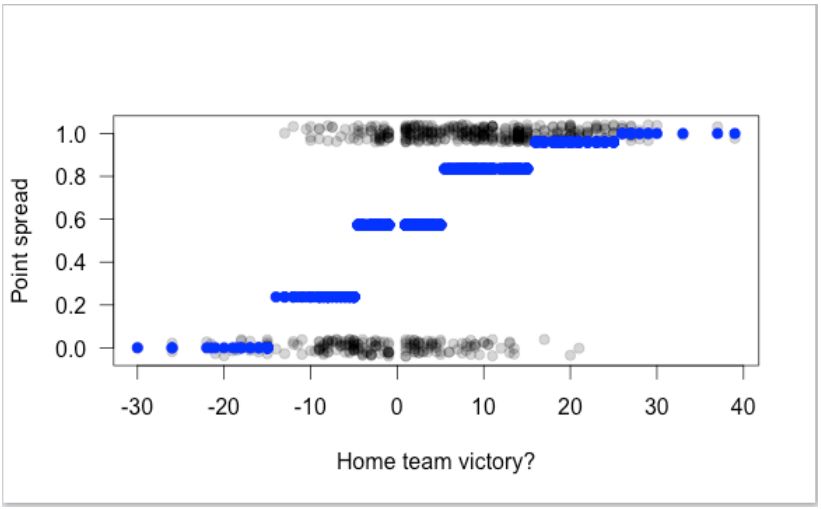
Monday, April 28, 2014

TOPIC: Logistic Regression

Main data set:
Bballbets – data set based on Las Vegas betting on college basketball home team in wins

- Homewin: binomial indicator on whether or not the home team won the game
- Spread: point spread in favor of home team. (The number of points that betting markets predict a team would win or lose by).

If you go through the Bballbets Rscript that is posted online:
KEEP IN MIND THAT GRAPH LABELS ARE FLIPPED
It should be:
X-axis → Point spread
Y-axis → Homewin, where 0 is a loss and 1 is a win.



Conclusions from graph:

Black Dots

- Initially, we just plotted the data (these are the black dots).
- We see that when Vegas tended to bet a negative point spread, there were losses. When the point spread was higher, there were more wins.

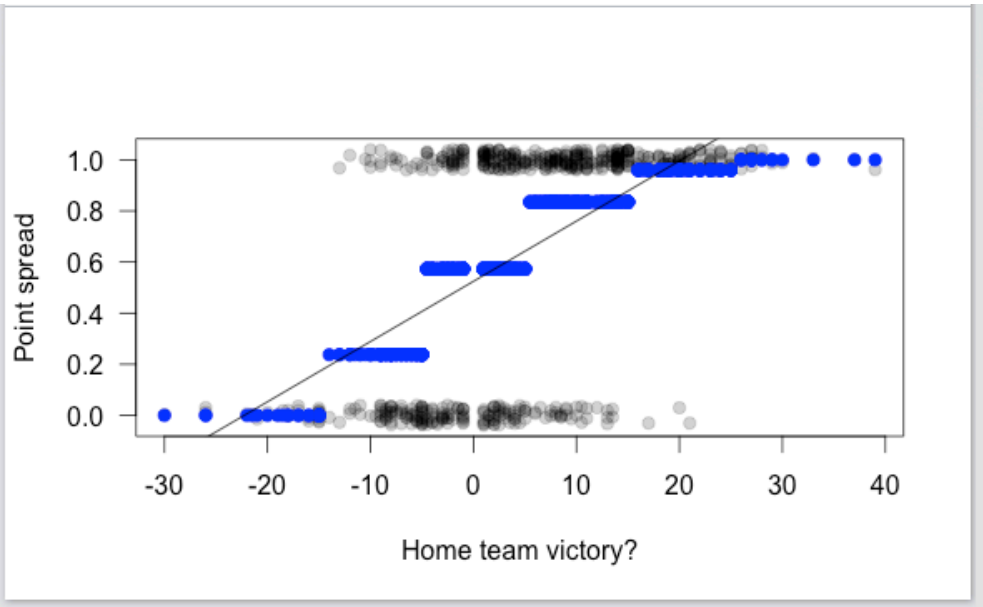
Blue Dots

- Then, we calculated the empirical frequency by splitting the point spreads into "buckets" (these are the blue dots).
- The continuous outcome is chunked into sets of bins that are in intervals of 5.
- Within the buckets, you get the trend you would expect → the number of victories increase as the point spread increases.

