The basic logistic regression model

Interpreting the model results

Evaluating the model

okcupid

Have an account? Sign in

DATING DESERVES BETTER

On OkCupid, you're more than just a photo. You have stories to tell, and passions to share, and things to talk about that are more interesting than the weather. Get noticed for who you are, not what you look like. Because you deserve what dating deserves: better.

JOIN okc

By clicking Join, you agree to our Terms of Service. Learn about how we process and use your data in our Privacy Policy and how we use cookies and similar technology in our Cookie Policy.

GET THE APP

The OkCupid data set

- The OkCupid data set contains information about 59946 profiles from users of the OkCupid online dating service.
- We have data on user age, height, sex, income, sexual orientation, education level, body type, ethnicity, and more.
- OkCupid often publishes their own analyses of their data—see https://theblog.okcupid.com/tagged/data.
- Let's see if we can predict the sex/gender of the user based on their height.

What's wrong with this regression?

$$\widehat{\text{sex}} = \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{height}$$

What's wrong with this regression?

$$\widehat{\text{sex}} = \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{height}$$

The *Y* variable here is categorical (two levels—everyone in this data set is either labeled male or female), so regular linear regression might not be the best choice here.

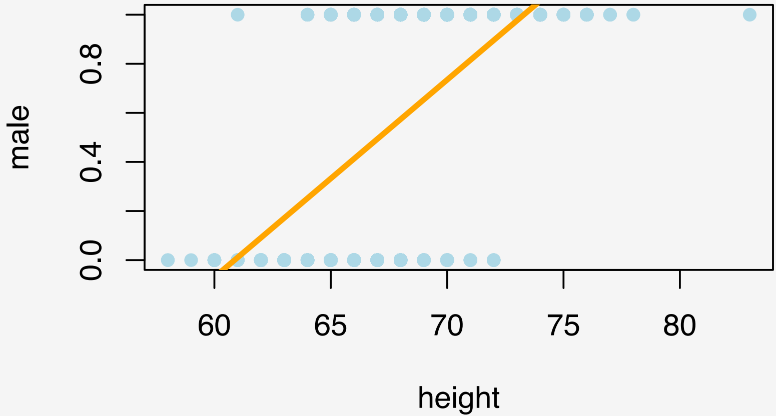
But what if we just do it anyway?

Let's first create a dummy variable to convert sex to a quantitative dummy variable:

```
profiles = profiles %>%
  mutate(male=ifelse(profiles$sex == "m", 1, 0))
```

We could do this with 1 representing either male or female (it wouldn't matter).

But what if we just do it anyway?



A line is a spectacularly bad fit to this data. And what does it mean to predict that male = 0.7 (or 1.2)?

The idea behind logistic regression

- Instead of predicting whether someone is male, let's predict the *probability* that they are male
- In logistic regression, one level of Y is always called "success" and the other called "failure." Since $Y = 1$ for males, in our setup we have designated males as "success." (You could also set $Y = 1$ for females and call females "success.")
- Let's fit a curve that is always between 0 and 1.

Odds

- When something has “even (1/1) odds,” the probability of success is 1/2

Odds

- When something has "even (1/1) odds," the probability of success is 1/2
- When something has "2/1 odds," the probability of success is 2/3

Odds

- When something has "even (1/1) odds," the probability of success is 1/2
- When something has "2/1 odds," the probability of success is 2/3
- When something has "3/2 odds," the probability of success is 3/5

Odds

- When something has "even (1/1) odds," the probability of success is $1/2$
- When something has "2/1 odds," the probability of success is $2/3$
- When something has "3/2 odds," the probability of success is $3/5$
- In general: The odds of something that happens with probability p are $p/(1-p)$

Odds

During March Madness on FanDuel a \$100 bet on San Diego State winning March Madness paid out \$360 – that is, they gave SDSU odds to *win* of $1/3.6 = 5/18$

What was the implied *probability* that SDSU wins March Madness?

Odds

During March Madness on FanDuel a \$100 bet on San Diego State winning March Madness paid out \$360 – that is, they gave SDSU odds to *win* of $1/3.6 = 5/18$

What was the implied *probability* that SDSU wins March Madness?

$$\frac{p}{1-p} = \frac{1}{3.6} \Rightarrow p = \frac{1}{4.6} \approx 0.2175$$

The logistic regression model

Logistic regression models the log odds of success p as a linear function of X:

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

This fits an S-shaped curve to the data (we'll see what it looks like later).

Let's try it

```
Call:
glm(formula = male ~ height, family = binomial, data = my.profiles)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6109  -0.4837   0.2032   0.5318   4.0110

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -44.448609   0.357510  -124.3   <2e-16 ***
height        0.661904   0.005293   125.1   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 80654  on 59825  degrees of freedom
Residual deviance: 44637  on 59824  degrees of freedom
AIC: 44641

Number of Fisher Scoring iterations: 6
```

How to interpret the curve?

The regression output tells us that our prediction is

$$\log(\text{odds}) = \log\left(\frac{P(\text{male})}{1 - P(\text{male})}\right) = -44.45 + 0.66 \cdot \text{height}.$$

How to interpret the curve?

