Chapter 1

Logistic Regression and Newton-Raphson

1.1 Introduction

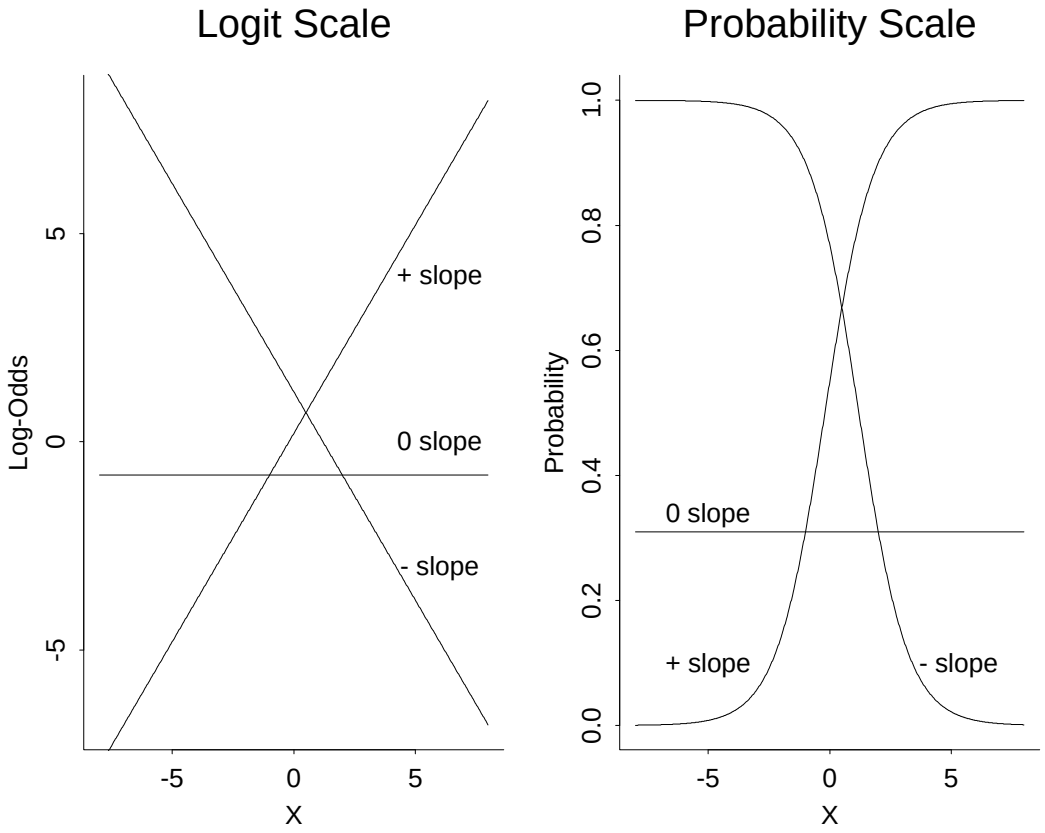
The logistic regression model is widely used in biomedical settings to model the probability of an event as a function of one or more predictors. For a single predictor X model stipulates that the log odds of "success" is

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

or, equivalently, as

$$p \;=\; \frac{\exp(\beta_0 + \beta_1 X)}{1 + \exp(\beta_0 + \beta_1 X)}$$

where p is the event probability. Depending on the sign of β_1, p either increases or decreases with X and follows a "sigmoidal" trend. If $\beta_1 = 1$ then p does not depend on X.



Note that the logit transformation is undefined when $\hat{p} = 0$ or $\hat{p} = 1$. To overcome this problem, researchers use the **empirical logits**, defined by $\log\{(\hat{p} + 0.5/n)/(1 - \hat{p} + 0.5/n)\}$, where n is the sample size or the number of observations on which $\hat{p}$ is based.

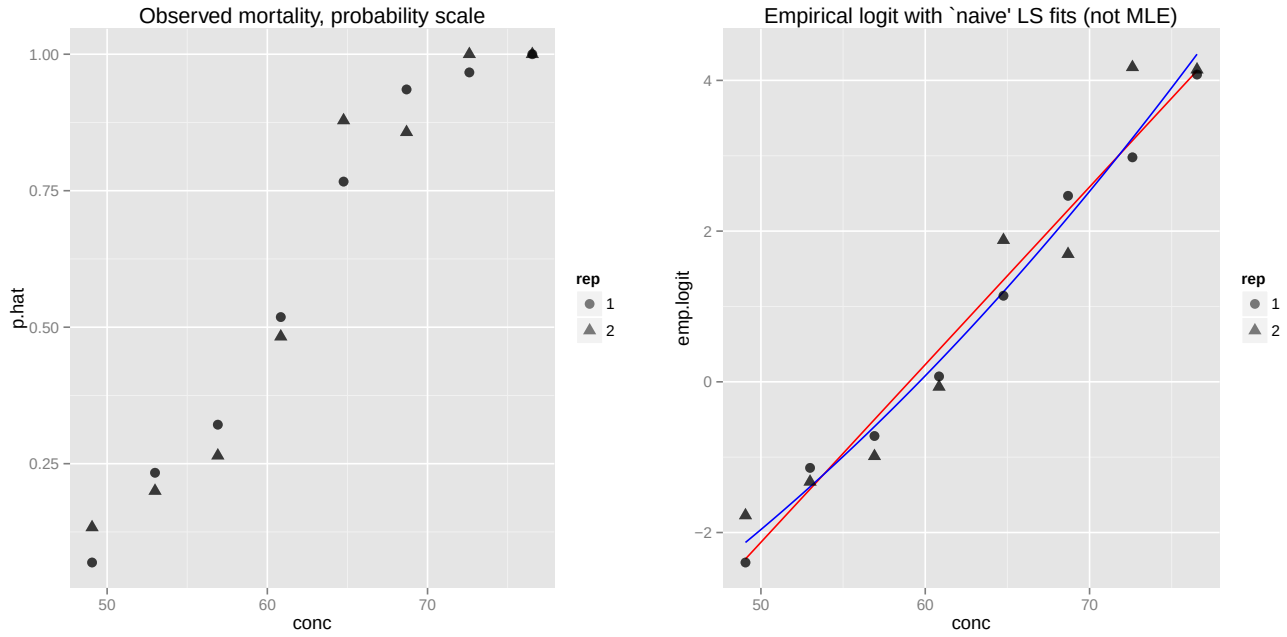
Example: Mortality of confused flour beetles The aim of an experiment originally reported by Strand (1930) and quoted by Bliss (1935) was to assess the response of the confused flour beetle, *Tribolium confusum*, to gaseous carbon disulphide (CS_2). In the experiment, prescribed volumes of liquid carbon disulphide were added to flasks in which a tubular cloth cage containing a batch of about thirty beetles was suspended. Duplicate batches of beetles were used for each concentration of CS_2. At the end of a five-hour period, the proportion killed was recorded and the actual concentration of gaseous CS_2 in the flask, measured in mg/l, was

determined by a volumetric analysis. The mortality data are given in the table below.

```
## Beetles data set
# conc = CS2 concentration
# y    = number of beetles killed
# n    = number of beetles exposed
# rep  = Replicate number (1 or 2)
beetles <- read.table("http://statacumen.com/teach/SC1/SC1_11_beetles.dat", header = TRU
beetles$rep   <- factor(beetles$rep)
```

	conc	y	n	rep		conc	y	n	rep
1	49.06	2	29	1	9	49.06	4	30	2
2	52.99	7	30	1	10	52.99	6	30	2
3	56.91	9	28	1	11	56.91	9	34	2
4	60.84	14	27	1	12	60.84	14	29	2
5	64.76	23	30	1	13	64.76	29	33	2
6	68.69	29	31	1	14	68.69	24	28	2
7	72.61	29	30	1	15	72.61	32	32	2
8	76.54	29	29	1	16	76.54	31	31	2

Plot the observed probability of mortality and the empirical logits with linear and quadratic LS fits (which are not the same as the logistic MLE fits).



