Logistic regression, 1

Frank Edwards

2/22/2019

Logistic regression

Read in the data for today

```
admissions <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
head(admissions)
```

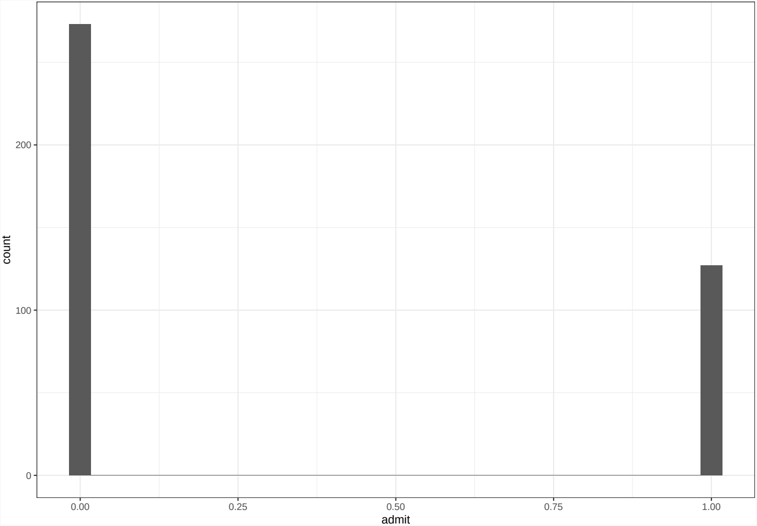
```
##   admit gre  gpa rank
## 1     0 380 3.61    3
## 2     1 660 3.67    3
## 3     1 800 4.00    1
## 4     1 640 3.19    4
## 5     0 520 2.93    4
## 6     1 760 3.00    2
```

```
nrow(admissions)
```

```
## [1] 400
```

Evaluate distribution of binary admission variable

```
ggplot(admissions, aes(x = admit)) + geom_histogram()
```



Properties of Bernoulli variables

If y is an i.i.d. Bernoulli variable with probability p:

$$y \sim Bernoulli(p)$$

$$\Pr(y = 1) = p = 1 - \Pr(y = 0)$$

$$E(y) = \bar{y} = p$$

$$Var(y) = p(1 - p)$$

Summary of admit: What can we say about the probability of admission?

```r
mean(admissions$admit)
```

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

```r
sum(admissions$admit==1)/nrow(admissions)
```

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

```r
var(admissions$admit)
```

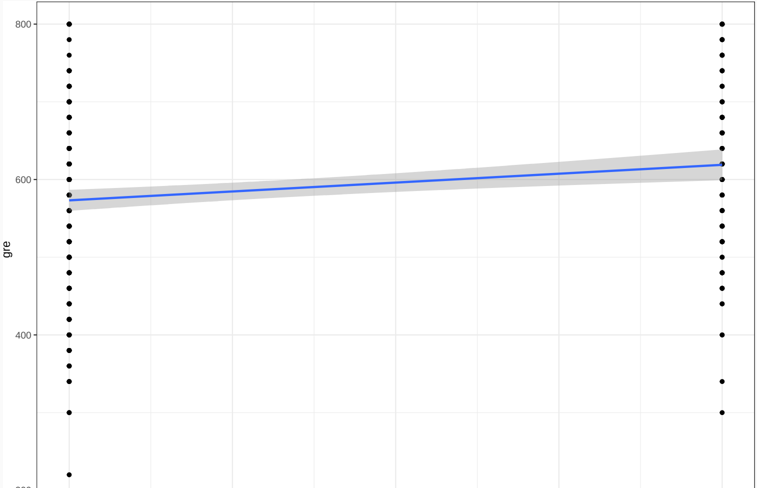
```
## [1] 0.2172368
```

```r
mean(admissions$admit) * (1 - mean(admissions$admit))
```

```
## [1] 0.2166937
```

