Generalized linear models

Outline for today

- What is a generalized linear model
- Linear predictors and link functions
- Example: estimate a proportion
- Analysis of deviance
- Example: fit dose-response data using logistic regression
- Example: fit count data using a log-linear model
- Advantages and assumptions of glm
- Quasi-likelihood models when there is excessive variance

Review: what is a (general) linear model

A model of the following form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \text{error}$$

- *Y* is the response variable
- The *X* 's are the explanatory variables
- The β 's are the parameters of the linear equation
- The errors are normally distributed with equal variance at all values of the *X* variables.
- Uses least squares to fit model to data, estimate parameters
- `lm` in R

The predicted *Y*, symbolized here by μ, is modeled as

$$\mu = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots$$

What is a <u>generalized</u> linear model

A model whose predicted values are of the form

$$g(\mu) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots$$

- The model still include a linear predictor (to right of "=")
- $g(\mu)$ is the "link function"
- Wide diversity of link functions accommodated
- Non-normal distributions of errors OK (specified by link function)
- Unequal error variances OK (specified by link function)
- Uses maximum likelihood to estimate parameters
- Uses log-likelihood ratio tests to test parameters
- `glm` in R

The two most common link functions

Log

- used for count data

$\eta = \log \mu$ The inverse function is $\mu = e^{\eta}$

Logistic or **logit**

- used for binary data

$\eta = \log \frac{\mu}{1-\mu}$ The inverse function is $\mu = \frac{e^{\eta}}{1+e^{\eta}}$

In all cases log refers to natural logarithm (base *e*).

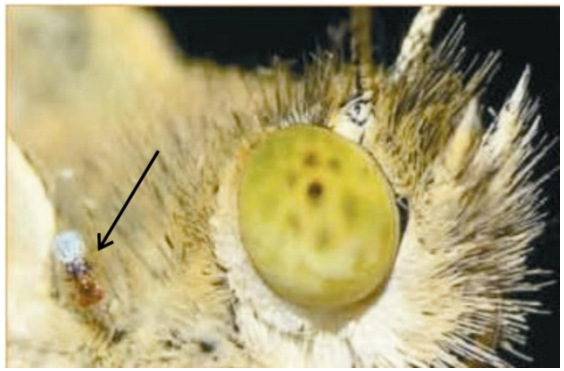
Example 1: Estimate a proportion

Example used previously in Likelihood lecture:

The wasp, *Trichogramma brassicae*, rides on female cabbage white butterflies, *Pieris brassicae*. When a butterfly lays her eggs on a cabbage, the wasp climbs down and parasitizes the freshly laid eggs.

Fatouros et al. (2005) carried out trials to determine whether the wasps can distinguish mated female butterflies from unmated females. In each trial a single wasp was presented with two female cabbage white butterflies, one a virgin female, the other recently mated.

Y = 23 of 32 wasps tested chose the mated female. What is the proportion p of wasps in the population choosing the mated female?



Number of wasps choosing mated female fits a binomial distribution

Under random sampling, the number of "successes" in *n* trials has a binomial distribution, with *p* being the probability of "success" in any one trial.

To model these data, let "success" be "wasp chose mated butterfly"