The regression output tells us that our prediction is

$$\log(\text{odds}) = \log\left(\frac{P(\text{male})}{1 - P(\text{male})}\right) = -44.45 + 0.66 \cdot \text{height}.$$

Let's solve for $P(\text{male})$:

$$\widehat{P(\text{male})} = \frac{e^{-44.45+0.66\cdot\text{height}}}{1 + e^{-44.45+0.66\cdot\text{height}}}$$

Making predictions

We can use `predict` to automate the process of plugging into the equation:

```
predict(model, data.frame(height=69), type="response")
```

```
   1
0.77
```

$$\frac{e^{-44.45+0.66\cdot 69}}{1+e^{-44.45+0.66\cdot 69}} = 0.77$$

Making predictions

We can use `predict` to automate the process of plugging into the equation:

```
predict(model, data.frame(height=69), type="response")
```

```
   1
0.77
```

$$\frac{e^{-44.45+0.66\cdot 69}}{1+e^{-44.45+0.66\cdot 69}} = 0.77$$

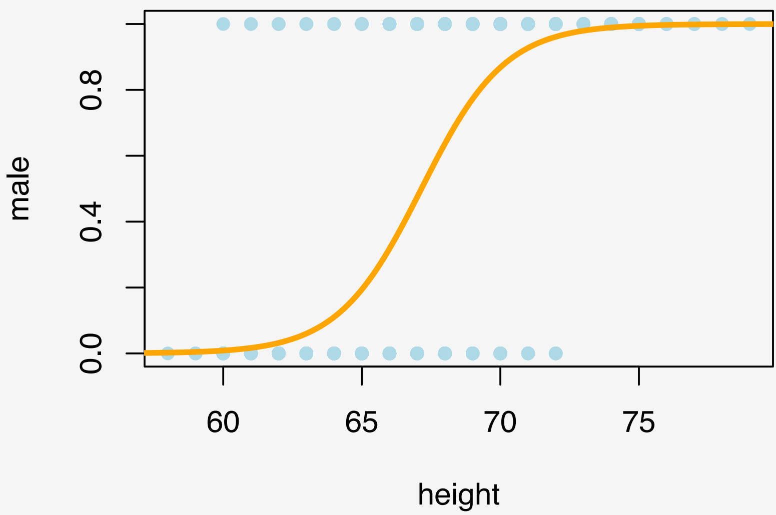
We predict that someone that is 5’9” has a 77% chance of being male.

The basic logistic regression model

Interpreting the model results

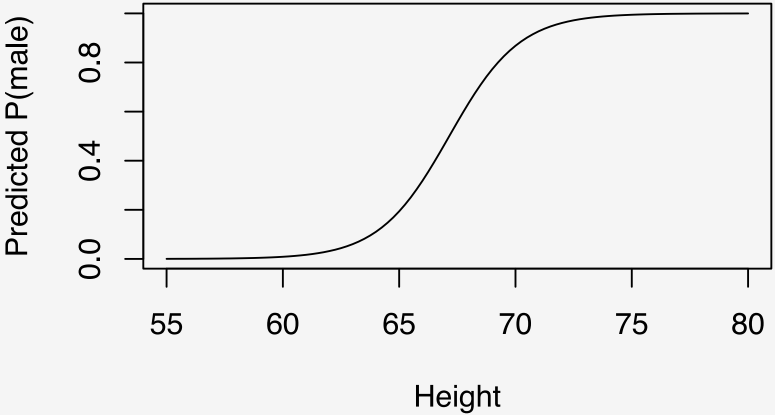
Evaluating the model

Visualizing the model



How to interpret the curve?

$$\widehat{P(\text{male})} = \frac{e^{-44.45+0.66\cdot\text{height}}}{1 + e^{-44.45+0.66\cdot\text{height}}}$$



Interpreting the coefficients

Our prediction equation is:

$$\log\left(\frac{P(\text{male})}{1 - P(\text{male})}\right) = -44.45 + 0.66 \cdot \text{height}.$$

Let's start with some basic, but not particularly useful, interpretations:

- When height $= 0$, we predict that the log odds will be -44.45

Interpreting the coefficients

Our prediction equation is:

$$\log\left(\frac{P(\text{male})}{1 - P(\text{male})}\right) = -44.45 + 0.66 \cdot \text{height}.$$

Let's start with some basic, but not particularly useful, interpretations:

- When height $= 0$, we predict that the log odds will be -44.45

Interpreting the coefficients

Our prediction equation is:

$$\log\left(\frac{P(\text{male})}{1 - P(\text{male})}\right) = -44.45 + 0.66 \cdot \text{height}.$$

Let's start with some basic, but not particularly useful, interpretations:

- When height $= 0$, we predict that the log odds will be -44.45 , so the probability of male is predicted to be very close to 0%.
- When height increases by 1 inch, we predict that the log odds of being male will increase by 0.66.

Interpreting the coefficients

Let's rewrite the prediction equation as:

$$\text{Predicted odds of male} = e^{-44.45+0.66\cdot\text{height}}$$

Increasing height by 1 inch will *multiply* the odds by $e^{0.66} = 1.94$; i.e., increase the odds by 94%.

