Binary Regression

Say have BINARY (0/1) response variable Y and predictors $X_1, \ldots, X_p$.

Can still fit a linear model, but does

$$E(Y|X_1, \ldots, X_p) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

still make sense.

Transform one side?

Logistic regression model based on “logit” function:
$logit(p) = log(p/(1-p))$

Transform LHS of LM expression to get:

$$\text{logit}\left\{Pr(Y=1|X_1,\ldots,X_p)\right\} \quad = \quad \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

Note that this a fully-specified model.

Mismatch problem solved. Other properties?

With LM, β_j interpretable via change in $E(Y|X)$ *wrt* X_j, when $X_{-j} = (X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_p)$ held fixed.

Corresponding story for logistic regression?

Say X_j is also binary (0/1). A little algebra yields

$$\frac{Odds(Y=1|X_j=1,X_{-j})}{Odds(Y=1|X_j=0,X_{-j})} \quad = \quad \exp(\beta_j),$$

provided there are no interaction terms involving X_j in the model.

Or, for continuous X_j:

$$\frac{Odds(Y=1|X_j=a+\Delta,X_{-j})}{Odds(Y=1|X_j=a,X_{-j})} \quad = \quad \exp(\beta_j\Delta).$$

This is the basis for *interpreting* the parameters in a logistic regression, and the reason that the logit function, rather than some other function, is the most popular choice for setting up a binary regression model.

Pushing the interpretation a bit further...

Sometimes (in health research particularly), $Y = 1$ is a 'rare' outcome.

Mathematically, if p, q close to zero,

$$\frac{p/(1-p)}{q(1-q)} \approx p/q$$

That is, odds ratio approximates *relative risk*.

Hence an approximate interpretation for logistic regression is

$$\frac{Pr(Y = 1|X_j = 1, X_{-j})}{Pr(Y = 1|X_j = 0, X_{-j})} \quad \approx \quad \exp(\beta_j)$$

A cornerstone of Epidemiology!

Mechanics of fitting a logistic regression model?

As simple as $\hat{\beta} = (X^T X)^{-1} X^T Y$ for LM?

Not quite. No closed-form expression for the value of β maximizing the logistic regression log-likelihood.

Interestingly though, if we apply the iterative **Newton-Raphson** algorithm to maximize the log-likelihood function, at each iteration we end up doing a **weighted least-squares** fit with a **psuedo-response** vector (more on this later).

So the computation is not bad. The logistic regression analogue to lm(y∼x) is glm(y∼x, family=binomial).

One thing that's easier with logistic regression than LM - **Goodness-of-fit** test.

Say you have chosen some variables to include as predictors in a logistic regression model. You would like to test whether this model is appropriate for the data, **without explicitly constructing alternative models**.

Consider the *saturated* model having one parameter for every datapoint, i.e., $p_i = Pr(Y_i = 1)$. Compare maximized log-likelihoods for the saturated model and your model:

$$\text{Deviance} \quad = \quad 2\left\{\max_p l(p_1, \ldots, p_n; y, x) - \max_\beta l(\beta_1, \ldots, \beta_p; y, x)\right\}.$$

If you generate data from your model, then, approximately, Deviance $\sim \chi^2_{n-p}$

So an observed deviance in the right-tail of the χ^2_{n-p} distribution suggests a poor-fitting model, (i.e., can report a P-value).

Aside: this is *not* usual LR test asymptotics. Slightly more delicate/iffy when the number of parameters grows with sample size.

Other aside: why can't this be done with linear models?

Linear models involve unknown σ^2 describing magnitude of error term, as estimated by RSS. Can test a smaller model versus a bigger model by looking at a *relative* reduction in RSS (per number of new parameters introduced). But there is no 'magic' *absolute* size of RSS that is expected when the fitted model is correct.