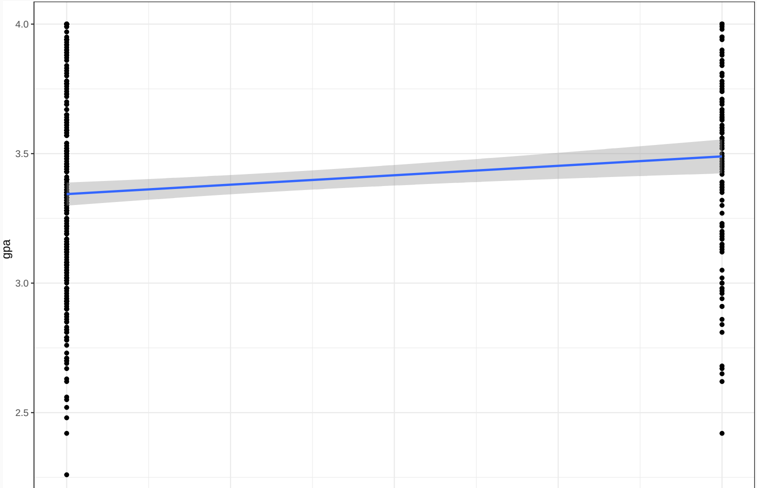
How does GRE relate to admission?

```
ggplot(admissions,
       aes(x = admit, y = gre)) + geom_point() +
  geom_smooth(method = "lm")
```



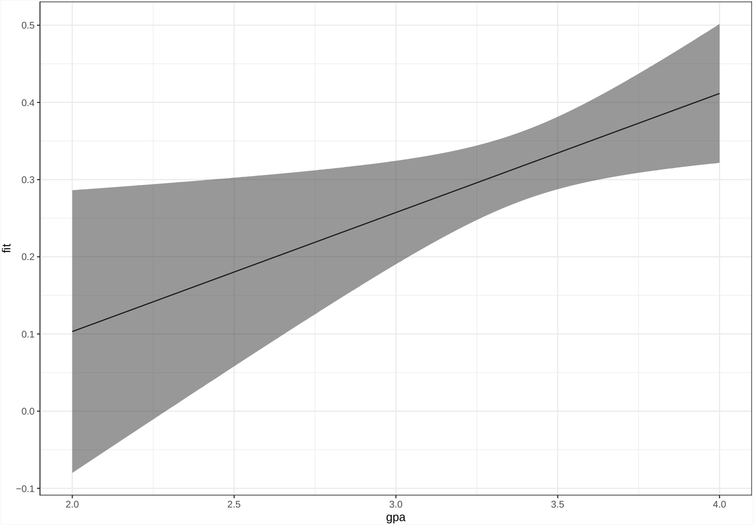
GPA?

```r
ggplot(admissions,
       aes(x = admit, y = gpa)) + geom_point() +
  geom_smooth(method = "lm")
```



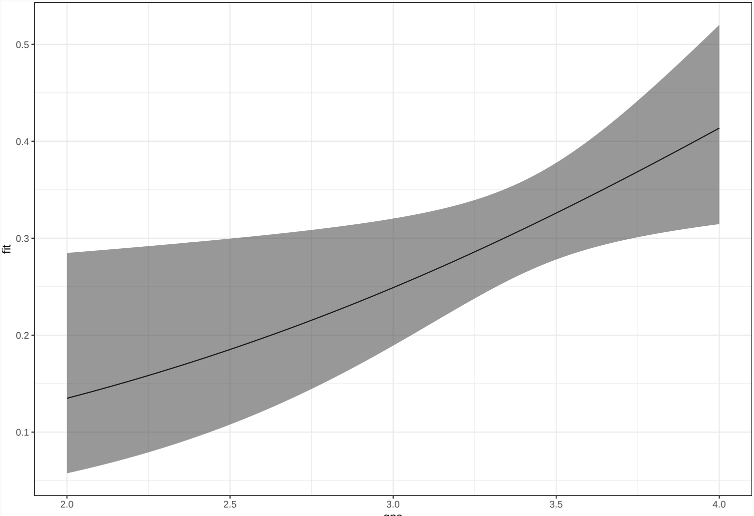
Can we fit a model to predict admission?

```r
m1<-lm(admit ~ gre + gpa,
       data = admissions)
```



Let's try a different approach

```
m2<-glm(admit ~ gre + gpa,
        data = admissions,
       family = "binomial")
```



A generalized linear model

Our linear probability model was:

$$Pr(admit = 1) = \beta_0 + \beta_1 GRE + \beta_2 GPA + \beta_3 Rank + \varepsilon$$

Our logistic regression model takes the form:

$$logit(Pr(admit = 1)) = \beta_0 + \beta_1 GRE + \beta_2 GPA + \beta_3 Rank$$

The logit function is our link between the linear predictor term $X\beta$ and the outcome *admit*.

