Interpreting logistic models

Frank Edwards

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: $g(y) = \mathrm{X}\beta$
- For logistic regression, g is the logit function: $\mathrm{logit}(y) = \mathrm{X}\beta$

Quick review

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

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 <- stan_glm(admit ~ 1, data = admits, family = "binomial", refresh = 0)
m0
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      admit ~ 1
##  observations: 400
##  predictors:   1
## ------
##             Median MAD_SD
## (Intercept) -0.8    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

- What does this model tell us?

What does this model tell us?

```r
coef(m0)  ## log odds
```

```
## (Intercept)
##  -0.7697838
```

```r
exp(coef(m0))  ## odds
```

```
## (Intercept)
##   0.4631132
```

```r
exp(coef(m0))/(1 + exp(coef(m0)))  ## probability
```

```
## (Intercept)
##   0.3165259
```

```r
mean(admits$admit)  ## mean proportion admitted
```

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

Let's add a predictor

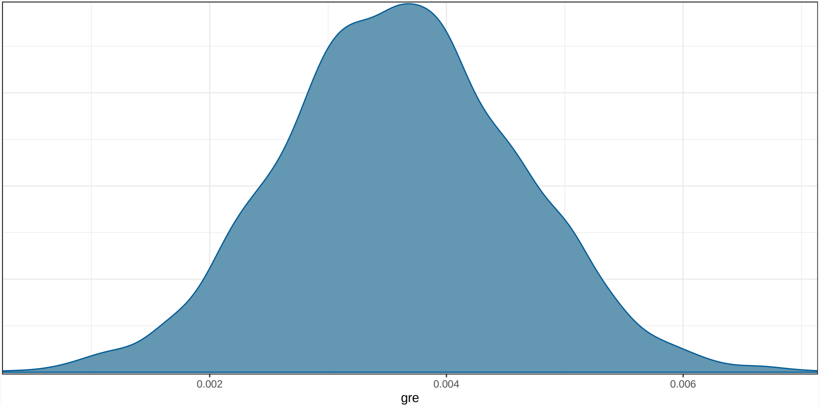
```r
m1 <- stan_glm(admit ~ gre, data = admits, family = "binomial", refresh = 0)
m1
```

```
## stan_glm
##  family:       binomial [logit]
##  formula:      admit ~ gre
##  observations: 400
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) -2.9    0.6
## gre          0.0    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Hmmmm. why is β_1 so small?

Posterior distribution of beta_1: What does this tell us about GRE scores and admission?

```
mcmc_dens(m1, pars = "gre")
```



To ease interpretation, let's scale GRE

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

```
m2 <- stan_glm(admit ~ scale(gre), data = admits, family = "binomial", refresh = 0)
```

Interpret the model

```r
coef(m2)
```

```
## (Intercept)  scale(gre)
##  -0.7968293   0.4154692
```

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

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

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

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

Interpret the model

```r
coef(m2)
```

```
## (Intercept)  scale(gre)
##  -0.7968293   0.4154692
```

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

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

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

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

- What does β_0 mean?

Interpret the model

```r
coef(m2)
```

```
## (Intercept)  scale(gre)
##  -0.7968293   0.4154692
```

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

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

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

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

- What does β_0 mean?
- What does β_1 mean?

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

$$e^{y_1-y_2} = \frac{e^{y_1}}{e^{y}_{2}}$$

Refresher on exponentials

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

and

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

Transforming the model

We can take our logistic regression:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1$$

Transforming the model

We can take our logistic regression:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1$$

Exponentiate both sides and we obtain

$$\left(\frac{p}{1-p}\right) = e^{\beta_0 + \beta_1 x_1}$$

A funny thing happens with our linear predictor when exponentiated

$$\left(\frac{p}{1-p}\right) = e^{\beta_0+\beta_1 x_1} = e^{\beta_0} e^{\beta_1 x_1}$$

A funny thing happens with our linear predictor when exponentiated

$$\left(\frac{p}{1-p}\right) = e^{\beta_0+\beta_1 x_1} = e^{\beta_0} e^{\beta_1 x_1}$$

The odds of p are related to our predictors through a multiplicative, rather than additive relationship

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:

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

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:

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

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:

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

The odds ratio can be interpreted as the change in odds of p for a one-unit change in x_1.

Interpreting odds ratios

- Odds ratios appear convenient - e^{β_1} is a percent change in $p/(1-p)$ for a one-unit change in x_1

Interpreting odds ratios

- Odds ratios appear convenient - e^{β_1} is a percent change in $p/(1-p)$ for a one-unit change in x_1

How do they work?

In our example

