Maximum Likelihood Estimation in Logistic Regression

- The distribution of each observation y_i is

$$f_i(y_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}, i = 1, 2, \ldots, n$$

- The **likelihood function** is

$$L(\mathbf{y}, \boldsymbol{\beta}) = \prod_{i=1}^{n} f_i(y_i) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

- We usually work with the log-likelihood:

$$\ln L(\mathbf{y}, \boldsymbol{\beta}) = \ln \prod_{i=1}^{n} f_i(y_i) = \sum_{i=1}^{n}\left[y_i \ln\left(\frac{\pi_i}{1-\pi_i}\right)\right] + \sum_{i=1}^{n} \ln(1-\pi_i)$$

The log of a product is equal to the sum of the logs.

$y' = (y_1, y_2, \ldots, y_n)$

$B' = (B_0, B_1, \ldots, B_k)$ $p=k+1$

Maximum Likelihood Estimation in Logistic Regression

- The maximum likelihood estimators (MLEs) of the model parameters are those values that maximize the likelihood (or log-likelihood) function
- ML has been around since the first part of the previous century
- Often gives estimators that are intuitively pleasing
- MLEs have nice **properties**; unbiased (for large samples), minimum variance (or nearly so), and they have an approximate normal distribution when n is large

Maximum Likelihood Estimation in Logistic Regression

- Solving the ML score equations in logistic regression isn't quite as easy, because

$$\mu_i = \frac{n_i}{1+\exp(-\mathbf{x}_i'\boldsymbol{\beta})}, i = 1, 2, \ldots, n$$

n_i = # obs at x_i

sum(n_i) = n

- Logistic regression is a nonlinear model
- It turns out that the solution is actually fairly easy, and is based on **iteratively reweighted least squares** or **IRLS**
- An iterative procedure is necessary because parameter estimates must be updated from an initial "guess" through several steps
- Weights are necessary because the variance of the observations is not constant
- The weights are functions of the unknown parameters