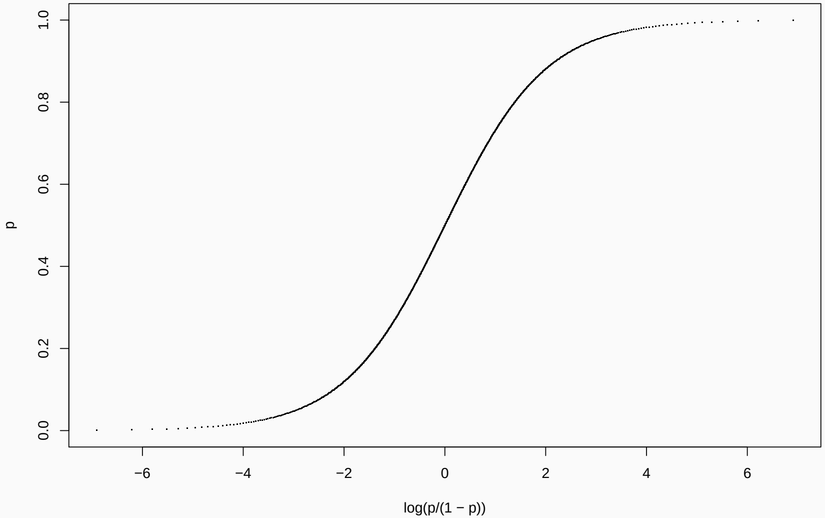
The logit function

The logit function transforms a probability value on $[0, 1]$ to a continuous distribution

$$logit(p) = log\frac{p}{1-p}$$

The logit function

```r
p<-seq(0,1,0.001)
plot(log(p/(1-p)), pch = ".", p)
```



Logistic regression is a GLM with a logit link

A generalized linear model with link function g takes the form:

$$g(y) = X\beta$$

For OLS, the link function is the identity function $g(y) = y$

For logistic regression, the link function is the logit function

$$logit(y) = X\beta$$

$$y = logit^{-1}(X\beta)$$

Defining logit and its inverse

$$logit(p) = log\frac{p}{1-p}$$

$$logit^{-1}(x) = \frac{exp(x)}{exp(x)+1}$$

We can use these functions to transform values back and forth from our logit-linear scale and the probability scale.

Defining logit and its inverse

$$logit(p) = log\frac{p}{1-p}$$

$$logit^{-1}(x) = \frac{exp(x)}{exp(x)+1}$$

We can use these functions to transform values back and forth from our logit-linear scale and the probability scale.

Challenge: create two functions in R. logit() and inv_logit() that will compute the logit for any p and the inverse logit for any x.

Defining logit and its inverse

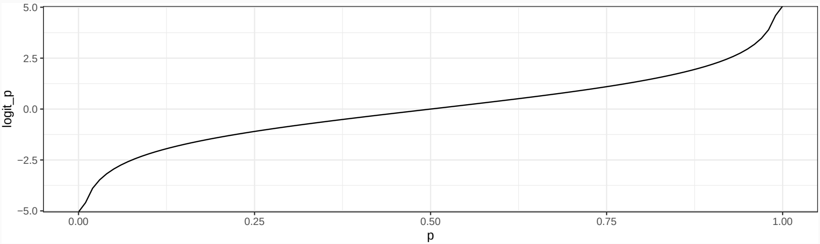
```
logit<-function(p){
  log(p/(1-p))
  }

inv_logit<-function(x){
  exp(x) / (exp(x) + 1)
}
```

Challenge: practice with these functions

1. Create a tibble with values of p ranging from 0 to 1
2. Use mutate to add a variable called logit_p that is equal to $\mathrm{logit}(p)$
3. Use ggplot to plot p and logit p

Solution

```r
#1.
new_dat<-tibble(p = seq(0, 1, by = 0.01))
#2.
new_dat<-new_dat %>%
  mutate(logit_p = logit(p))
#3.
ggplot(new_dat,
       aes(x = p, y = logit_p)) +
  geom_line()
```

