mean

```
fake_data <- data.frame(gre = c(mean(admits$gre), mean(admits$gre) - sd(admits$gre),
    mean(admits$gre) + sd(admits$gre)))

fake_data
```

```
##        gre
## 1 587.7000
## 2 472.1835
## 3 703.2165
```

Generating expected probabilities

3. Plug these fake data into the linear predictor (regression equation)

Because $logit(p(admit_i)) = \beta_0 + \beta_1 x_i$, we can compute the expected probability of admission for a student with mean GRE scores as

```
coef(m1)
```

```
##  (Intercept)          gre
## -2.901344270  0.003582212
```

```
### mean GRE scenario: linear predictor
exp(-2.9 + 0.0036 * 587.7)/(1 + exp(-2.9 + 0.0036 * 587.7))
```

```
## [1] 0.3133982
```

Generating expected probabilities

3. Plug these fake data into the linear predictor (regression equation)

The `predict()` function makes life very easy here

```
## linear predictor
predict(m1, newdata = fake_data)
```

```
##          1          2          3
## -0.7960785 -1.2098832 -0.3822738
```

```
## probability scale (inverse logit)
predict(m1, newdata = fake_data, type = "response")
```

```
##         1         2         3
## 0.3108650 0.2297217 0.4055786
```

Intrepretation through visuals

4. Visualize!

```
### set up our data frame with predictions for plotting
fake_data <- fake_data %>%
    mutate(p_hat = predict(m1, newdata = fake_data, type = "response"))

ggplot(fake_data, aes(x = gre, y = p_hat)) + geom_point()
```