In a number of articles that refer to these data, the responses from the first two concentrations are omitted because of apparent non-linearity. Bliss himself remarks that

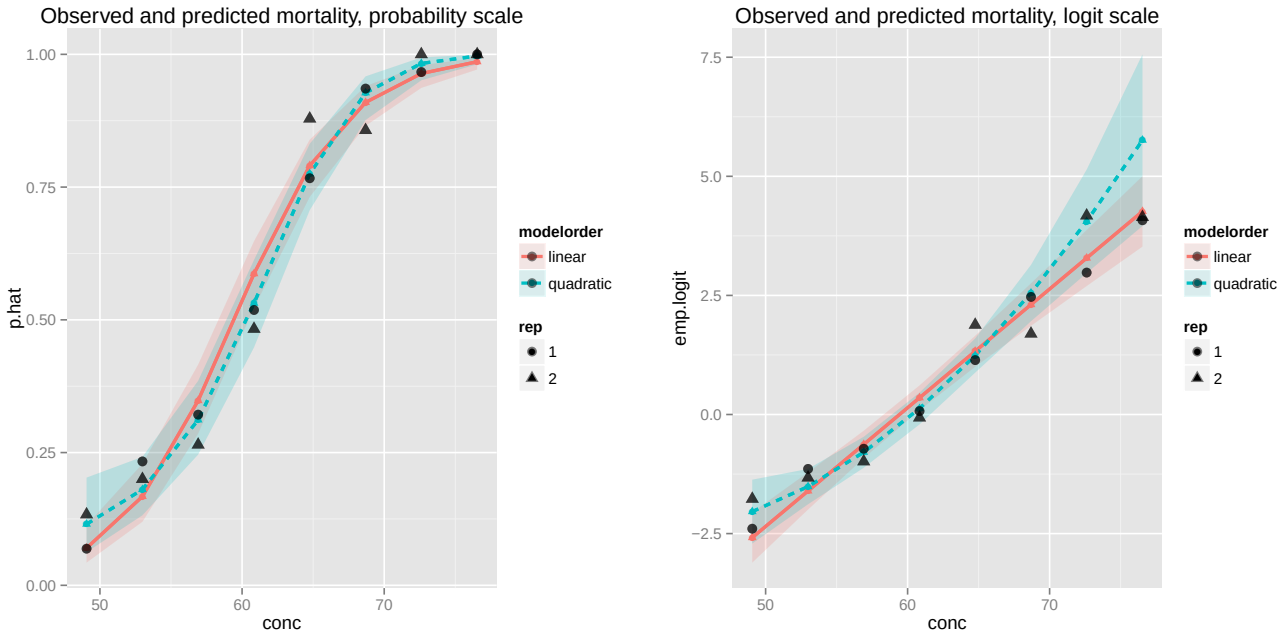
> ...in comparison with the remaining observations, the two lowest concentrations gave an exceptionally high kill. Over the remaining concentrations, the plotted values seemed to form a moderately straight line, so that the data were handled as two separate sets, only the results at 56.91 mg of CS_2 per litre being included in both sets.

However, there does not appear to be any biological motivation for this and so here they are retained in the data set.

Combining the data from the two replicates and plotting the empirical logit of the observed proportions against concentration gives a relationship that is better fit by a quadratic than a linear relationship,

$$\log\left(\frac{p}{1-p}\right) \;=\; \beta_0 + \beta_1 X + \beta_2 X^2.$$

The right plot below shows the linear and quadratic model fits to the observed values with point-wise 95% confidence bands on the logit scale, and on the left is the same on the proportion scale.



We will focus on how to estimate parameters of a logistic regression model using maximum likelihood (MLEs).

1.2 The Model

Suppose $Y_i \overset{\text{ind}}{\sim} \text{Binomial}(m_i, p_i)$ random variables, $i = 1, 2, \ldots, n$. For example, Y_i is the number of beetle deaths from a total of m_i beetles at concentration X_i over the $i = 1, 2, \ldots, n$ concentrations. Note that m_i can equal 1 (and often does in observational studies). Recall that the probability mass function for a Binomial is

$$\Pr[Y_i = y_i | p_i] = \binom{m_i}{y_i} p_i^{y_i} (1 - p_i)^{m_i - y_i}, \quad y_i = 0, 1, 2, \ldots, m_i.$$

So the joint distribution of $Y_1, Y_2, \ldots, Y_n$ is

$$\Pr[Y_1 = y_1, \ldots, Y_n = y_n | p_1, \ldots, p_n] = \prod_{i=1}^{n} \binom{m_i}{y_i} p_i^{y_i} (1 - p_i)^{m_i - y_i}.$$

The log-likelihood, ignoring the constant, is

$$\begin{aligned}
\ell &= \log\{\Pr[Y_1 = y_1, \ldots, Y_n = y_n | p_1, \ldots, p_n]\} \\
&\propto \log\left\{\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{m_i - y_i}\right\} \\
&= \sum_{i=1}^{n}\{y_i \log(p_i) + (m_i - y_i)\log(1-p_i)\} \\
&= \sum_{i=1}^{n}\left\{m_i \log(1-p_i) + y_i \log\left(\frac{p_i}{1-p_i}\right)\right\}. \qquad (1.1)
\end{aligned}$$

The logistic regression model assumes that p_i depends on r covariates $x_{i1}, x_{i2}, \ldots, x_{ir}$ through

$$\begin{aligned}
\log\left(\frac{p_i}{1-p_i}\right) &= \beta_0 + \beta_1 x_{i1} + \cdots + \beta_r x_{ir} \\
&= \begin{bmatrix} 1 & x_{i1} & x_{i2} & \cdots & x_{ir} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_r \end{bmatrix} \\
&= \underset{\sim}{x}_i^{\top} \underset{\sim}{\beta}.
\end{aligned}$$

The covariates or predictors are fixed, while $\underset{\sim}{\beta}$ is an unknown parameter vector. Regardless, p_i is a function of both $\underset{\sim}{x}_i$ and $\underset{\sim}{\beta}$,

$$p_i \equiv p_i(\underset{\sim}{x}_i, \underset{\sim}{\beta}) \quad \text{or} \quad p_i(\underset{\sim}{\beta}) \quad \text{(suppressing } \underset{\sim}{x}_i\text{, since it is known)}.$$

Note that the model implies

$$\begin{aligned} p_i &= \frac{\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})}{1+\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})} \qquad \text{and} \\ 1-p_i &= \frac{1}{1+\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})}. \end{aligned}$$

To obtain the MLEs we first write the log-likelihood in (1.1) as a function of $\underset{\sim}{\beta}$,

$$\begin{aligned} \ell(\underset{\sim}{\beta}) &= \sum_{i=1}^{n} \left\{ m_i \log\left(\frac{1}{1+\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})}\right) + y_i \log\left(\frac{\frac{\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})}{1+\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})}}{\frac{1}{1+\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})}}\right) \right\} \\ &= \sum_{i=1}^{n} \left\{ m_i \log\left(\frac{1}{1+\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})}\right) + y_i(\underset{\sim}{x}_i^\top \underset{\sim}{\beta}) \right\} \\ &= \sum_{i=1}^{n} \left\{ y_i(\underset{\sim}{x}_i^\top \underset{\sim}{\beta}) - m_i \log(1+\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})) \right\}. \end{aligned} \tag{1.2}$$

To maximize $\ell(\underset{\sim}{\beta})$, we compute the score function

$$\dot{\ell}(\underset{\sim}{\beta}) = \begin{bmatrix} \partial\ell(\underset{\sim}{\beta})/\partial\beta_0 \\ \partial\ell(\underset{\sim}{\beta})/\partial\beta_1 \\ \vdots \\ \partial\ell(\underset{\sim}{\beta})/\partial\beta_r \end{bmatrix}$$

and solve the likelihood equations

$$\dot{\ell}(\underset{\sim}{\beta}) = \underset{\sim}{0}_{r+1}.$$

Note that $\dot{\ell}(\underset{\sim}{\beta})$ is an $(r+1)$-by-1 vector, so we are solving a system of $r+1$ non-linear equations.

Let us now compute $\partial\ell(\underset{\sim}{\beta})/\partial\beta_j$ where β_j is a generic element of $\underset{\sim}{\beta}$. It is important to realize that $\ell(\underset{\sim}{\beta})$ depends on the elements of $\underset{\sim}{\beta}$ only through the values of $\underset{\sim}{x}_i$, which is linear. Thus each of the partial derivatives in $\dot{\ell}(\underset{\sim}{\beta})$ will have the same form!

