Logistic Regression

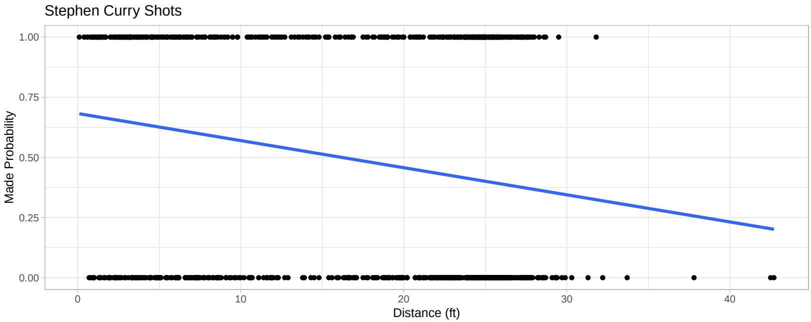
Part 1 - model basics and coefficient interpretations

Ryan Miller

Grinnell College
Statistics

Introduction

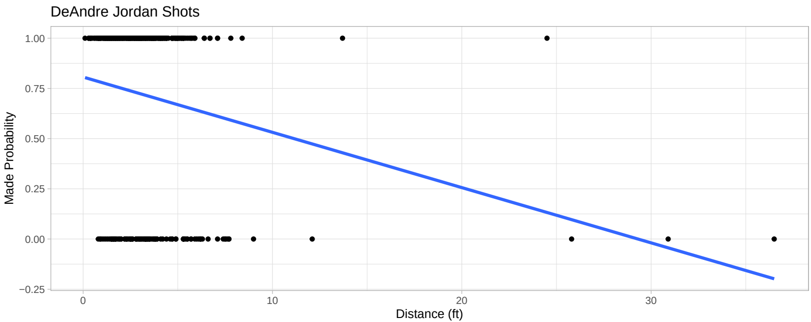
Below is a graph of each shot attempted by Stephen Curry in the 2014-15 NBA season where the outcome was recorded as binary:



The blue line is a simple linear regression model fit to these data, are there any problems with this model?

Introduction (cont.)

Below is the same graph for DeAndre Jordan:



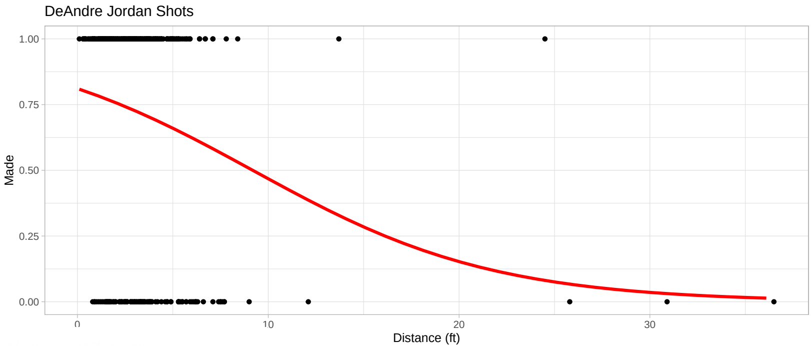
Clearly this is an inappropriate modeling approach if it suggests negative probabilities for a large range of values that were observed in the data.

Logistic Curve

Logistic regression takes the form:

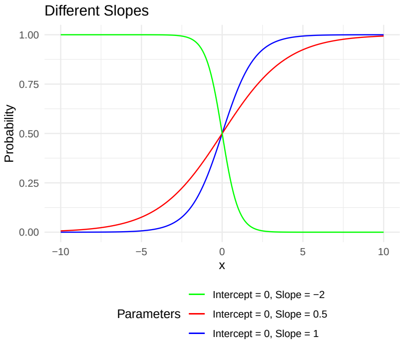
$$log\left(\frac{Pr(y=1)}{1-Pr(y=1)}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

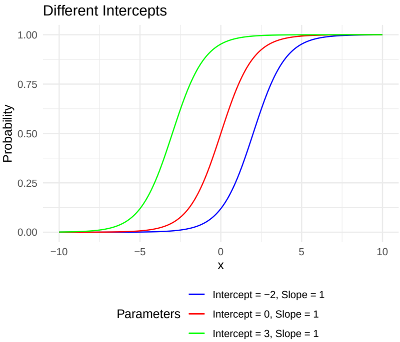
This produces a *logistic curve* for $Pr(y=1)$ as a function of X_j:



Logistic Curves

The shape of a logistic curve depends upon its parameters. We won't cover the details, but we'll rely upon R to estimate the parameters of the best-fitting logistic curve for our sample data.