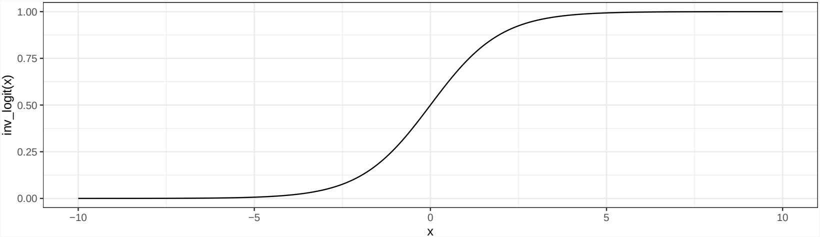


Challenge: practice with these functions

1. Create a tibble with values of x ranging from -10 to 10
2. Use mutate to add a variable called inv_logit_x that is equal to $\mathrm{logit}^{-1}(x)$
3. Use ggplot to plot x and inv_logit_x

Solution

```r
#1.
new_dat<-tibble(x = seq(-10, 10, by = 0.1))
#2.
new_dat<-new_dat %>%
  mutate(inv_logit_x = inv_logit(x))
#3.
ggplot(new_dat,
       aes(x = x, y = inv_logit(x))) +
  geom_line()
```



Motivation for logistic regression

1. Select a model that works for binary outcomes
2. Preserve the linear structure for predictors
3. Map unbounded $(-\infty, \infty)$ linear predictors onto probability $(0, 1)$
4. Map expected probability into binary outcomes

Running logistic models in R: the glm() function

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

m1_b<-stan_glm(admit ~ gpa,
               data = admissions,
               family = "binomial")
```

What do these models mean?

```r
coef(m1_b)
```

```
## (Intercept)         gpa
##   -4.365064    1.053550
```

This is the direct interpretation in terms of `admit`

$$Pr(admit_i = 1 | gpa) = \mathrm{logit}^{-1}(\beta_0 + \beta_1 gpa_i)$$

What do these models mean?

```
coef(m1_b)
```

```
## (Intercept)         gpa
##   -4.365064    1.053550
```

This is the direct interpretation in terms of `admit`

$$Pr(admit_i = 1 | gpa) = \mathrm{logit}^{-1}(\beta_0 + \beta_1 gpa_i)$$

What does β_1 mean?

```r
coef(m1_b)
```

```
## (Intercept)         gpa
##   -4.365064    1.053550
```

$$Pr(admit_i = 1|gpa) = \mathrm{logit}^{-1}(\beta_0 + \beta_1 gpa_i)$$

```r
coef(m1_b)
```

```
## (Intercept)         gpa
##   -4.365064    1.053550
```

$$Pr(admit_i = 1|gpa) = \mathrm{logit}^{-1}(\beta_0 + \beta_1 gpa_i)$$

What does β_1 mean?

Common interpretations

Because $\mathrm{logit}(p_i) = \log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_i$

Common interpretations

Because $\mathrm{logit}(p_i) = \log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_i$

We can interpret β_1 several ways

- Log odds: β_1

Common interpretations

Because $\mathrm{logit}(p_i) = \log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_i$

We can interpret β_1 several ways

- Log odds: β_1
- Odds ratio: e^{β_1}

Common interpretations

Because $\mathrm{logit}(p_i) = \log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_i$

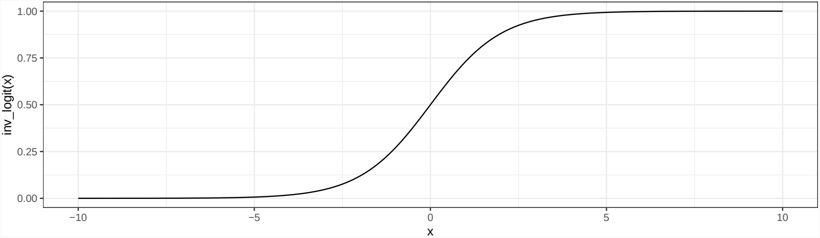
We can interpret β_1 several ways

- Log odds: β_1
- Odds ratio: e^{β_1}
- Probability: $logit^{-1}(x) = \frac{exp(X\beta)}{exp(X\beta)+1}$

The challenge of interpretation

The expected change in probability for p for a 1 unit change in x (or slope) is not constant!

