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)
```

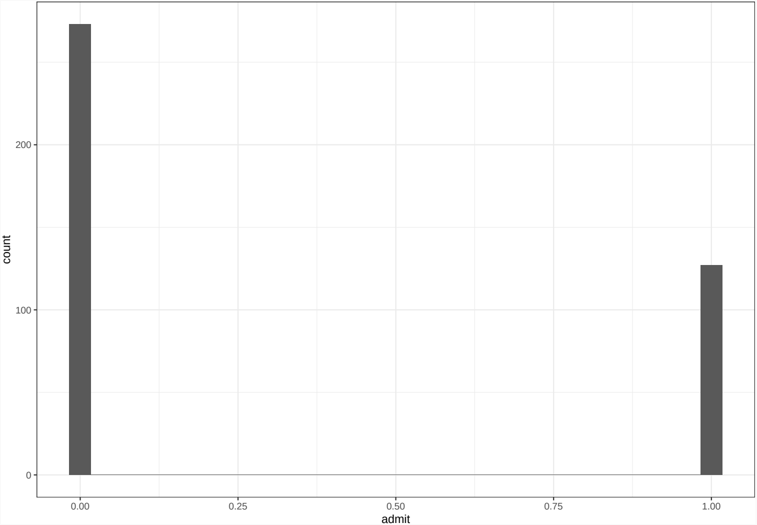
```
##   admit gre  gpa rank
## 1     0 380 3.61    3
## 2     1 660 3.67    3
## 3     1 800 4.00    1
## 4     1 640 3.19    4
## 5     0 520 2.93    4
## 6     1 760 3.00    2
```

```
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)
```

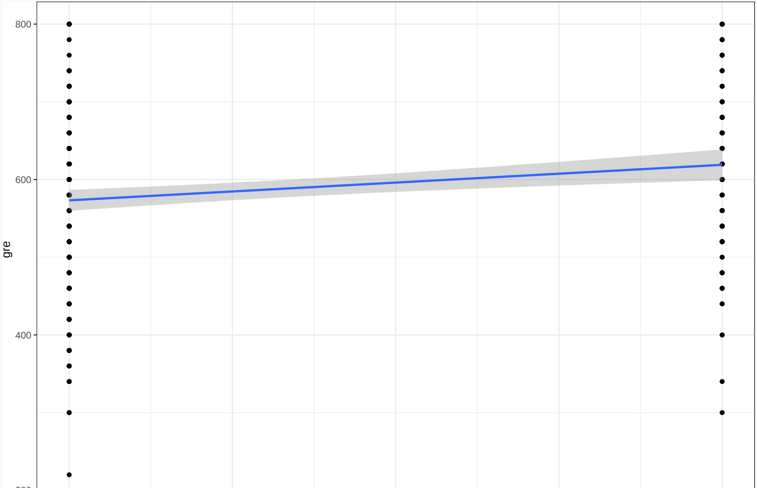
```
## [1] 0.2172368
```

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

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

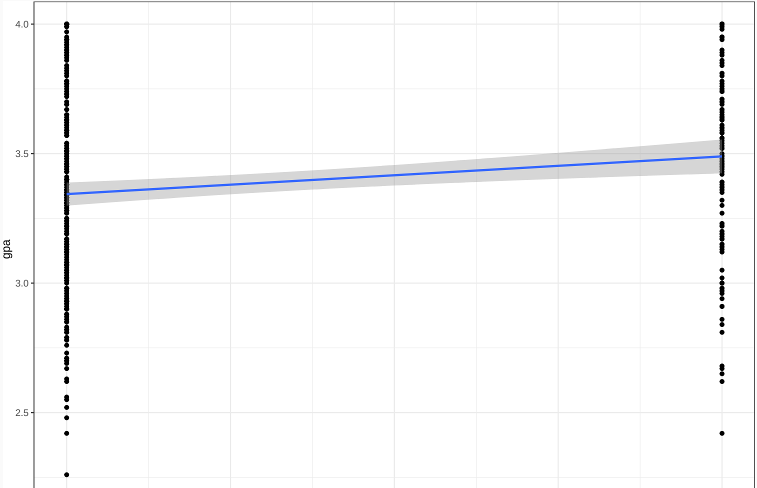
How does GRE relate to admission?

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



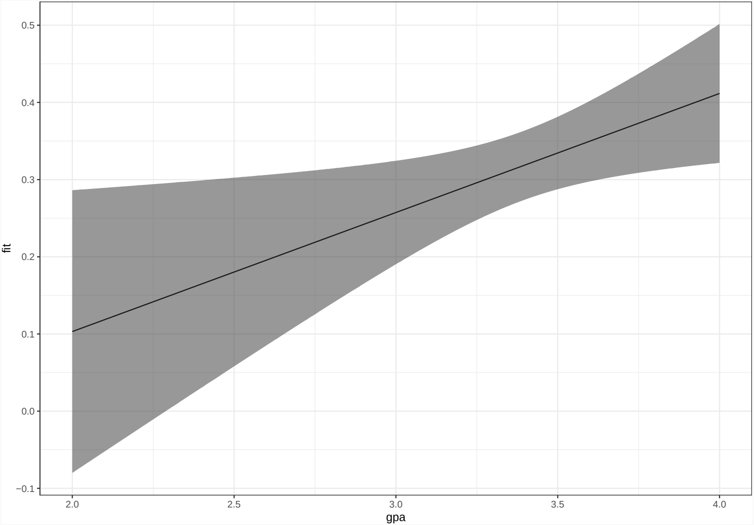
GPA?

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



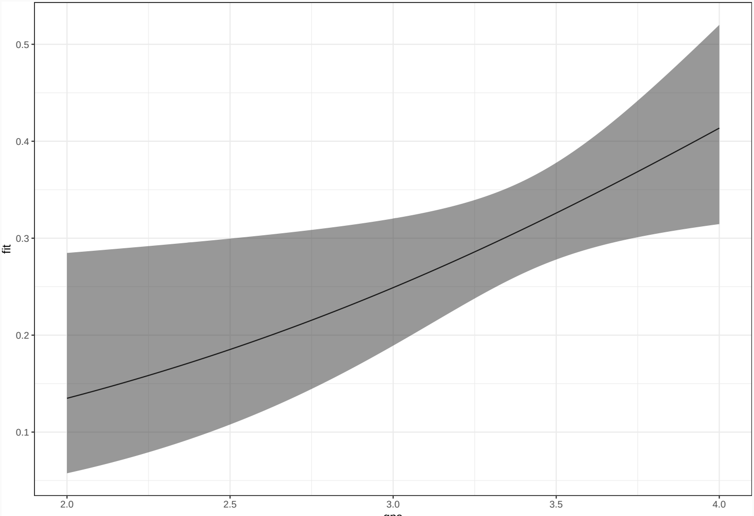
Can we fit a model to predict admission?

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



Let’s try a different approach

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



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*.

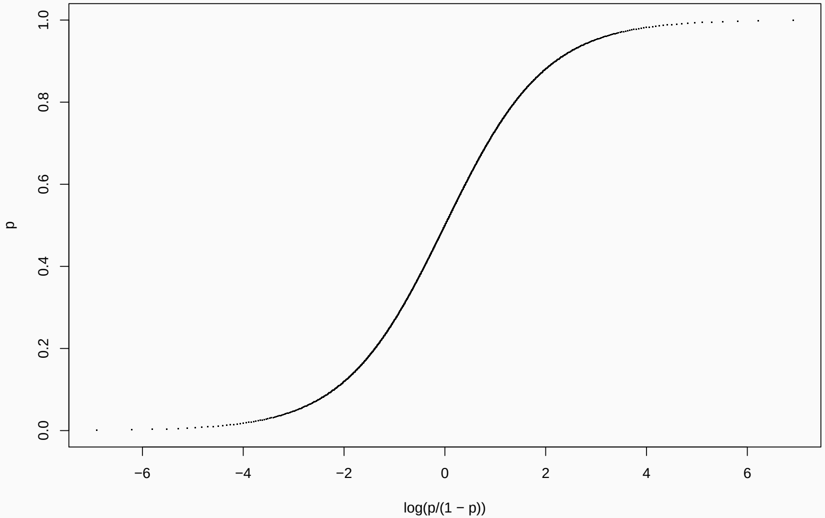
The logit function

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

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

The logit function