Now

$$\frac{\partial\ell(\underset{\sim}{\beta})}{\partial\beta_j} = \sum_{i=1}^{n}\left\{y_i\frac{\partial}{\partial\beta_j}(\underset{\sim}{x}_i^\top\underset{\sim}{\beta}) - m_i\frac{\partial}{\partial\beta_j}\log(1+\exp(\underset{\sim}{x}_i^\top\underset{\sim}{\beta}))\right\} \quad (1.3)$$

where

$$\begin{aligned}\frac{\partial}{\partial\beta_j}(\underset{\sim}{x}_i^\top\underset{\sim}{\beta}) &= \frac{\partial}{\partial\beta_j}\{\beta_0+\beta_1x_{i1}+\cdots+\beta_rx_{ir}\}\\ &= x_{ij} \qquad (\text{where } x_{i0}\equiv 1)\end{aligned} \quad (1.4)$$

and

$$\begin{aligned}\frac{\partial}{\partial\beta_j}\log(1+\exp(\underset{\sim}{x}_i^\top\underset{\sim}{\beta})) &= \frac{\frac{\partial}{\partial\beta_j}\exp(\underset{\sim}{x}_i^\top\underset{\sim}{\beta})}{1+\exp(\underset{\sim}{x}_i^\top\underset{\sim}{\beta})}\\ &= \frac{\exp(\underset{\sim}{x}_i^\top\underset{\sim}{\beta})}{1+\exp(\underset{\sim}{x}_i^\top\underset{\sim}{\beta})}\frac{\partial}{\partial\beta_j}(\underset{\sim}{x}_i^\top\underset{\sim}{\beta})\\ &= p_i(\underset{\sim}{x}_i,\underset{\sim}{\beta})x_{ij},\end{aligned} \quad (1.5)$$

and so

$$\begin{aligned}\frac{\partial\ell(\underset{\sim}{\beta})}{\partial\beta_j} &= \sum_{i=1}^{n}\left\{y_ix_{ij}-m_ip_i(\underset{\sim}{x}_i,\underset{\sim}{\beta})x_{ij}\right\}\\ &= \sum_{i=1}^{n}\left\{x_{ij}(y_i-m_ip_i(\underset{\sim}{x}_i,\underset{\sim}{\beta}))\right\}, \quad j=0,1,\ldots,r.\end{aligned} \quad (1.6)$$

For NR, we also need the second partial derivatives

$$\begin{aligned}\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} &= \frac{\partial}{\partial \beta_k}\frac{\partial \ell(\underset{\sim}{\beta})}{\partial \beta_j} \\ &= \sum_{i=1}^{n}\left\{x_{ij}\left(y_i - m_i\frac{\partial p_i(\underset{\sim}{x}_i, \underset{\sim}{\beta})}{\partial \beta_k}\right)\right\}.\end{aligned}$$

It is straightforward to show

$$\frac{\partial p_i(\underset{\sim}{x}_i, \underset{\sim}{\beta})}{\partial \beta_k} = \underset{\sim}{x}_{ik} p_i(\underset{\sim}{x}_i, \underset{\sim}{\beta})(1 - p_i(\underset{\sim}{x}_i, \underset{\sim}{\beta})).$$

So

$$\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} = -\sum_{i=1}^{n}\left\{x_{ij}x_{ik}m_i p_i(\underset{\sim}{x}_i, \underset{\sim}{\beta})(1 - p_i(\underset{\sim}{x}_i, \underset{\sim}{\beta}))\right\}.$$

Recall that $\mathrm{Var}(Y_i) = m_i p_i(\underset{\sim}{x}_i, \underset{\sim}{\beta})(1 - p_i(\underset{\sim}{x}_i, \underset{\sim}{\beta}))$, from the variance of the binomial distribution. Let $\mathrm{Var}(Y_i) = v_i(\underset{\sim}{\beta}) = v_i(\underset{\sim}{x}_i, \underset{\sim}{\beta})$.

For programming, it is convenient to use vector/matrix notation. Let

$$\underset{\sim}{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \qquad \underset{\sim}{p} = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} \qquad \underset{\sim}{m} = \begin{bmatrix} m_1 \\ \vdots \\ m_n \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} \underset{\sim}{x}_1^\top \\ \vdots \\ \underset{\sim}{x}_n^\top \end{bmatrix} \qquad \log\left(\frac{\underset{\sim}{p}}{1 - \underset{\sim}{p}}\right) = \begin{bmatrix} \log\left(\frac{p_1}{1-p_1}\right) \\ \vdots \\ \log\left(\frac{p_n}{1-p_n}\right) \end{bmatrix} \quad \text{operate elementwise.}$$

The model can be written

$$\log\left(\frac{\underset{\sim}{p}}{1 - \underset{\sim}{p}}\right) = \mathbf{X}\underset{\sim}{\beta},$$

or, for the ith element,

$$\log\left(\frac{p_i}{1-p_i}\right) \;=\; \underset{\sim}{x}_i^\top \underset{\sim}{\beta}.$$

Also, define vectors

$$\exp(\mathbf{X}\underset{\sim}{\beta}) \;=\; \begin{bmatrix} \exp(\underset{\sim}{x}_1^\top \underset{\sim}{\beta}) \\ \vdots \\ \exp(\underset{\sim}{x}_n^\top \underset{\sim}{\beta}) \end{bmatrix} \qquad \text{implies} \quad \underset{\sim}{p} \;=\; \frac{\exp(\mathbf{X}\underset{\sim}{\beta})}{1+\exp(\mathbf{X}\underset{\sim}{\beta})}$$

$$\log(\underset{\sim}{1}+\exp(\mathbf{X}\underset{\sim}{\beta})) \;=\; \begin{bmatrix} \log(1+\exp(\underset{\sim}{x}_1^\top \underset{\sim}{\beta})) \\ \vdots \\ \log(1+\exp(\underset{\sim}{x}_n^\top \underset{\sim}{\beta})) \end{bmatrix},$$

where operations are performed elementwise.

Then

$$\begin{aligned} \ell(\underset{\sim}{\beta}) \;&=\; \sum_{i=1}^n \{y_i \log(p_i) + (m_i - y_i)\log(1-p_i)\} \\ &= \underset{\sim}{y}^\top \log(\underset{\sim}{p}) + (\underset{\sim}{m}-\underset{\sim}{y})^\top \log(1-\underset{\sim}{p}) \\ &=\; \sum_{i=1}^n \left\{ y_i \underset{\sim}{x}_i^\top \underset{\sim}{\beta} - m_i \log(1+\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})) \right\} \\ &= \underset{\sim}{y}^\top \mathbf{X}\underset{\sim}{\beta} - \underset{\sim}{m}^\top \log(1+\exp(\mathbf{X}\underset{\sim}{\beta})) \end{aligned} \tag{1.7}$$

and

$$\dot{\ell}(\underset{\sim}{\beta}) \;=\; \begin{bmatrix} \partial\ell(\underset{\sim}{\beta})/\partial\beta_0 \\ \partial\ell(\underset{\sim}{\beta})/\partial\beta_1 \\ \vdots \\ \partial\ell(\underset{\sim}{\beta})/\partial\beta_r \end{bmatrix} \;=\; \mathbf{X}^\top(\underset{\sim}{y} - \underset{\sim}{m} \circ \underset{\sim}{p}(\underset{\sim}{\beta})),$$

where $\circ$ denotes the Hadamard or elementwise product, so that

$$\underset{\sim}{m} \circ \underset{\sim}{p}(\underset{\sim}{\beta}) \;=\; \begin{bmatrix} m_1 p_1(\underset{\sim}{\beta}) \\ \vdots \\ m_n p_n(\underset{\sim}{\beta}) \end{bmatrix}.$$