Why buckets aren't the most effective method:

- There is a big difference between a point spread of 15 versus 5. Its difficult to compare the two.
- It would be preferable to have the model treat point spread a continuous variable instead of putting point spread into buckets as a discrete variable.

Take home lesson – We can fit binary outcomes using an ordinary least squares regression model. The model will give us the conditional probability of a "1" or a "Yes" of that binary outcome. This model is seen below:



Why using an ordinary least squares regression line isn't the most effective method:

- The line doesn't fit at the extremities
- There technically cannot be a probability below 0 or above 1 (since the only two options are lose (0) or win (1). The regression line does not account for this.

The SOLUTION: Generalized Linear Models

Math behind the concept:

Generalized linear models focus on logistic regression.

We start with the basic equation for linear prediction:

$$\hat{Y}_i = \beta_0 + \beta_1 X_i$$

Where $\hat{Y}_i$:

- Is the Fitted Value, or Conditional Expected Value
- It is the "0"/"1", Win/Loss variable

Where X_i:

- Is the point spread

For a binary outcome:

X_i	$\hat{Y}i$
0: loss	-
1: win	$\beta_0 + \beta_1 X_i$

The conditional expected value of a binary outcome is exactly the probability of getting a 1.
So, from this table we can form the equation for the expected value of y, given x:

$$\Sigma(Y|x) = 1 * (\beta_0 + \beta_1 X_i) + 0 * (-) = \beta_0 + \beta_1 X_i$$

This is known as the Linear Probability Model.
Examples of this include:

- Google using the model to predict whether or not you will click an ad
- Facebook using the model to predict whether or not you'll send someone a friend request

This brings us to the root problem:
Pr $(\hat{Y}_i = 1 \mid X_i)$ should be between 0 and 1, but $(\beta_0 + \beta_1 X_i)$ can be any real number.

For the Logistic Regression Model:

$$P(\hat{Y}_i = 1 \mid X_i) = (e^{\beta 0 + \beta 1 X i}) / (1 + e^{\beta 0 + \beta 1 X i})$$

Since this looks complicated, we can create a function to simplify:

$$g(\beta_0 + \beta_1 X_i)$$

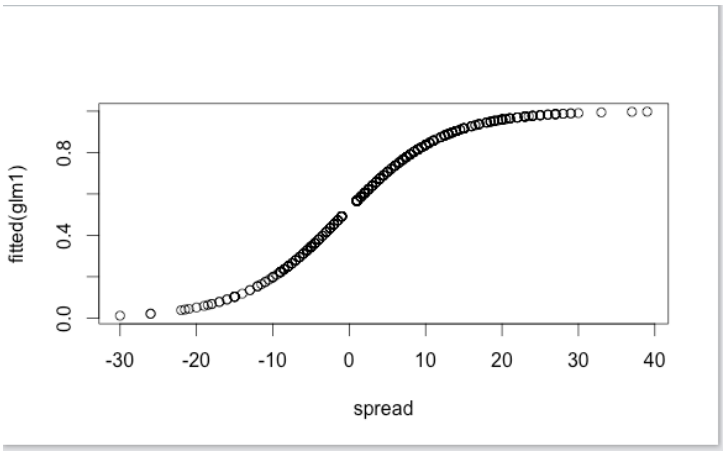
Where:

$$g(s) = g^s / (1 + g^s)$$

First we compute the ordinary $\beta_0 + \beta_1 X_i$ for a regression model, and then we run it through the speed limit function g.

Rstudio:

- The syntax is very similar to LM.
- Instead of LM we use GLM
- Only one extra command, ie:
    - glm = glm(homewin~spread, data = bballbets, family = binomial).
    - We have to say "family = binomial" to tell Rstudio that the outcome we want is binomial. R automatically knows to fit a logistic function.



Conclusions from graph:

- If you plot the fitted values, you get the pretty s shaped curve. It imposes the speed limit so probability isn't below 0 or above 1.
- Within a few percentage points, the linear and logistic regression lines agree, expect for at the tail ends.

NOTE: Logistic regression really useful for forecasting binary outcomes. Linear Regression is usually okay as long as you aren't talking about outcomes that are two extremes (like 0 or 1).