```
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.

Logistic regression

Uses the logit function to model the probability of a binary outcome being equal to 1. The logit function transforms the bounded interval $[0, 1]$ to a continuous distribution, allowing us to proceed with building a regression model as we ordinarily would.

Logistic regression may have more accurate uncertainty estimates than a linear probability model for binary outcomes. Logistic regression also constrains model predictions to $[0, 1]$.

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")
```

How do we interpret the coefficients?

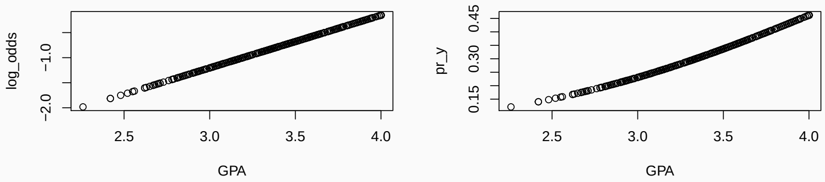
- Log odds: β_1
- Odds ratio: e^{β_1}
- Probability: $logit^{-1}(x) = \frac{exp(X\beta)}{exp(X\beta)+1}$

I tend to prefer transforming to a probability scale, as log odds and odds ratios are a bit confusing to define and are not especially intuitive.

To get predicted probabilities from m1

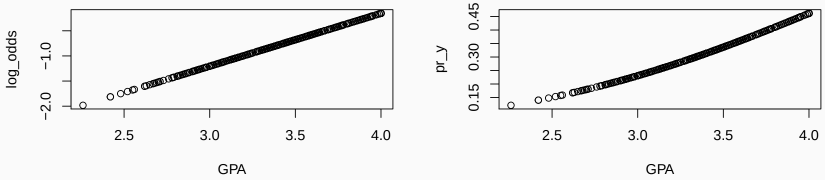
We need $X\beta$, then apply the logit inverse function

```
x<-cbind(rep(1, nrow(admissions)), admissions$gpa)
log_odds<-coef(m1)%*%t(x)
pr_y<-exp(log_odds)/(exp(log_odds) +1)
par(mfrow=c(1,2))
plot(x[,2], log_odds, xlab = "GPA")
plot(x[,2], pr_y, xlab = "GPA")
```



Alternatively

```
log_odds<-predict(m1)
pr_y<-predict(m1, type = "response")
```



Next week