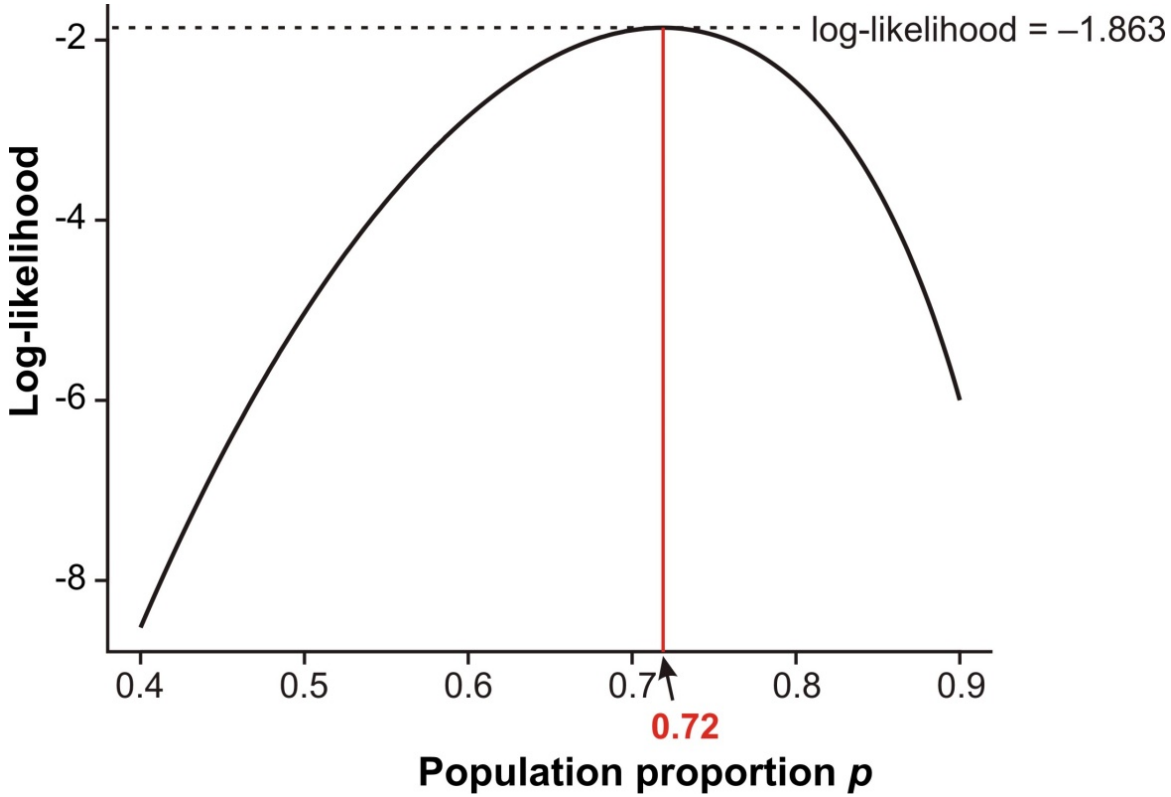
Y = 23 successes
n = 32 trials

What is *p*?

Previously we used maximum likelihood to estimate *p*

The *maximum likelihood estimate* is that value for the parameter having the highest (log)likelihood, given the data.

$$\ln L[p \mid 23 \text{ choose mated}] = \ln\left[\left|\binom{32}{23}\right|\right] + 23\ln[p] + 9\ln[1-p]$$



Consider first how we could use `lm` to estimate a population mean

Estimate a mean of a variable using a random sample of the population.

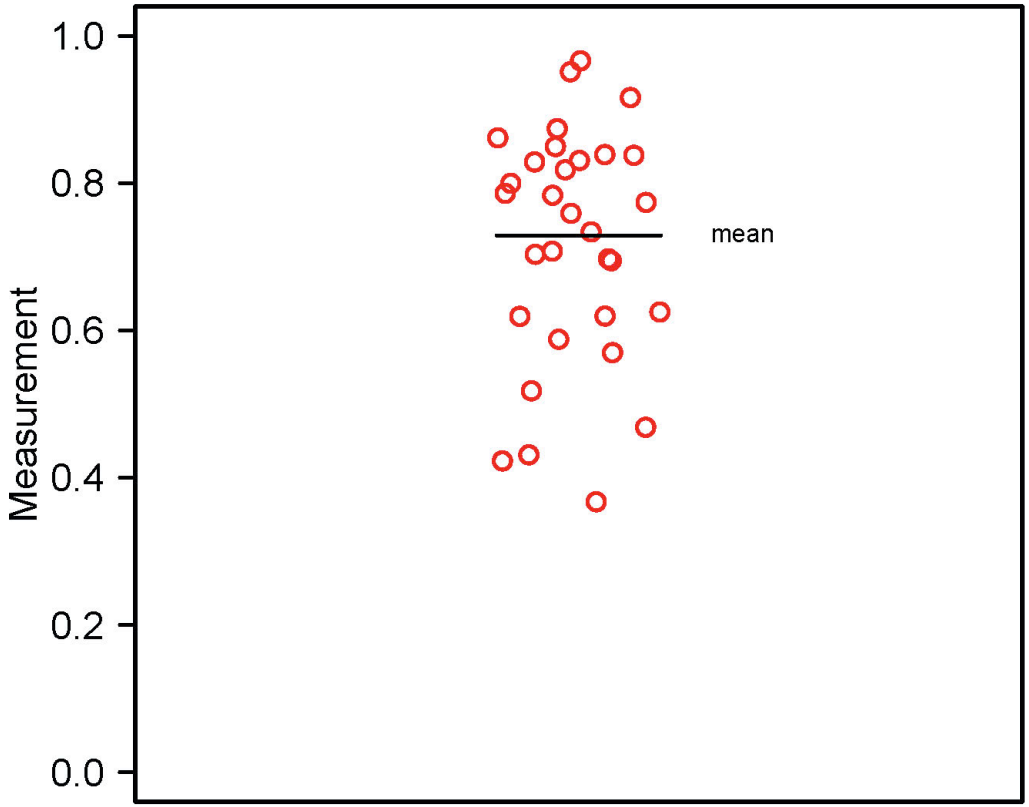
If the data are approximately normally distributed, we could fit a linear model,

$$\mu = \beta$$

where μ is the mean in the population.