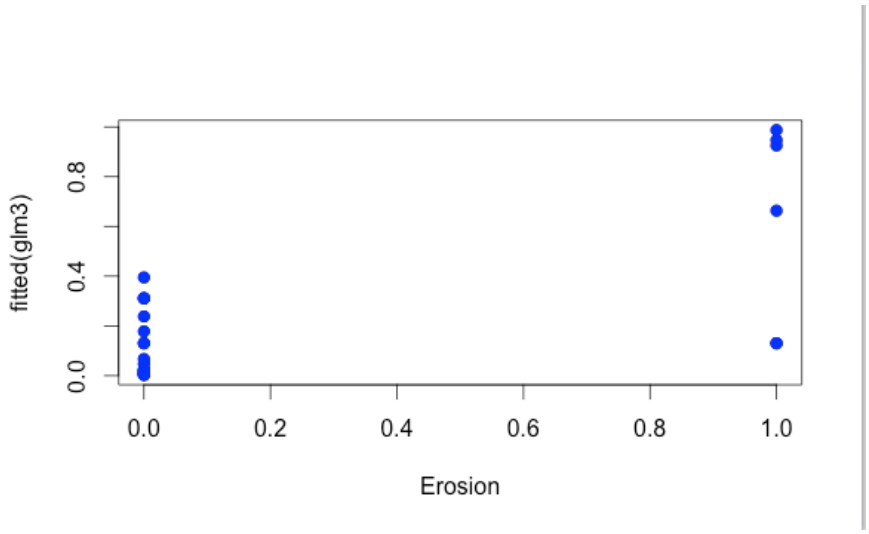
Secondary Example: Orings Data

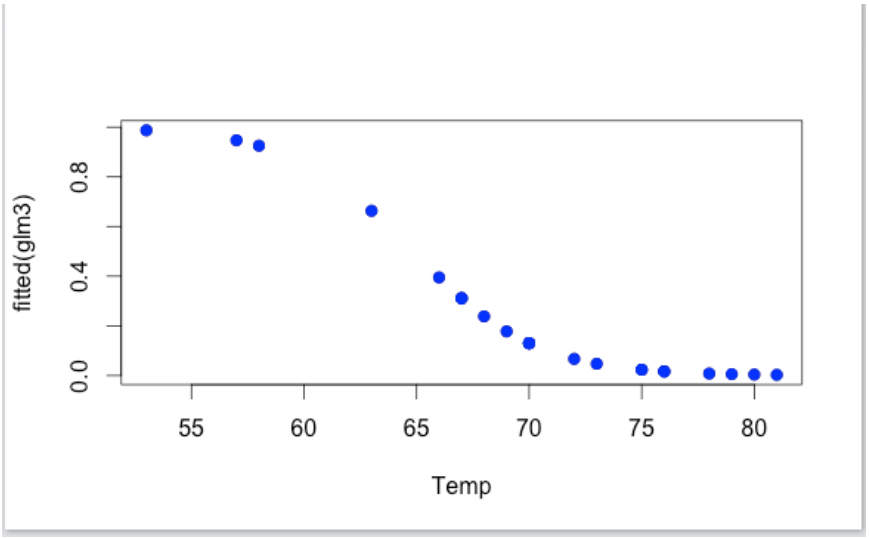
We also worked on a separate example using Orings data.
Background:

- The Challenger space shuttle explosion was caused by the erosion of the O ring.
- The extremely low temperature outside when the shuttle took off caused the O ring to erode.
- The night before, engineers advised NASA not to go through with the launch, but NASA resisted.
- Our goal was to create a simple statistical model that would show how dangerous it was to launch the shuttle during temperatures in the 20s Fahrenheit, as it was the day of the launch.

We tried to forecast probability of oring erosion on the morning of the 29th using the Oring data.
X axis → Temperature
Y axis → 0 = no erosion, 1 = yes erosion

Code
```
glm3 = glm(Erosion~Temp, data=orings, family=binomial)
plot(fitted(glm3)~Temp,data=orings)
points(fitted(glm3)~Temp,data=orings, col='blue', pch=19)
```





Conclusions from graph:

- S curve already basically at 1 when the temp is in 50s, so with temp in 20s the Oring would definitely erode.