If we think of

$$\mathrm{E}[\underset{\sim}{Y}] \;=\; \begin{bmatrix} \mathrm{E}[Y_1] \\ \vdots \\ \mathrm{E}[Y_n] \end{bmatrix} \;=\; \begin{bmatrix} m_1 p_1(\underset{\sim}{\beta}) \\ \vdots \\ m_n p_n(\underset{\sim}{\beta}) \end{bmatrix} \;=\; \begin{bmatrix} \mu_1(\underset{\sim}{\beta}) \\ \vdots \\ \mu_n(\underset{\sim}{\beta}) \end{bmatrix} \;\equiv\; \underset{\sim}{\mu}(\underset{\sim}{\beta}).$$

then the likelihood equations have the form

$$\dot{\ell}(\underset{\sim}{\beta}) \;=\; \mathbf{X}^\top(\underset{\sim}{y} - \underset{\sim}{m} \circ \underset{\sim}{p}(\underset{\sim}{\beta})) \;=\; \mathbf{X}^\top(\underset{\sim}{y} - \underset{\sim}{\mu}(\underset{\sim}{\beta})) \;=\; \underset{\sim}{0}.$$

This is the same form as the "Normal equations" for computing LS estimates normal-theory regression. Also, with

$$\ddot{\ell}(\underset{\sim}{\beta}) \;=\; \left[\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k}\right] \;=\; -\sum_{i=1}^{n} \left\{ x_{ij} x_{ik} v_i(\underset{\sim}{\beta}) \right\},$$

if we define the diagonal matrix

$$\mathbf{v}(\underset{\sim}{\beta}) \;=\; \mathrm{diag}(v_1(\underset{\sim}{\beta}), v_2(\underset{\sim}{\beta}), \ldots, v_n(\underset{\sim}{\beta})) \;=\; \begin{bmatrix} v_1(\underset{\sim}{\beta}) & & & 0 \\ & v_2(\underset{\sim}{\beta}) & & \\ & & \ddots & \\ 0 & & & v_n(\underset{\sim}{\beta}) \end{bmatrix},$$

then it is easy to see that

$$\ddot{\ell}(\underset{\sim}{\beta}) \;=\; -\mathbf{X}^\top \mathbf{v}(\underset{\sim}{\beta}) \mathbf{X},$$

that is, the jth row and kth column element of $\mathbf{X}^\top \mathbf{v}(\underset{\sim}{\beta})\mathbf{X}$ is $\sum_{i=1}^{n} x_{ij} x_{ik} v_i(\underset{\sim}{\beta})$.

It is important to recognize that for the logistic regression model

$$\mathbf{I}(\underset{\sim}{\beta}) \;=\; \mathrm{E}[-\ddot{\ell}(\underset{\sim}{\beta})] \;=\; \mathbf{X}^\top \mathbf{v}(\underset{\sim}{\beta})\mathbf{X} \;=\; -\ddot{\ell}(\underset{\sim}{\beta}),$$

that is, NR and Scoring methods are equivalent. In particular, the NR methods iterates via

$$\begin{aligned}\underset{\sim}{\hat{\beta}}_{i+1} &= \underset{\sim}{\hat{\beta}}_i - [\ddot{\ell}(\underset{\sim}{\hat{\beta}}_i)]^{-1}\dot{\ell}(\underset{\sim}{\hat{\beta}}_i) \\ &= \underset{\sim}{\hat{\beta}}_i + (\mathbf{X}^\top \mathbf{v}(\underset{\sim}{\beta})\mathbf{X})^{-1}\mathbf{X}^\top(\underset{\sim}{y} - \underset{\sim}{\mu}(\underset{\sim}{\beta})), \quad i = 0, 1, \ldots,\end{aligned}$$

until convergence (hopefully) to the MLE $\underset{\sim}{\hat{\beta}}$.

I will note that the observed information matrix $\ddot{\ell}(\underset{\sim}{\beta})$ is independent of $\underset{\sim}{Y}$ for logistic regression with the logit link, but not for other binomial response models, such as probit regression. Thus, for other models there is a difference between NR and Fisher Scoring. Many packages, including SAS, use Fisher Scoring as default.

For logistic regression, large sample theory indicates that the MLE $\underset{\sim}{\hat{\beta}}$ has an approximate multivariate normal distribution

$$\underset{\sim}{\hat{\beta}} \;\dot{\sim}\; \mathrm{Normal}_{r+1}(\underset{\sim}{\beta}, \mathbf{I}^{-1}(\underset{\sim}{\hat{\beta}}))$$

where

$$\mathbf{I}^{-1}(\underset{\sim}{\hat{\beta}}) \;\dot{\sim}\; (\mathbf{X}^\top \mathbf{v}(\underset{\sim}{\hat{\beta}})\mathbf{X})^{-1}.$$

This result can be used to get estimated standard deviations for each regression coefficient and p-values for testing significance of effects. In particular, if

$$\sigma_j(\underset{\sim}{\hat{\beta}}) \;=\; \sqrt{i\text{th diagonal element of } \mathbf{I}^{-1}(\underset{\sim}{\hat{\beta}})}$$

then

$$\hat{\beta}_j \;\dot{\sim}\; \mathrm{Normal}(\beta_j, \sigma_j^2(\underset{\sim}{\hat{\beta}})).$$

A p-value for testing $H_0 : \hat{\beta}_j = 0$ can be based on

$$\frac{\hat{\beta}_j - 0}{\sigma_j(\underset{\sim}{\hat{\beta}})} \;\dot{\sim}\; \text{Normal}(0, 1).$$

General remarks

1. There is an extensive literature on conditions for existence and uniqueness of MLEs for logistic regression.

2. MLEs may not exist. One case is when you have "separation" of covariates (e.g., all successes to left and all failures to right for some value of x).

3. Convergence is sensitive to starting values.

   For the model

   $$\log\left(\frac{p_i}{1-p_i}\right) \;=\; \beta_0 + \beta_1 x_{i1} + \cdots + \beta_r x_{ir}$$

   the following starting values often work well, especially if regression effects are not too strong:

   $$\begin{aligned}
   \beta_{0\text{ start}} &= \log\left(\frac{\tilde{p}}{1-\tilde{p}}\right) \\
   &= \log\left(\frac{\sum_{i=1}^n \frac{y_i}{m_i}}{1-\sum_{i=1}^n \frac{y_i}{m_i}}\right) \\
   &= \log\left(\frac{\sum_{i=1}^n y_i}{\sum_{i=1}^n (m_i - y_i)}\right),
   \end{aligned}$$

   and $\beta_{1\text{ start}} = \cdots = \beta_{r\text{ start}} = 0$, where $\tilde{p} = \sum_{i=1}^n \frac{y_i}{m_i}$ is the overall proportion. This is the MLE for β_0 if $\hat{\beta}_1 = \cdots = \hat{\beta}_r = 0$.

4. If you have two observations $Y_1 \overset{\text{ind}}{\sim} \text{Binomial}(m_1, p)$ and $Y_2 \overset{\text{ind}}{\sim} \text{Binomial}(m_2, p)$ with the same success probability p, then the log-likelihood (excluding constants) is the same regardless of whether you treat Y_1 and Y_2 as separate binomial observations or you combine them as $Y_1 + Y_2 \overset{\text{ind}}{\sim} \text{Binomial}(m_1 + m_2, p)$. More generally, Bernoulli observations with the same covariate vector can be combined into a single binomial response (provided observations are independent) when defining the log-likelihood.

1.3 Implementation

Function `f.lr.p()` computes the probability vector under a logistic regression model

$$p_i = \frac{\exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})}{1 + \exp(\underset{\sim}{x}_i^\top \underset{\sim}{\beta})}$$

from the design matrix $\mathbf{X}$ and regression vector $\underset{\sim}{\beta}$. The function assumes that $\mathbf{X}$ and $\underset{\sim}{\beta}$ are of the correct dimensions.

```
f.lr.p <- function(X, beta) {
  # compute vector p of probabilities for logistic regression with logit link

  X <- as.matrix(X)
  beta <- as.vector(beta)
  p <- exp(X %*% beta) / (1 + exp(X %*% beta))
  return(p)
}
```

Function `f.lr.l()` computes the binomial log-likelihood function

$$\ell \propto \sum_{i=1}^{n} \{y_i \log(p_i) + (m_i - y_i) \log(1 - p_i)\} \tag{1.8}$$

from three input vectors: the counts $\underset{\sim}{y}$, the sample sizes $\underset{\sim}{m}$, and the probabilities $\underset{\sim}{p}$. The function is arbitrary, working for all Binomial models.

```
f.lr.l <- function(y, m, p) {
  # binomial log likelihood function
  # input:   vectors: y = counts; m = sample sizes; p = probabilities
  # output: log-likelihood l, a scalar

  l <- t(y) %*% log(p) + t(m - y) %*% log(1 - p)
  return(l)
}
```

The Fisher's scoring routine for logistic regression `f.lr.FS()` finds the MLE $\hat{\underset{\sim}{\beta}}$ (without line-search), following from the derivation above.