In R, use `lm` to estimate,

```
z <- lm(y ~ 1)

summary(z)
```



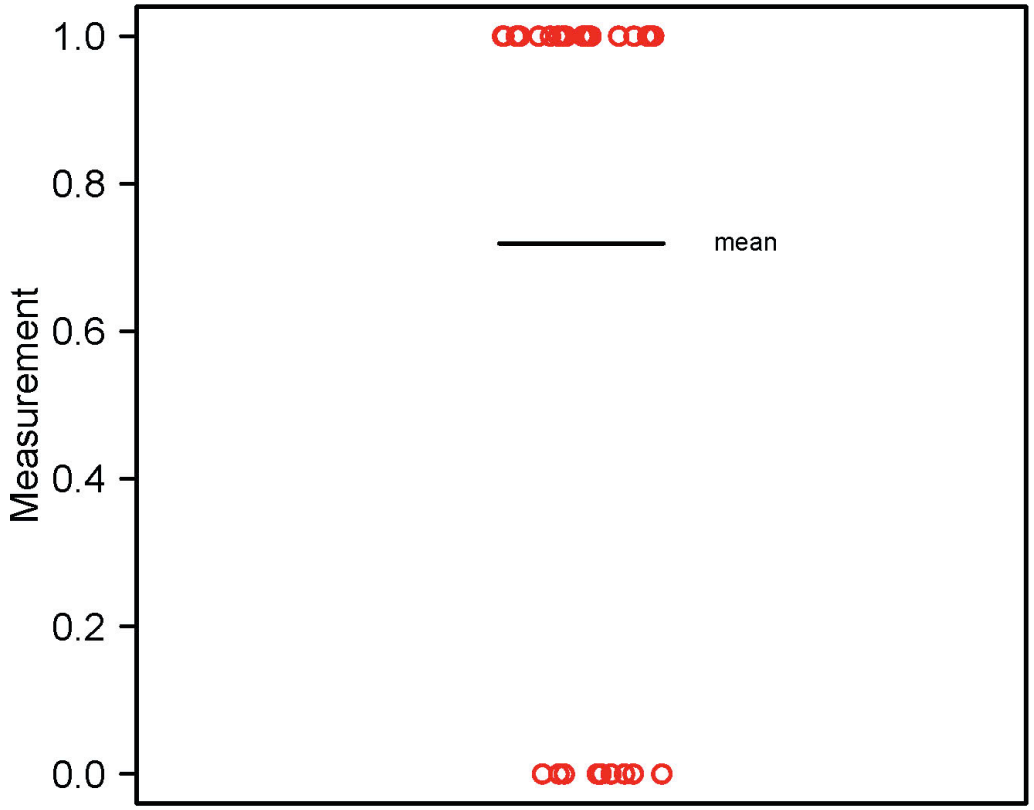
A population proportion is also a mean

Each wasp has a measurement of 1 or 0 (for "number of successes").
The mean of these 0's and 1's is the proportion we wish to estimate.

But the data are not normally-distributed: they are binary

Instead of a linear model, we must use a <u>generalized linear model</u> with a link function appropriate for binary data.