```r
coef(m2)
```

```
## (Intercept)  scale(gre)
##  -0.7968293   0.4154692
```

```r
exp(coef(m2))
```

```
## (Intercept)  scale(gre)
##   0.4507559   1.5150815
```

What does β_1 mean? What does e^{β_1} mean?

Interpreting the odds ratio

The odds of admission are 1.52 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 1.52 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?

Expected changes for a 1 SD increase in GRE scores

```
mean(admits$gre)
```

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

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

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

```
# log odds increase for +1 SD GRE
coef(m2)[2]
```

```
## scale(gre)
##  0.4154692
```

```
# odds proportional change for +1 SD GRE
exp(coef(m2)[2])
```

```
## scale(gre)
##   1.515082
```

On the probability scale

```
# change in probability for a low GRE + 1
p_low <- predict(m2, newdata = data.frame(gre = 200), type = "response")
p_lowP1 <- predict(m2, newdata = data.frame(gre = 200 + sd(admits$gre)), type = "response")

p_lowP1 - p_low
```

```
##          1
## 0.04228963
```

On the probability scale

```r
# change in probability for a mid GRE + 1
p_mid <- predict(m2, newdata = data.frame(gre = 500), type = "response")
p_midP1 <- predict(m2, newdata = data.frame(gre = 500 + sd(admits$gre)), type = "response")

p_midP1 - p_mid
```

```
##          1
## 0.08470265
```

On the probability scale

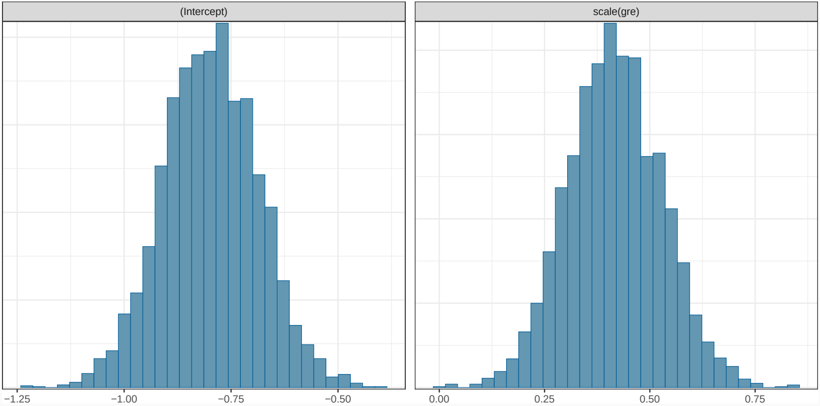
```r
# change in probability for a mid GRE + 1
p_hi <- predict(m2, newdata = data.frame(gre = 650), type = "response")
p_hiP1 <- predict(m2, newdata = data.frame(gre = 650 + sd(admits$gre)), type = "response")

p_hiP1 - p_hi
```

```
##         1
## 0.1003912
```

Bayesian inference with logistic regression

Posterior distributions of Beta parameters

