Sumanth Reddy
4/28/14

Stat Notes – Logistic Regression

Announcements

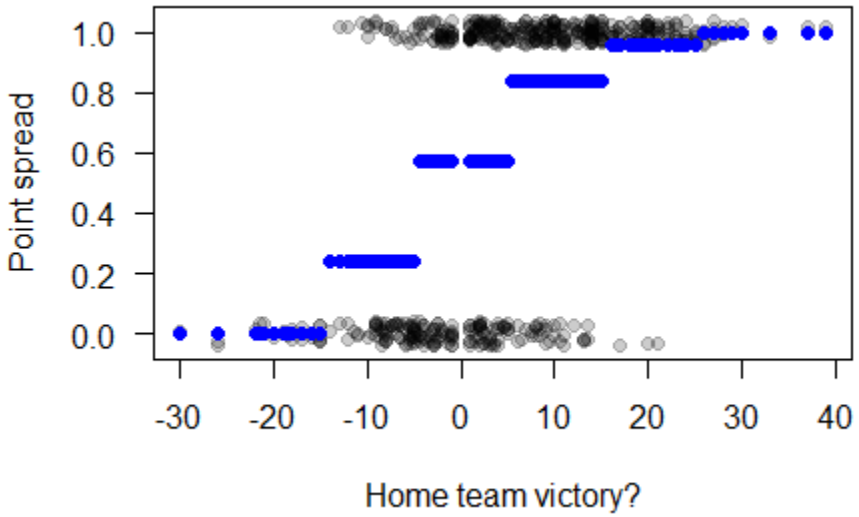
Today we'll be learning about logistic regression

- We want to build a regression model for a binary outcome
- On Wednesday we'll have a short review and talk about some big picture things on everything we've learned. We'll also cover common statistical fallacies in the real world.

Linear Regression for Binary Outcomes

A) Import bballbets.csv and bballbets.R into R Studio
   - The data set shows the point spread of various NCAA basketball games and whether or not the home team won. Negative spread indicates the number of points by which the visiting team was favored, and positive spread indicates the number of points by which the home team was favored
   - Plot the data, and add some jitter so that the plot is easier to read

B) Looking at the empirical win frequency within "buckets"
   - The next part of the code cuts the data points into several buckets based on point spread, and uses these buckets to perform a linear regression. Plotting the results shows the probability of the home team winning a game based on the 10 point range that the spread falls in.



   - The main limitation here is that the regression does not provide useful information along a continuum – we can only predict the chance that the home team wins based on the range that the spread falls in

C) Fit a linear model to the data
   - Next, we created a linear model through the lm() function and plotted the model alongside the data to get a model in the form of $\mathbf{y_i(hat) = \beta_0 + \beta_1 x_i}$

- This model (a linear probability model) predicts the probability of a success (in this example, the probability of the home team winning the game) based on a given x value (the point spread)
- Using this model, the home team wins 52% of the time when the teams are equally matched (there is no spread on the game). For each point of spread added in favor of the home team, the probability of a win increases by 2.35% for the home team.
- Theoretical Background: This is the Conditional Expected Value of the binary outcome:

  For a Binary outcome:

x_i	w_i
0: loss	-
1: win	$\beta_0 + \beta_1 x_i$

  $E(y|x) = \text{sum of all } (x_i)(w_i) = 1(\beta_0 + \beta_1 x_i) + 0(w_1) = \beta_0 + \beta_1 x_i$

- This kind of model does have statistical applications
  - Google uses a linear probability model to estimate the likelihood of a user clicking on a particular advertisement
  - Facebook uses a linear probability model to make friend recommendations
- However, there are some issues with this type of model
  - The model does not fit the extreme values well (or at all)
  - The model can give us values that are mathematically impossible:
    - $P(y_i = 1|x_i)$ should be on (0,1). (in other words, the probability of success needs to be somewhere between 0 and 1)
    - However, $\beta_0 + \beta_1 x_i$ can be any real number
    - This is why logistic regression is useful

Logistic Regression and the Generalized Linear Model

- We need something that fundamentally can't be less than 0 or greater than one – a function that is an s-shaped curve.
- The answer to this is the logistic regression **(generalized linear model)**:

$$P(y_i = 1|x_i) = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}$$

  - First, compute the ordinary regression you would normally do for the model ($\beta_0 + \beta_1 x_i$). This is known as the linear predictor portion of the model
  - Then, plug the regression into g(s), the link function

$$g(s) = \frac{e^s}{1 + e^s}$$

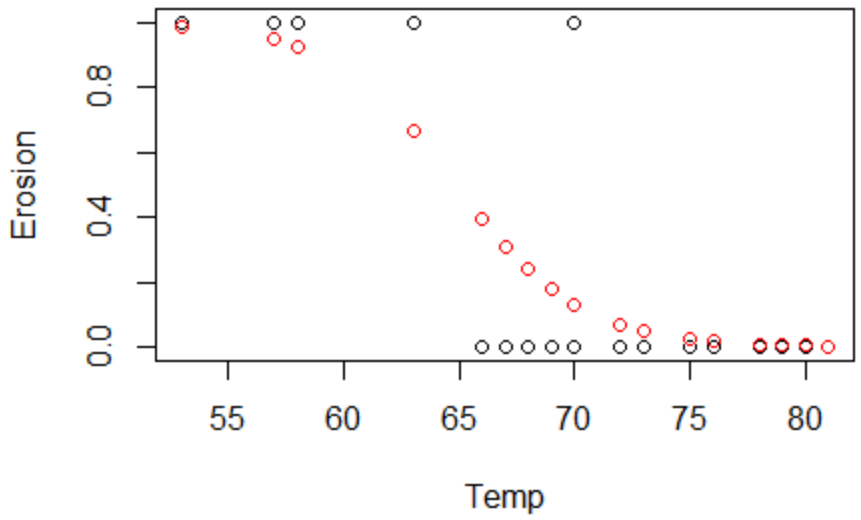
    - For very large values of s, g(s) approaches 1

  - For very small values of s, g(s) approaches 0, but is still positive
  - When the linear predictor part is equal to 0, g(s) = ½, which means that the outcome is equally likely to be a success or failure
- To fit a generalized linear model in R Studio, we use the glm() function.
  - Example: glm1 = glm(homewin~spread, data=bballbets, family=binomial)
  - The highlighted portion is the only difference from the lm()function
  - Plotting the regression, we can see that we get the s shaped curve

Example of the Generalized Linear Model

- In 1986, space shuttle Challenger exploded soon after liftoff. The disaster was caused by the abnormally cold weather (29^{o} F). The O rings, which are used to seal the sections of the rocket booster together, became hard and brittle in the cold. The O rings failed to provide an adequate seal, causing a catastrophic explosion a few seconds after takeoff.
  - Download and import the orings.csv data file. The data file contains information on the temperature at liftoff, and whether or not there was erosion on the O rings
  - Plot a graph of erosion vs. temperature at liftoff. We can see a general trend of O ring erosion at lower temperatures
  - The following code creates a generalized linear model for O ring erosion vs. temp:

```
glm1=glm(Erosion~Temp, family=binomial, data=orings)
plot(Erosion~Temp, data=orings)
points(fitted(glm1)~Temp, data=orings, col='red')
```



  - We can see that if NASA had this data and fit the generalized linear model, it would have been quite obvious that the conditions were too dangerous for the shuttle launch

- Using the predict function, the generalized linear model shows that there is a **99.99%** chance that there will be O ring failure when the temperature is 29 degrees. Below is the code to do this:

```
newdata = data.frame(Temp=29)
predict(glm1, newdata, type='response')
```