$$g(\mu) = \beta$$



Use `glm` to estimate the proportion

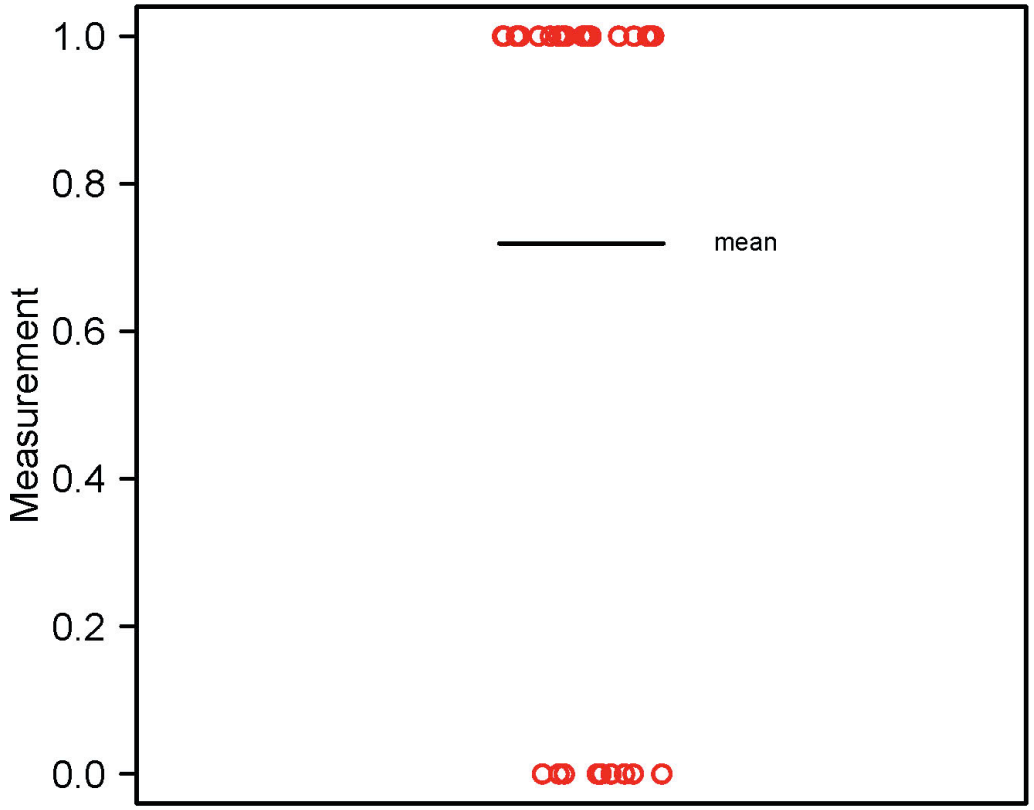
We need a link function appropriate for binary data.

$$\text{logit}(\mu) = \beta$$

or equivalently,

$$\log \frac{\mu}{1-\mu} = \beta$$

where μ is the population proportion
(i.e., *p*, but let's stick with μ here to be consistent with glm notation)

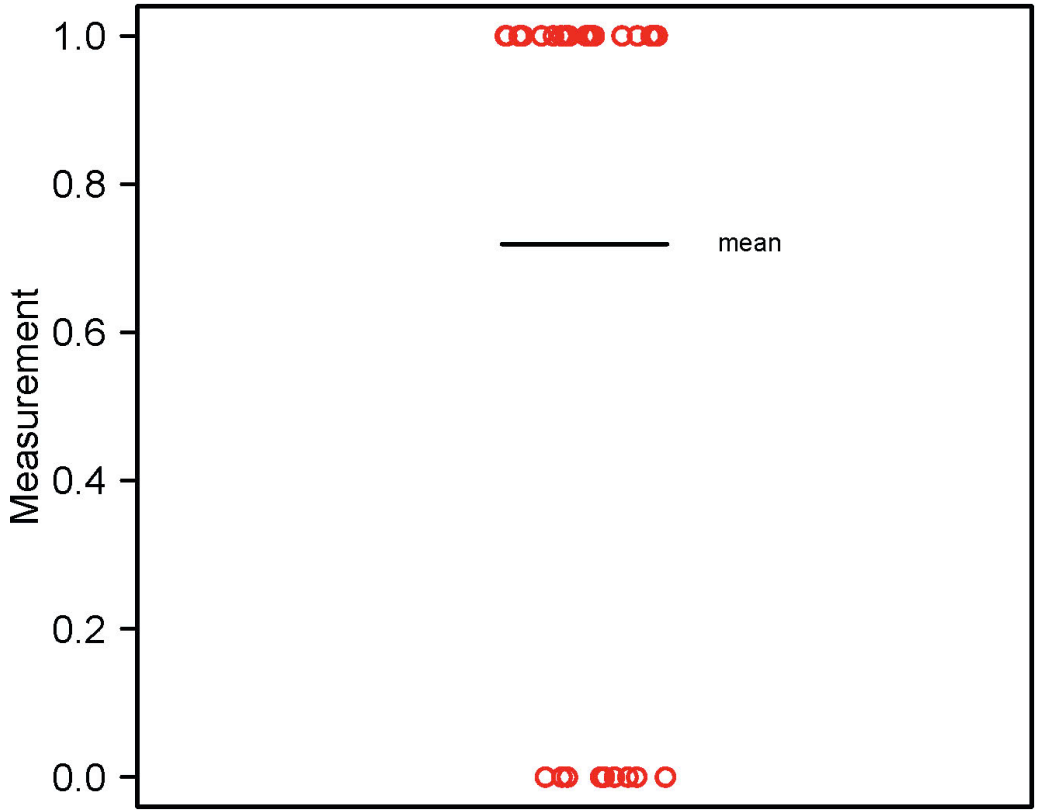


Use `glm` to estimate a proportion

```
z <- glm(y ~ 1, family = binomial(link="logit"))
```

Formula structure is the same as it would be using `lm`.