```r
library(bayesplot)
mcmc_hist(m2)
```



To interpret, let's use the linear predictor

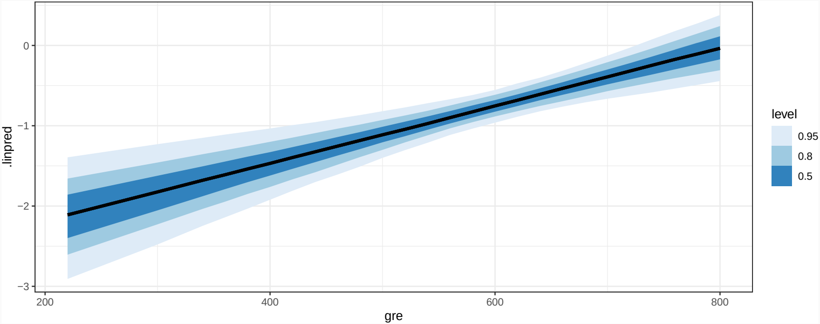
Rather than interpret β_1, let's look at the linear relationship directly

```
library(tidybayes)
m2_lp <- linpred_draws(m2, newdata = admits, ndraws = 1000)

head(m2_lp)
```

```
## # A tibble: 6 x 9
## # Groups:   admit, gre, gpa, rank, .row [1]
##   admit   gre   gpa  rank  .row .chain .iteration .draw .linpred
##   <dbl> <dbl> <dbl> <dbl> <int>  <int>      <int> <int>    <dbl>
## 1     0   380  3.61     3     1     NA         NA     1    -1.09
## 2     0   380  3.61     3     1     NA         NA     2    -1.24
## 3     0   380  3.61     3     1     NA         NA     3    -1.69
## 4     0   380  3.61     3     1     NA         NA     4    -1.56
## 5     0   380  3.61     3     1     NA         NA     5    -1.17
## 6     0   380  3.61     3     1     NA         NA     6    -1.51
```

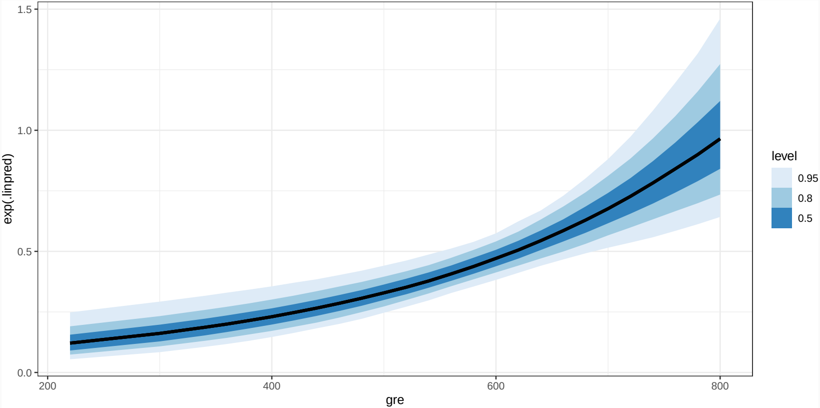
To interpret, let's use the linear predictor

```r
ggplot(m2_lp, aes(x = gre, y = .linpred)) + stat_lineribbon() + scale_fill_brewer()
```



How about on the odds scale

```
ggplot(m2_lp, aes(x = gre, y = exp(.linpred))) + stat_lineribbon() + scale_fill_brewer()
```



We can also estimate expected probability

```
m2_ep <- epred_draws(m2, newdata = admits, ndraws = 1000)

head(m2_ep)
```

```
## # A tibble: 6 x 9
## # Groups:   admit, gre, gpa, rank, .row [1]
##   admit   gre   gpa  rank  .row .chain .iteration .draw .epred
##   <dbl> <dbl> <dbl> <dbl> <int>  <int>      <int> <int>  <dbl>
## 1     0   380  3.61     3     1     NA         NA     1  0.201
## 2     0   380  3.61     3     1     NA         NA     2  0.185
## 3     0   380  3.61     3     1     NA         NA     3  0.172
## 4     0   380  3.61     3     1     NA         NA     4  0.180
## 5     0   380  3.61     3     1     NA         NA     5  0.221
## 6     0   380  3.61     3     1     NA         NA     6  0.199
```

And visualized

```r
ggplot(m2_ep, aes(x = gre, y = .epred)) + stat_lineribbon() + scale_fill_brewer()
```