Grinnell College
Statistics

Model Coefficients

In logistic regression each variable makes a linear contribution to the *log-odds* of the outcome coded as "1"

$$log\big(\frac{Pr(y=1)}{1-Pr(y=1)}\big) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

The fitted logistic regression model for Stephen Curry is:

$$log\big(\frac{\hat{y}}{1-\hat{y}}\big) = 0.75 - 0.05 \cdot \text{Distance}$$

So, for every 1-ft increase in distance we'd expect the log-odds of Steph making the shot to decrease by 0.05, which isn't a very digestible interpretation.

Model Coefficients (cont.)

Fortunately, we can use arithmetic to make sense of things:

$$
\begin{aligned}
log\left(\frac{Pr(y=1)}{1-Pr(y=1)}\right) &= \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p \\
\implies \frac{Pr(y=1)}{1-Pr(y=1)} &= exp(\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p) \\
\implies \frac{Pr(y=1)}{1-Pr(y=1)} &= exp(\beta_0) \cdot exp(\beta_1 X_1) \cdot \ldots \cdot exp(\beta_p X_p)
\end{aligned}
$$

- The exponent of the intercept represents the *baseline odds*
- The exponent of $\beta_1, \ldots, \beta_p$ is a *multiplier* of the baseline odds

Interpreting the Intercept

Our fitted logistic regression model for Stephen Curry is:

$$log\left(\frac{\hat{y}}{1-\hat{y}}\right) = 0.75 - 0.05 \cdot \text{Distance}$$

Because $exp(0.75) = 2.12$, the odds Steph makes a shot from a distance of zero feet (right underneath the basket) are 2.12

- Steph is expected to make 2.12 shots from this distance for every 1 he misses
- This is an implied probability of 68%

Grinnell College
Statistics

Interpreting the Slope

Our fitted logistic regression model for Stephen Curry is:

$$log\left(\frac{\hat{y}}{1-\hat{y}}\right) = 0.75 - 0.05 \cdot \text{Distance}$$

Because $exp(-0.05) = 0.951$, each additional 1-ft in distance changes the odds of Steph making a shot by a multiplicative factor of 0.951, or a 4.9% decrease

Grinnell College
Statistics

Interpreting Binary Explanatory Variables

When an explanatory variable in logistic regression is binary, $exp(\beta_j)$ gives us an estimated *odds ratio*. Consider the model:

$$log\big(\tfrac{Pr(y=1)}{1-Pr(y=1)}\big) = \beta_0 + \beta_1 \cdot (\text{Location}{=}\text{Home})$$

- When a shot is taken during an away game the expected odds it being made are $\text{Odds}_A = exp(\beta_0)$
  - For a home game: $\text{Odds}_H = exp(\beta_0 + \beta_1)$
- Dividing the expected odds to get an odds ratio:

$$\frac{\text{Odds}_H}{\text{Odds}_A} = \frac{exp(\beta_0 + \beta_1)}{exp(\beta_0)} = \frac{exp(\beta_0) \cdot exp(\beta_1)}{exp(\beta_0)} = exp(\beta_1)$$

Conclusion

- Logistic regression gives us a sensible statistical model for expressing a binary outcome as a function of several explanatory variables
  - The log-odds of the outcome encoded as "1" are modeled by a linear combination of the explanatory variables
  - This implies an S-shaped logistic curve relating each explanatory variable and $Pr(y = 1)$
- To interpret an estimated coefficient, we must first apply an inverse function (*exp*) to undo the log-odds transformation on the outcome