Here, `family` specifies the error distribution and link function.



Use `glm` to estimate a proportion

```
summary(z)

 Coefficients:
             Estimate Std. Error z value Pr(>|z|)
 (Intercept)   0.9383     0.3932   2.386    0.017 *
```

0.9383 is the estimate of β (the mean on the logit scale)
Convert back to ordinary scale (plug into inverse equation) to get estimate of population proportion

$$\hat{\mu} = \frac{e^{\hat{\beta}}}{1+e^{\hat{\beta}}} = \frac{e^{0.9383}}{1+e^{0.9383}} = 0.719$$

This is the ML estimate of the population proportion.

[In the workshop we'll obtain likelihood-based confidence intervals too.]

Use `glm` to test a null hypothesis about a proportion

```
summary(z)

 Coefficients:
             Estimate Std. Error z value Pr(>|z|)
 (Intercept)   0.9383     0.3932   2.386    0.017 *
```

The *z*-value (a.k.a Wald statistic) and *P*-value test the null hypothesis that $\beta = \beta_0 = 0$. This is the same as a test of the null hypothesis that the true (population) proportion is 0.5, because

$$\frac{e^{\beta_0}}{1+e^{\beta_0}} = \frac{e^{0}}{1+e^{0}} = 0.5$$

Agresti (2002, *Categorical data analysis*, 2nd ed., Wiley) says that for small to moderate sample size, the Wald test is usually less reliable than the log-likelihood ratio test. So I have crossed it out!

Use `glm` to test a null hypothesis about a proportion

I calculated the log-likelihood ratio test for these data in the Likelihood lecture. Here we'll use glm to accomplish the same task.

"Full" model (β is estimated from data):

```
z <- glm(y ~ 1, family = binomial(link="logit"))
```

"Reduced" model ($\beta = 0$):

```
z0 <- glm(y ~ -1, family = binomial(link="logit"))
```

Use `glm` to test a null hypothesis about a proportion

```
anova(z0, z, test  = "Chi") # Analysis of deviance

 Model 1: y ~ -1    # Reduced model
 Model 2: y ~ 1     # Full model

   Resid. Df Resid. Dev Df Deviance P(>|Chi|)
 1        32     44.361
 2        31     38.024  1    6.337   0.01182 *
```

The deviance is the log-likelihood ratio statistic (*G*-statistic),
it has an approximate χ^2 distribution under the null hypothesis.

Looking back to the Likelihood lecture, we see that the result is the same:

$$L[0.72 \mid 23 \text{ chose mated female}] = 0.1553$$
$$L[0.50 \mid 23 \text{ chose mated female}] = 0.00653$$