One thing that's harder with logistic regression than LM - **Graphical Diagnostics**

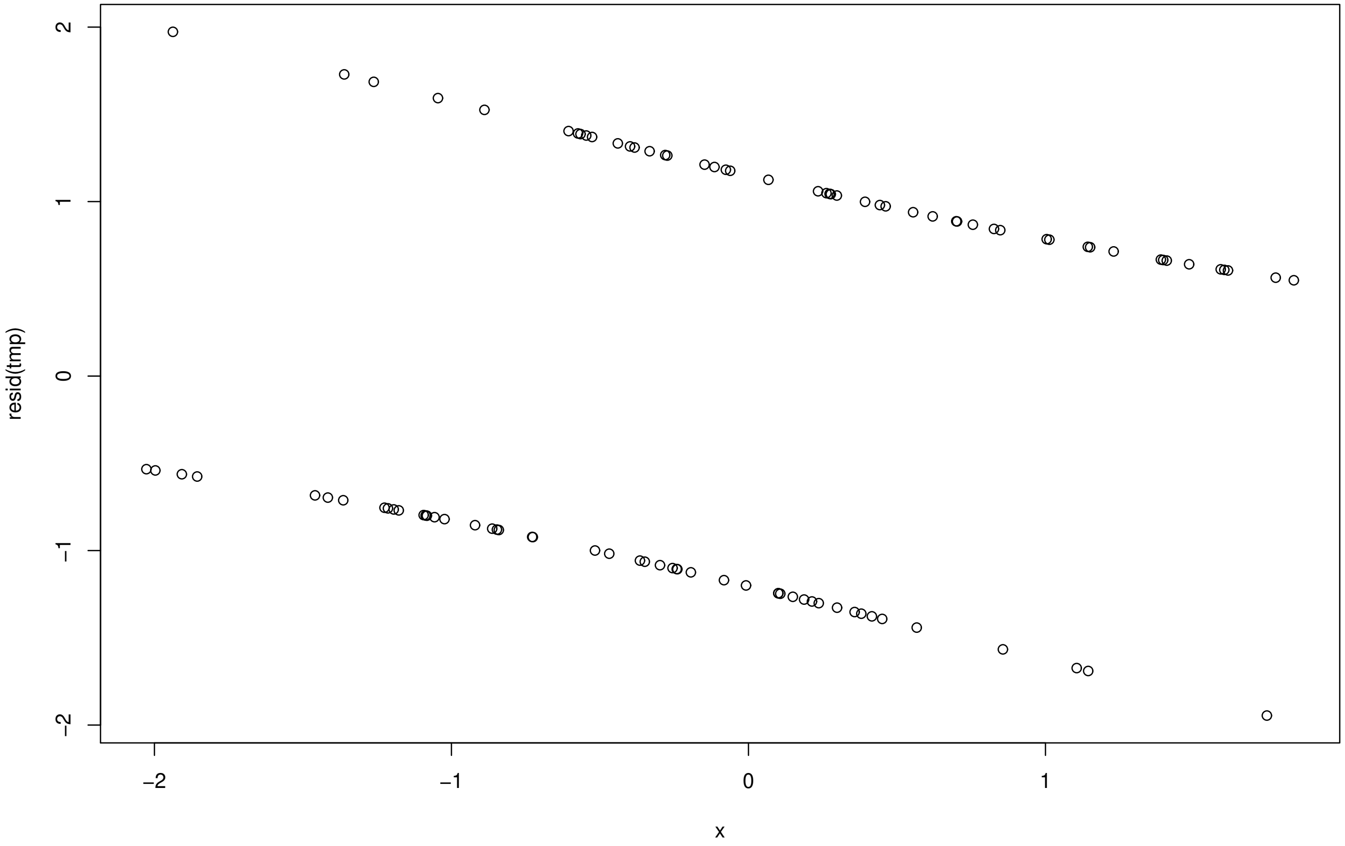
There are schemes for defining **residuals** for logistic regression (and other GLMs).

For instance, *deviance residuals* are defined by analogy. Deviance for logistic regression plays role of RSS for LM. In particular, deviance is a sum of n terms for the n datapoints. So define residuals as signed square-roots of these terms.

```
> x <- rnorm(100)
> y <- rbinom(100, size=1, prob=1/(1+exp(-(0+.7*x))))
> tmp <- glm(y~x, family=binomial)
```

```
> summary(tmp)
Call: glm(formula = y ~ x, family = binomial)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.94557  -1.05909   0.00772   1.03679   1.97291

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.06058    0.21836   0.277 0.781451
x            0.95606    0.26012   3.675 0.000237 ***

    Null deviance: 138.63  on 99  degrees of freedom
Residual deviance: 122.06  on 98  degrees of freedom

Number of Fisher Scoring iterations: 4
```

```
> plot(x, resid(tmp))
```



Useful?