Convergence is based on the number of iterations, `maxit = 50`, Euclidean distance between successive iterations of $\hat{\underset{\sim}{\beta}}$, `eps1`, and distance between successive iterations of the log-likelihood, `eps2`. The absolute difference in log-likelihoods between successive steps is new for us, but a sensible addition.

Comments

1. The iteration scheme

$$\begin{aligned} \hat{\underset{\sim}{\beta}}_{i+1} &= \hat{\underset{\sim}{\beta}}_i + (\mathbf{X}^\top \mathbf{v}(\underset{\sim}{\beta})\mathbf{X})^{-1}\mathbf{X}^\top(\underset{\sim}{y} - \underset{\sim}{\mu}(\underset{\sim}{\beta})) \\ &= \hat{\underset{\sim}{\beta}}_i + (\text{inverse Info})(\text{Score func}) \end{aligned}$$

   is implemented below in two ways. The commented method takes the inverse of the information matrix, which can be computationally intensive and (occasionally) numerically unstable. The uncommented method solves

$$(\mathbf{X}^\top \mathbf{v}(\underset{\sim}{\beta})\mathbf{X})(\hat{\underset{\sim}{\beta}}_{i+1} - \hat{\underset{\sim}{\beta}}_i) = \mathbf{X}^\top(\underset{\sim}{y} - \underset{\sim}{\mu}(\underset{\sim}{\beta}))$$

   for $(\text{increm}) = (\hat{\underset{\sim}{\beta}}_{i+1} - \hat{\underset{\sim}{\beta}}_i)$. The new estimate is $\hat{\underset{\sim}{\beta}}_{i+1} = \hat{\underset{\sim}{\beta}}_i + (\text{increm})$.

2. Line search is implemented by evaluating the log-likelihood over a range $(-1, 2)$ of α step sizes and choosing the step that gives the largest log-likelihood.

3. It calls both `f.lr.l()`, the function to calculate log-likelihood, and `f.lr.p()`, the function to compute vector p of probabilities for LR.

```
f.lr.FS <- function(X, y, m, beta.1
                    , eps1 = 1e-6, eps2 = 1e-7, maxit = 50) {
  # Fisher's scoring routine for estimation of LR model (with line search)
  # Input:
  #   X        = n-by-(r+1) design matrix
  #   y        = n-by-1 vector of success counts
  #   m        = n-by-1 vector of sample sizes
  #   beta.1   = (r+1)-by-1 vector of starting values for regression est
  # Iteration controlled by:
  #   eps1  = absolute convergence criterion for beta
  #   eps2  = absolute convergence criterion for log-likelihood
  #   maxit = maximum allowable number of iterations
  # Output:
  #   out   = list containing:
  #     beta.MLE  = beta MLE
  #     NR.hist   = iteration history of convergence differences
  #     beta.hist = iteration history of beta
  #     beta.cov  = beta covariance matrix (inverse Fisher's information matrix at MLE)
  #     note      = convergence note

  beta.2 <- rep(-Inf, length(beta.1)) # init beta.2
  diff.beta <- sqrt(sum((beta.1 - beta.2)^2)) # Euclidean distance

  llike.1 <- f.lr.l(y, m, f.lr.p(X, beta.1)) # update loglikelihood
  llike.2 <- f.lr.l(y, m, f.lr.p(X, beta.2)) # update loglikelihood
  diff.like <- abs(llike.1 - llike.2) # diff
  if (is.nan(diff.like)) { diff.like <- 1e9 }

  i <- 1                              # initial iteration index

  alpha.step <- seq(-1, 2, by = 0.1)[-11]  # line search step sizes, excluding 0

  NR.hist <- data.frame(i, diff.beta, diff.like, llike.1, step.size = 1) # iteration history
  beta.hist <- matrix(beta.1, nrow = 1)
  while ((i <= maxit) & (diff.beta > eps1) & (diff.like > eps2)) {
```

```
  i <- i + 1                        # increment iteration

  # update beta
  beta.2 <- beta.1                  # old guess is current guess
  mu.2 <- m * f.lr.p(X, beta.2)     # m * p is mean
  # variance matrix
  v.2  <- diag(as.vector(m * f.lr.p(X, beta.2) * (1 - f.lr.p(X, beta.2))))
  score.2 <- t(X) %*% (y - mu.2)    # score function
  # this increment version inverts the information matrix
    # Iinv.2 <- solve(t(X) %*% v.2 %*% X)  # Inverse information matrix
    # increm <- Iinv.2 %*% score.2     # increment, solve() is inverse
  # this increment version solves for (beta.2-beta.1) without inverting Information
    increm <- solve(t(X) %*% v.2 %*% X, score.2) # solve for increment

  # line search for improved step size
  llike.alpha.step <- rep(NA, length(alpha.step)) # init llike for line search
  for (i.alpha.step in 1:length(alpha.step)) {
    llike.alpha.step[i.alpha.step] <- f.lr.l(y, m
                  , f.lr.p(X, beta.2 + alpha.step[i.alpha.step] * increm))
  }
  # step size index for max increase in log-likelihood (if tie, [1] takes first)
  ind.max.alpha.step <- which(llike.alpha.step == max(llike.alpha.step))[1]

  beta.1 <- beta.2 + alpha.step[ind.max.alpha.step] * increm   # update beta

  diff.beta <- sqrt(sum((beta.1 - beta.2)^2)) # Euclidean distance

  llike.2 <- llike.1                # age likelihood value
  llike.1 <- f.lr.l(y, m, f.lr.p(X, beta.1)) # update loglikelihood
  diff.like <- abs(llike.1 - llike.2) # diff

  # iteration history
  NR.hist   <- rbind(NR.hist, c(i, diff.beta, diff.like, llike.1, alpha.step[ind.max.a
  beta.hist <- rbind(beta.hist, matrix(beta.1, nrow = 1))
}

# prepare output
out <- list()
out$beta.MLE  <- beta.1
out$iter      <- i - 1
out$NR.hist   <- NR.hist
out$beta.hist <- beta.hist
  v.1  <- diag(as.vector(m * f.lr.p(X, beta.1) * (1 - f.lr.p(X, beta.1))))
  Iinv.1 <- solve(t(X) %*% v.1 %*% X)  # Inverse information matrix
out$beta.cov  <- Iinv.1
```

```
  if (!(diff.beta > eps1) & !(diff.like > eps2)) {
    out$note <- paste("Absolute convergence of", eps1, "for betas and"
                      , eps2, "for log-likelihood satisfied")
  }
  if (i > maxit) {
    out$note <- paste("Exceeded max iterations of ", maxit)
  }
  return(out)
}
```

1.3.1 Example (cont.): Mortality of confused flour beetles

Load the beetles dataset and fit quadratic model. The model is

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X + \beta_2 X^2.$$

where $X = \mathrm{CS}_2$ level.

```
## Beetles data set
# conc = CS2 concentration
# y    = number of beetles killed
# n    = number of beetles exposed
# rep  = Replicate number (1 or 2)
beet <- read.table("http://statacumen.com/teach/SC1/SC1_11_beetles.dat", header = TRUE)
beet$rep   <- factor(beet$rep)

# create data variables: m, y, X
n <- nrow(beet)
m <- beet$n
y <- beet$y
X.temp <- beet$conc

# quadratic model
X <- matrix(c(rep(1,n), X.temp, X.temp^2), nrow = n)
colnames(X) <- c("Int", "conc", "conc2")
r <- ncol(X) - 1  # number of regression coefficients - 1
```

```
# initial beta vector
beta.1 <- c(log(sum(y) / sum(m - y)), rep(0, r))

# fit betas using our Fisher Scoring function
out <- f.lr.FS(X, y, m, beta.1)
out
```

```
## $beta.MLE
##            [,1]
## Int    7.968410
## conc  -0.516593
## conc2  0.006372
##
## $iter
## [1] 6
##
## $NR.hist
##   i diff.beta diff.like llike.1 step.size
## 1 1       Inf       Inf  -322.7       1.0
## 2 2 2.531e+01 1.329e+02  -189.8       1.4
## 3 3 2.701e+01 6.658e+00  -183.2       1.2
## 4 4 4.931e+00 1.050e+00  -182.1       1.2
## 5 5 9.305e-01 8.664e-03  -182.1       1.0
## 6 6 6.066e-03 1.195e-06  -182.1       1.0
## 7 7 1.171e-06 8.527e-14  -182.1       0.9
##
## $beta.hist
##          [,1]    [,2]      [,3]
## [1,]   0.4263  0.0000  0.000000
## [2,] -24.8787  0.5947 -0.002996
## [3,]   2.1174 -0.2900  0.004244
## [4,]   7.0444 -0.4867  0.006130
## [5,]   7.9745 -0.5168  0.006373
## [6,]   7.9684 -0.5166  0.006372
## [7,]   7.9684 -0.5166  0.006372
##
## $beta.cov
##              Int      conc      conc2
## Int   121.80053 -4.115854  3.444e-02
## conc   -4.11585  0.139603 -1.172e-03
## conc2   0.03444 -0.001172  9.878e-06
```