G = 2ln(0.1553 / 0.00653) = 6.336

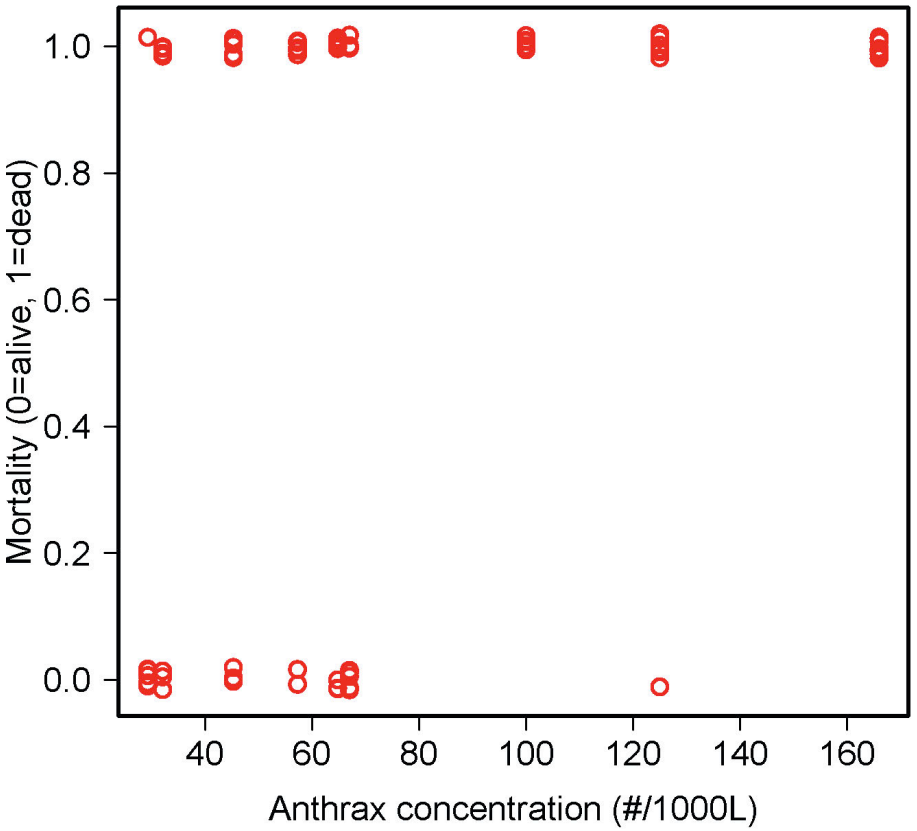
This is because glm is based on maximum likelihood.

[break]

Example 2: Logistic regression

One of the most common uses of generalized linear models

Data: 72 rhesus monkeys (*Macacus rhesus*) exposed for 1 minute to aerosolized preparations of anthrax (*Bacillus anthracis*). Want to estimate the relationship between dose and probability of death.

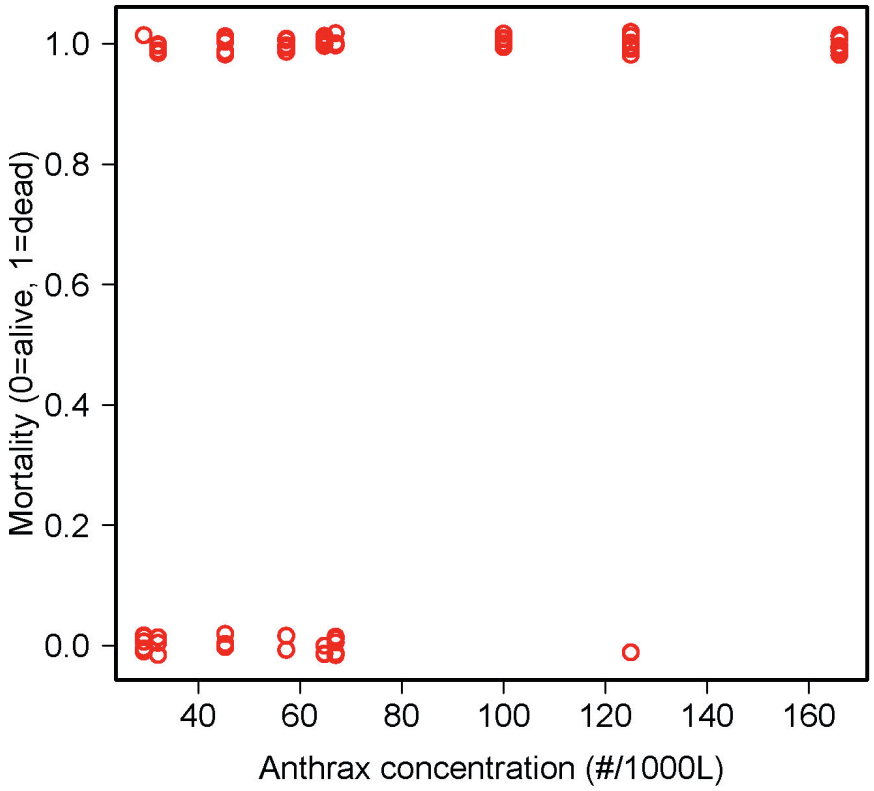


Logistic regression

Measurements of individuals are 1 (dead) or 0 (alive)

Linear regression model not appropriate because

- For each *X* the *Y* observations are binary, not normal
- For every *X* the variance of *Y* is not constant
- A linear relationship is not bounded between 0 and 1
- 0-1 data can't simply be transformed



The generalized linear model

$$g(\mu) = \beta_0 + \beta_1 X$$

μ is the probability of death, which depends on concentration *X*.

$g(\mu)$ is the link function.