```
ggplot(new_dat,
       aes(x = x, y = inv_logit(x))) +
  geom_line()
```



1. Compute the expected probability of admission for a student with a 2.0 GPA

The challenge of interpretation

1. Compute the expected probability of admission for a student with a 2.0 GPA
2. For a student with a 3.0 GPA

The challenge of interpretation

1. Compute the expected probability of admission for a student with a 2.0 GPA
2. For a student with a 3.0 GPA
3. Assume the 2.0 and 3.0 student each bumped their GPA up by 0.5. How much does their expected probability of admission change?

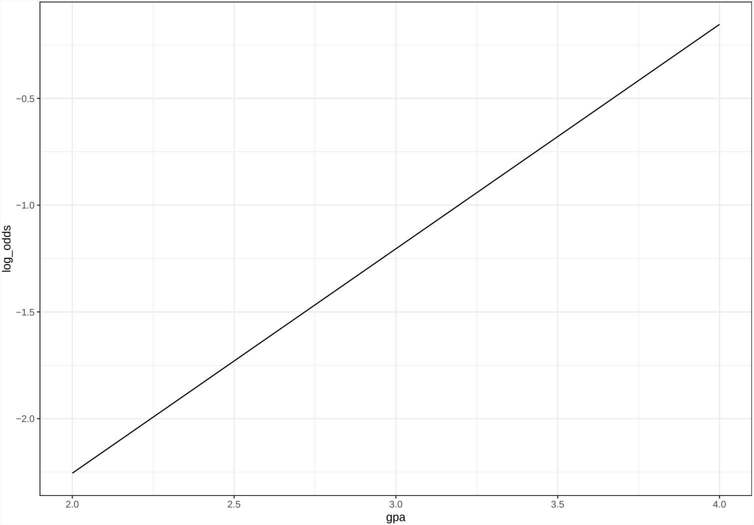
The challenge of interpretation

```
new_dat<-data.frame(gpa = seq(2, 4, by= 0.25))
new_dat<-new_dat %>%
  mutate(log_odds = predict(m1, new_dat),
         probability = predict(m1, new_dat, type = "response"))
new_dat
```

```
##    gpa   log_odds probability
## 1 2.00 -2.2553699  0.09488728
## 2 2.25 -1.9925927  0.11998284
## 3 2.50 -1.7298155  0.15061118
## 4 2.75 -1.4670383  0.18739319
## 5 3.00 -1.2042611  0.23071805
## 6 3.25 -0.9414839  0.28060069
## 7 3.50 -0.6787068  0.33655000
## 8 3.75 -0.4159296  0.39749117
## 9 4.00 -0.1531524  0.46178656
```

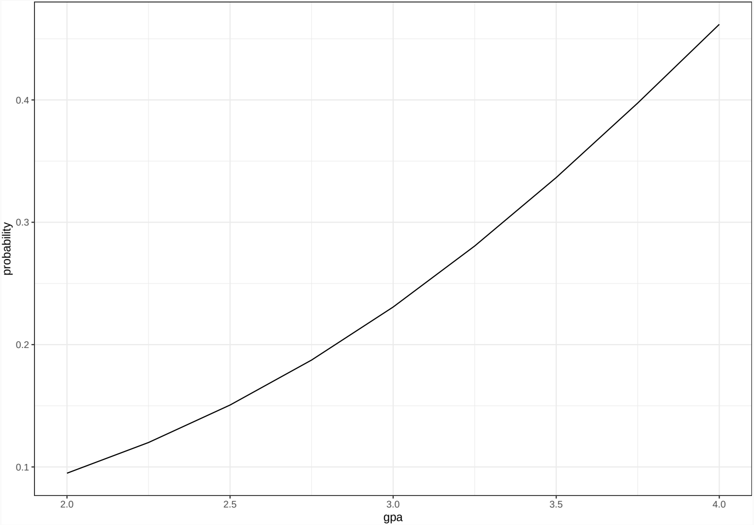
The challenge of interpretation: log odds scale

```
ggplot(new_dat, aes(x = gpa, y = log_odds)) +
  geom_line()
```



The challenge of interpretation: probability scale

```
ggplot(new_dat, aes(x = gpa, y = probability)) +
  geom_line()
```



Break

Lab: Let's fit a more complex model

What else might predict admission?

```
head(admissions)
```

```
##   admit gre  gpa rank
## 1     0 380 3.61    3
## 2     1 660 3.67    3
## 3     1 800 4.00    1
## 4     1 640 3.19    4
## 5     0 520 2.93    4
## 6     1 760 3.00    2
```

1. Write out a model
2. Fit the model
3. Think about revising the model
4. Compare model fits
5. Interpret the model

Homework 5

Who was most (and least) likely to die on the Titanic? Use `~/hw/data/titanic.csv` for this one.

1. Write out a model
2. Fit the model
3. Think about revising the model
4. Compare model fits
5. Interpret the model