Looking at the output we see that the routine converged in 6 iterations. At each step, the log-likelihood increased, and the norm of the difference

between successive estimates eventually decreased to zero. The estimates are 7.968 for the constant term, −0.5166 for the linear term, and 0.0064 for the quadratic term.

```
# create a parameter estimate table
beta.Est  <- out$beta.MLE
beta.SE   <- sqrt(diag(out$beta.cov)) # sqrt diag inverse Information matrix
beta.z    <- beta.Est / beta.SE
beta.pval <- 2 * pnorm(-abs(beta.z))

beta.coef <- data.frame(beta.Est, beta.SE, beta.z, beta.pval)
beta.coef
```

```
##        beta.Est   beta.SE beta.z beta.pval
## Int    7.968410 11.036328  0.722   0.47028
## conc  -0.516593  0.373635 -1.383   0.16678
## conc2  0.006372  0.003143  2.027   0.04262
```

Compare our parameter estimate table above to the one from the `glm()` function.

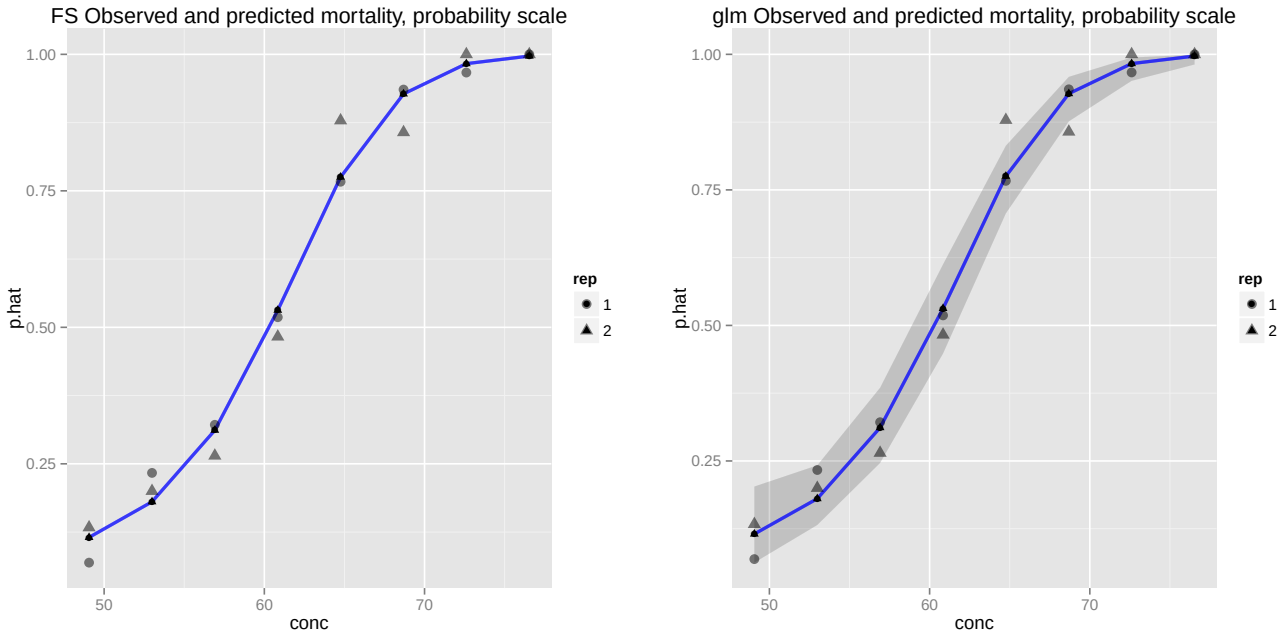
```
## compare to the glm() fit:
summary(glm.beetles2)$call
```

```
## glm(formula = cbind(y, n - y) ~ conc + conc2, family = binomial,
##     data = beetles)
```

```
summary(glm.beetles2)$coefficients
```

```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  7.968410  11.036327   0.722  0.47028
## conc        -0.516593   0.373635  -1.383  0.16678
## conc2        0.006372   0.003143   2.027  0.04262
```

Therefore, our model predictions match those from the beginning of the chapter using the `glm()` function.



Also note that the observed and fitted proportions are fairly close, which qualitatively suggests a reasonable model for the data.

1.3.2 Example: Leukemia white blood cell types

This example illustrates modeling with continuous and factor predictors.

Feigl and Zelen[1] reported the survival time in weeks and the white cell blood count (WBC) at time of diagnosis for 33 patients who eventually died of acute leukemia. Each person was classified as AG+ or AG−, indicating the presence or absence of a certain morphological characteristic in the white cells. Four variables are given in the data set: WBC, a binary factor or **indicator variable** AG (1 for AG+, 0 for AG−), NTOTAL (the number of patients with the given combination of AG and WBC),

[1]Feigl, P., Zelen, M. (1965) Estimation of exponential survival probabilities with concomitant information. Biometrics 21, 826–838. Survival times are given for 33 patients who died from acute myelogenous leukaemia. Also measured was the patient's white blood cell count at the time of diagnosis. The patients were also factored into 2 groups according to the presence or absence of a morphologic characteristic of white blood cells. Patients termed AG positive were identified by the presence of Auer rods and/or significant granulation of the leukaemic cells in the bone marrow at the time of diagnosis.

and NRES (the number of NTOTAL that survived at least one year from the time of diagnosis).

The researchers are interested in modelling the probability p of surviving at least one year as a function of WBC and AG. They believe that WBC should be transformed to a log scale, given the skewness in the WBC values.

```
## Leukemia white blood cell types example
# ntotal = number of patients with IAG and WBC combination
# nres   = number surviving at least one year
# ag     = 1 for AG+, 0 for AG-
# wbc    = white cell blood count
# lwbc   = log white cell blood count
# p.hat  = Emperical Probability
leuk <- read.table("http://statacumen.com/teach/SC1/SC1_11_leuk.dat", header = TRUE)
leuk$lwbc  <- log(leuk$wbc)
leuk$p.hat <- leuk$nres / leuk$ntotal
```

	ntotal	nres	ag	wbc	lwbc	p.hat
1	1	1	1	75	4.32	1.00
2	1	1	1	230	5.44	1.00
3	1	1	1	260	5.56	1.00
4	1	1	1	430	6.06	1.00
5	1	1	1	700	6.55	1.00
6	1	1	1	940	6.85	1.00
7	1	1	1	1000	6.91	1.00
8	1	1	1	1050	6.96	1.00
9	3	1	1	10000	9.21	0.33
10	1	1	0	300	5.70	1.00
11	1	1	0	440	6.09	1.00
12	1	0	1	540	6.29	0.00
13	1	0	1	600	6.40	0.00
14	1	0	1	1700	7.44	0.00
15	1	0	1	3200	8.07	0.00
16	1	0	1	3500	8.16	0.00
17	1	0	1	5200	8.56	0.00
18	1	0	0	150	5.01	0.00
19	1	0	0	400	5.99	0.00
20	1	0	0	530	6.27	0.00
21	1	0	0	900	6.80	0.00
22	1	0	0	1000	6.91	0.00
23	1	0	0	1900	7.55	0.00
24	1	0	0	2100	7.65	0.00
25	1	0	0	2600	7.86	0.00
26	1	0	0	2700	7.90	0.00
27	1	0	0	2800	7.94	0.00
28	1	0	0	3100	8.04	0.00
29	1	0	0	7900	8.97	0.00
30	2	0	0	10000	9.21	0.00