Linear predictor (right side of equation) is like an ordinary linear regression, with intercept β_0 and slope β_1

Logistic regression uses the logit link function

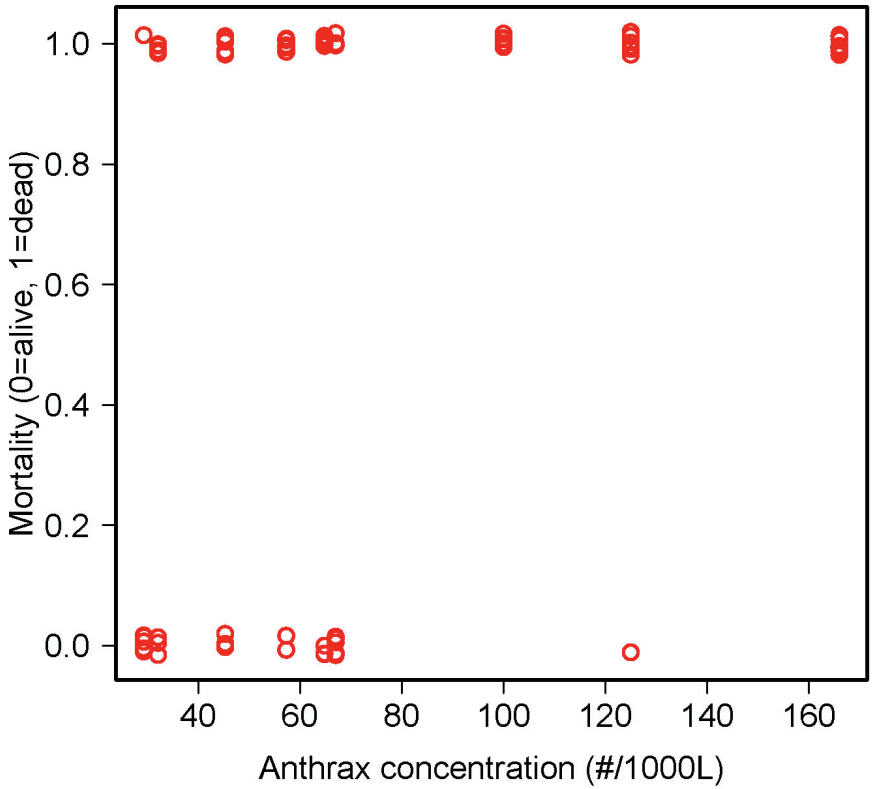
```
z <- glm(mortality ~ concentration,
                      family = binomial(link = "logit"))
```

The generalized linear model

$$\text{logit}(\mu) = \beta_0 + \beta_1 X$$

glm uses maximum likelihood: the method finds those values of β_0 and β_1 for which the data have maximum probability of occurring. These are the maximum likelihood estimates.

No formula for the solution. glm uses an iterative procedure to find the maximum likelihood estimates.



The generalized linear model

```
z <- glm(mortality ~ concentration,
                        family = binomial(link = "logit"))

summary(z)
                Estimate Std. Error z value Pr(>|z|)
 (Intercept)    -1.74452    0.69206  -2.521  0.01171 *
 concentration   0.03643    0.01119   3.255  0.00113 **

 Number of Fisher Scoring iterations: 5
```

Numbers in red are the estimates of β_0 and β_1 (intercept and slope) on the logit scale.

`Number of Fisher Scoring iterations` in the output refers to the number of iterations used before the algorithm used by glm converged on the maximum likelihood solution.