Interpreting the coefficients

Let's rewrite the prediction equation as:

$$\text{Predicted odds of male} = e^{-44.45+0.66\cdot\text{height}}$$

Increasing height by 1 inch will *multiply* the odds by $e^{0.66} = 1.94$; i.e., increase the odds by 94%.

Increasing height by 2 inches will *multiply* the odds by $e^{2\cdot 0.66} = 3.76$; i.e., increase the odds by 276%.

The basic logistic regression model

Interpreting the model results

Evaluating the model

How good is our model?

- Unfortunately, the typical R^2 metric isn't available for logistic regression.

How good is our model?

- Unfortunately, the typical R^2 metric isn't available for logistic regression.
- However, there are many "pseudo-R^2" metrics that indicate model fit.

How good is our model?

- Unfortunately, the typical R^2 metric isn't available for logistic regression.
- However, there are many "pseudo-R^2" metrics that indicate model fit.
- But: most of these pseudo-R^2 metrics are difficult to interpret, so we'll focus on something simpler to interpret and communicate.

How many cases did we accurately predict?

We could use our model to make a prediction of sex based on the probability. Suppose we say that our prediction is:

$$\text{Prediction} = \begin{cases} \text{male}, & \text{if } \widehat{P(\text{male})} \geq 0.5, \\ \text{female}, & \text{if } \widehat{P(\text{male})} < 0.5. \end{cases}$$

How many cases did we accurately predict?

We could use our model to make a prediction of sex based on the probability. Suppose we say that our prediction is:

$$\text{Prediction} = \begin{cases} \text{male}, & \text{if } \widehat{P(\text{male})} \geq 0.5, \\ \text{female}, & \text{if } \widehat{P(\text{male})} < 0.5. \end{cases}$$

Now we can compute the fraction of people whose sex we correctly predicted:

```
predicted.sex <- ifelse(predict(model, type="response") >= 0.5,
                        "m", "f")
correct <- ifelse(predicted.sex == my.profiles$sex, 1, 0)
sum(correct) / nrow(my.profiles)
```

```
[1] 0.83
```

How many cases did we accurately predict?

83% sounds pretty good—what should we compare it against?

How many cases did we accurately predict?

83% sounds pretty good—what should we compare it against?

We should compare 83% against what we would have gotten if we just predicted the most common outcome (male) for everyone, without using any other information:

How many cases did we accurately predict?

83% sounds pretty good—what should we compare it against?

We should compare 83% against what we would have gotten if we just predicted the most common outcome (male) for everyone, without using any other information:

```
xtabs(~ sex, data=my.profiles) %>% prop.table()
```

```
sex
  f   m
0.4 0.6
```

How many cases did we accurately predict?

83% sounds pretty good—what should we compare it against?

We should compare 83% against what we would have gotten if we just predicted the most common outcome (male) for everyone, without using any other information:

```
xtabs(~ sex, data=my.profiles) %>% prop.table()

sex
  f   m
0.4 0.6
```

In other words, our model provided a “lift” in accuracy from 60% to 83%.

The confusion matrix

Sometimes it is useful to understand what kinds of errors our model is making.

First, we have to pick one category to be "positive" and the other to be "negative." Since we used 1 for male, positive (P) = male and negative (N) = female.

The confusion matrix

Every case we try to predict/classify falls into one of these buckets:

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

- True positives (TP): predicting male for someone that is male
- True negatives (TN): predicting female for someone that is female
- False positives (FP): predicting male for someone that is female
- False negatives (FN): predicting female for someone that is male

The confusion matrix

```
xtabs(~ predicted.sex + my.profiles$sex)

             my.profiles$sex
predicted.sex     f     m
            f 19466  5494
            m  4623 30243
```

5494 false negatives (true M predicted as F) and 4623 false positives (true F predicted as M).

The confusion matrix

We can also normalize the confusion matrix to estimate error rates:

- True positive rate TP/(TP + FN): what proportion of things that are really positive do we predict to be positive?
- True negative rate TN/(TN + FP): what proportion of things that are really negative do we predict to be negative?
- False positive rate FP/(TN + FP): what proportion of things that are really negative do we (wrongly) predict to be positive?
- False negative rate FN/(TP + FN): what proportion of things that are really positive do we (wrongly) predict to be negative?

The confusion matrix

```
xtabs(~ predicted.sex + my.profiles$sex) %>%
  prop.table(2)
```

```
             my.profiles$sex
predicted.sex    f    m
            f 0.81 0.15
            m 0.19 0.85
```

Important: Make sure that you use predicted + actual on the right-hand side of the ~ here.

Model evaluation: Final thoughts

- You can trade off false positives and negatives by using different classification rules (predicting 1 when the predicted probability is $> p$ for some $p \neq 0.5$).
- False positives and negatives can have very different costs – e.g., using predicted default probabilities when deciding whether to write a loan (”positive”) or deny an application (”negative”)
- Here we've computed in-sample measures of accuracy. But just like R^2 or residual SE in linear regression, this measure is optimisitic! Use train/test splits or cross-validation for good estimates of accuracy on new data