As an initial step in the analysis, consider the following model:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \text{LWBC} + \beta_2 \text{AG},$$

where LWBC = log(WBC). The model is best understood by separating the AG+ and AG− cases. For AG− individuals, AG=0 so the model reduces to

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \text{LWBC} + \beta_2 * 0 = \beta_0 + \beta_1 \text{LWBC}.$$

For AG+ individuals, AG=1 and the model implies

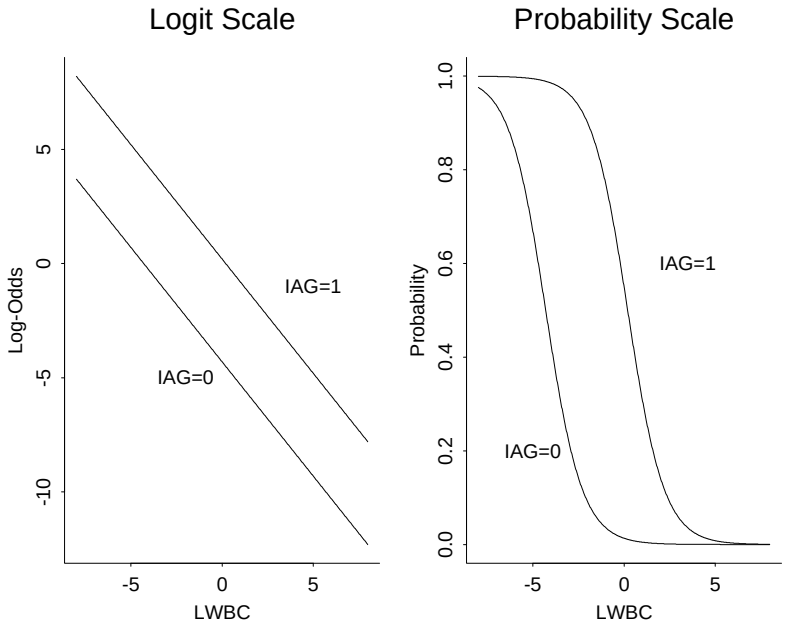
$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \text{LWBC} + \beta_2 * 1 = (\beta_0 + \beta_2) + \beta_1 \text{LWBC}.$$

The model without AG (i.e., $\beta_2 = 0$) is a simple logistic model where the log-odds of surviving one year is linearly related to LWBC, and is independent of AG. The reduced model with $\beta_2 = 0$ implies that there is no effect of the AG level on the survival probability once LWBC has been taken into account.

Including the **binary predictor** AG in the model implies that there is a linear relationship between the log-odds of surviving one year and LWBC, with a constant slope for the two AG levels. This model includes an effect for the AG morphological factor, but more general models are possible. A natural extension would be to include a product or interaction effect, a point that I will return to momentarily.

The parameters are easily interpreted: β_0 and $\beta_0+\beta_2$ are intercepts for the population logistic regression lines for AG− and AG+, respectively. The lines have a common slope, β_1. The β_2 coefficient for the AG indicator is the difference between intercepts for the AG+ and AG− regression lines.

A picture of the assumed relationship is given below for $\beta_1 < 0$. The population regression lines are parallel on the logit scale only, but the order between AG groups is preserved on the probability scale.



Before looking at output for the equal slopes model, note that the data set has 30 distinct AG and LWBC combinations, or 30 "groups" or samples. Only two samples have more than 1 observation. The majority of the observed proportions surviving at least one year (number surviving ≥ 1 year/group sample size) are 0 (i.e., 0/1) or 1 (i.e., 1/1). This sparseness of the data makes it difficult to graphically assess the suitability of the logistic model (because the estimated proportions are almost all 0 or 1).

Let's fit the model with our Fisher's Scoring method.

```
# create data variables: m, y, X
n <- nrow(leuk)
m <- leuk$ntotal
y <- leuk$nres
X <- matrix(c(rep(1,n), leuk$lwbc, leuk$ag), nrow = n)

colnames(X) <- c("Int", "lwbc", "ag")
```

```
r <- ncol(X) - 1  # number of regression coefficients - 1

# initial beta vector
beta.1 <- c(log(sum(y) / sum(m - y)), rep(0, r))

# fit betas using our Fisher Scoring function
out <- f.lr.FS(X, y, m, beta.1)
out


## $beta.MLE
##        [,1]
## Int   5.543
## lwbc -1.109
## ag    2.520
##
## $iter
## [1] 5
##
## $NR.hist
##   i diff.beta diff.like llike.1 step.size
## 1 1       Inf 1.000e+09  -21.00       1.0
## 2 2 6.081e+00 7.168e+00  -13.84       1.3
## 3 3 5.602e-01 4.164e-01  -13.42       1.2
## 4 4 1.814e-01 4.077e-03  -13.42       1.0
## 5 5 3.747e-03 1.267e-06  -13.42       1.0
## 6 6 1.368e-06 1.901e-13  -13.42       0.9
##
## $beta.hist
##         [,1]    [,2]  [,3]
## [1,] -0.6931  0.0000 0.000
## [2,]  4.9039 -0.9312 2.188
## [3,]  5.3702 -1.0819 2.460
## [4,]  5.5399 -1.1082 2.518
## [5,]  5.5433 -1.1088 2.520
## [6,]  5.5433 -1.1088 2.520
##
## $beta.cov
##          Int    lwbc      ag
## Int   9.1350 -1.3400  0.4507
## lwbc -1.3400  0.2125 -0.1798
## ag    0.4507 -0.1798  1.1896
```

Looking at the output we see that the routine converged in 5 iterations.

At each step, the log-likelihood increased, and the norm of the difference between successive estimates eventually decreased to zero. The estimates are 5.543 for the constant term, −1.109 for the linear term, and 2.52 for the quadratic term.

```
# create a parameter estimate table
beta.Est  <- out$beta.MLE
beta.SE   <- sqrt(diag(out$beta.cov)) # sqrt diag inverse Information matrix
beta.z    <- beta.Est / beta.SE
beta.pval <- 2 * pnorm(-abs(beta.z))

beta.coef <- data.frame(beta.Est, beta.SE, beta.z, beta.pval)
beta.coef
```

```
##      beta.Est beta.SE beta.z beta.pval
## Int     5.543  3.0224  1.834   0.06664
## lwbc   -1.109  0.4609 -2.405   0.01616
## ag      2.520  1.0907  2.310   0.02088
```

Compare our parameter estimate table above to the one from the `glm()` function.

```
## compare to the glm() fit:
summary(glm.i.l)$call
```

```
## glm(formula = cbind(nres, ntotal - nres) ~ ag + lwbc, family = binomial,
##     data = leuk)
```

```
summary(glm.i.l)$coefficients
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    5.543     3.0224   1.834  0.06664
## ag1            2.520     1.0907   2.310  0.02088
## lwbc          -1.109     0.4609  -2.405  0.01615
```

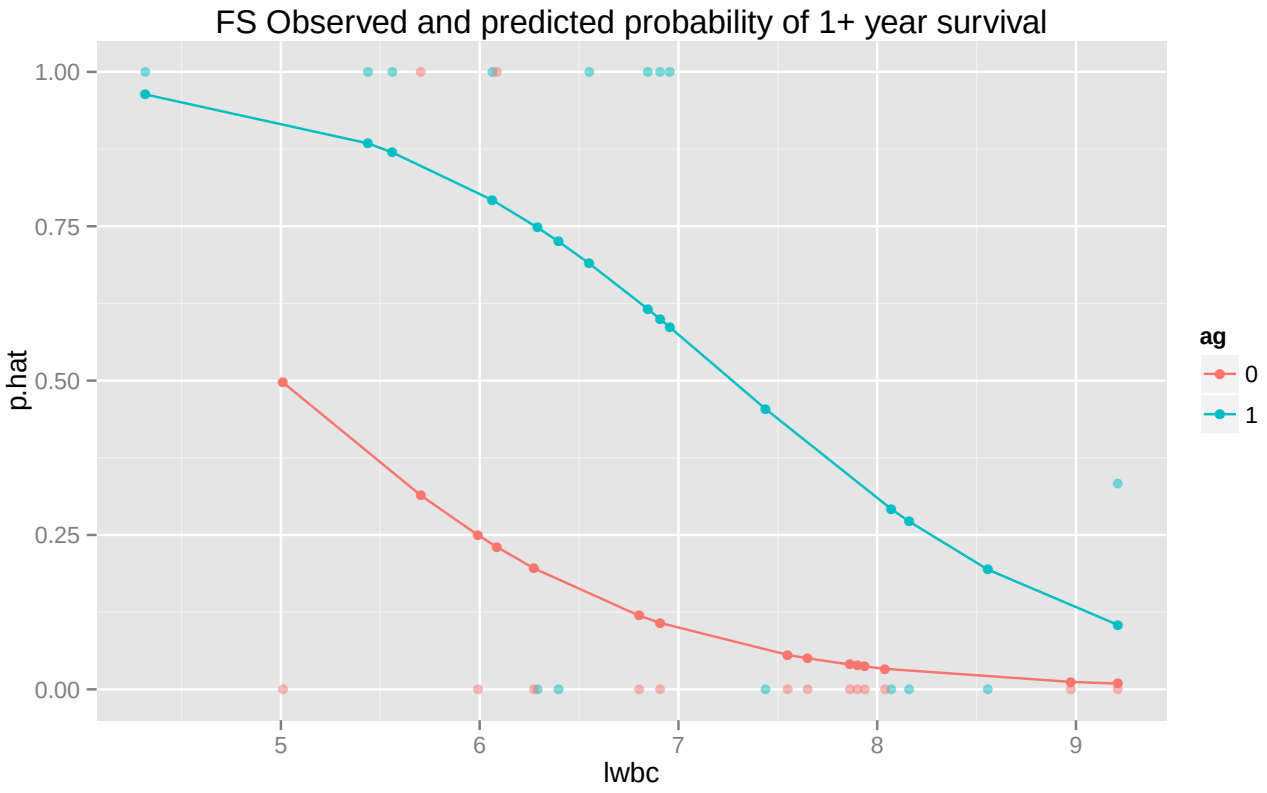
Given that the model fits reasonably well, a test of $H_0 : \beta_2 = 0$ might be a primary interest here. This checks whether the regression lines are identical for the two AG levels, which is a test for whether AG affects the

survival probability, after taking LWBC into account. This test is rejected at any of the usual significance levels, suggesting that the AG level affects the survival probability (assuming a very specific model).

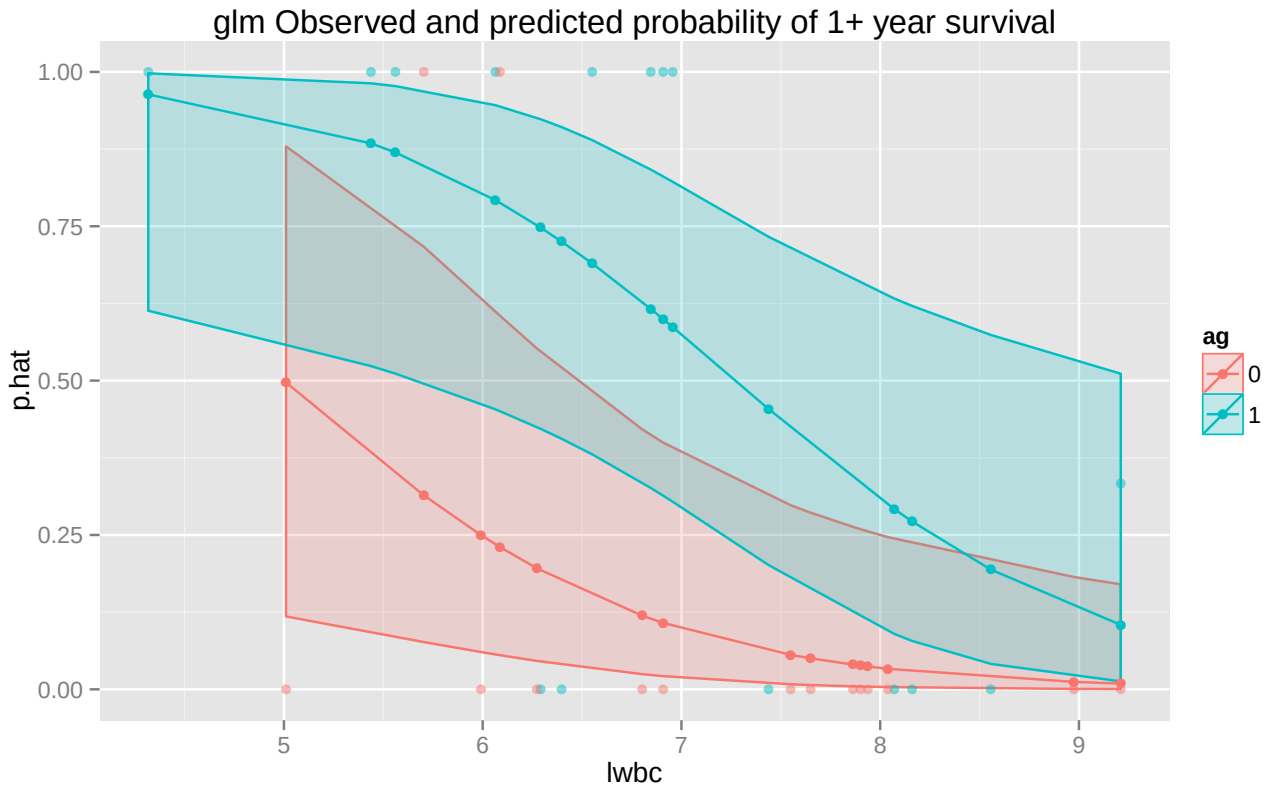
A plot of the predicted survival probabilities as a function of LWBC, using AG as the plotting symbol, indicates that the probability of surviving at least one year from the time of diagnosis is a decreasing function of LWBC. For a given LWBC the survival probability is greater for AG+ patients than for AG− patients. This tendency is consistent with the observed proportions, which show little information about the exact form of the trend.

```
# plot observed and predicted proportions
# leuk$p.hat calculated earlier
leuk$p.MLE <- f.lr.p(X, out$beta.MLE)  #$

library(ggplot2)
p <- ggplot(leuk, aes(x = lwbc, y = p.hat, colour = ag))
p <- p + geom_line(aes(y = p.MLE))
# fitted values
p <- p + geom_point(aes(y = p.MLE), size=2)
# observed values
p <- p + geom_point(size = 2, alpha = 0.5)
p <- p + labs(title = "FS Observed and predicted probability of 1+ year survival")
print(p)
```



The plot from our Fisher's Scoring method above is the same as the plot below from the glm() procedure.



To complete this example, the estimated survival probabilities satisfy

$$\log\left(\frac{\tilde{p}}{1-\tilde{p}}\right) = 5.54 - 1.11 \text{ LWBC} + 2.52 \text{ AG}.$$

For AG− individuals with AG=0, this reduces to

$$\log\left(\frac{\tilde{p}}{1-\tilde{p}}\right) = 5.54 - 1.11 \text{ LWBC},$$

or equivalently,

$$\tilde{p} = \frac{\exp(5.54 - 1.11 \text{ LWBC})}{1 + \exp(5.54 - 1.11 \text{ LWBC})}.$$

For AG+ individuals with AG=1,

$$\log\left(\frac{\tilde{p}}{1-\tilde{p}}\right) = 5.54 - 1.11 \text{ LWBC} + 2.52(1) = 8.06 - 1.11 \text{ LWBC},$$

or

$$\tilde{p} = \frac{\exp(8.06 - 1.11 \text{ LWBC})}{1 + \exp(8.06 - 1.11 \text{ LWBC})}.$$

Although the equal slopes model appears to fit well, a more general model might fit better. A natural generalization here would be to add an **interaction**, or product term, AG $*$ LWBC to the model. The logistic model with an AG effect and the AG $*$ LWBC interaction is equivalent to fitting separate logistic regression lines to the two AG groups. This interaction model provides an easy way to test whether the slopes are equal across AG levels. I will note that the interaction term is not needed here.