The generalized linear model

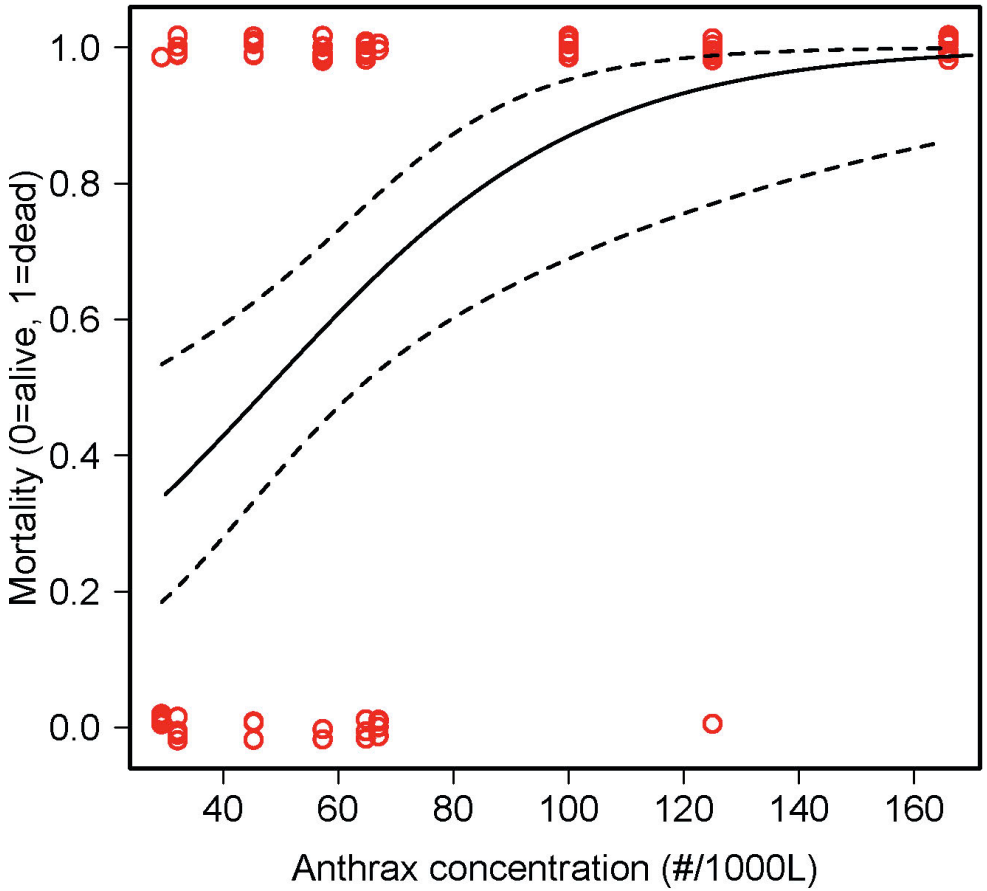
Use `predict(z)` to obtain predicted values on the logit scale

$$\hat{\eta} = -1.7445 + 0.03643X$$

Use `fitted(z)` to obtain predicted values on the original scale

$$\hat{\mu} = \frac{e^{\hat{\eta}}}{1 + e^{\hat{\eta}}}$$

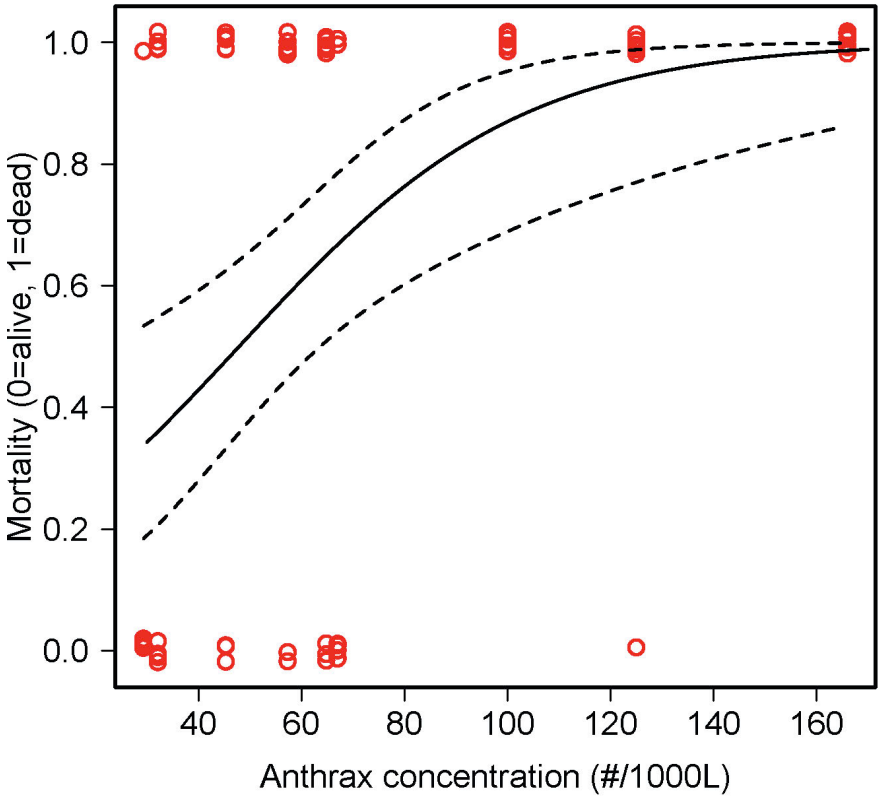
Can calculate (very approximate) confidence bands as in ordinary regression



The generalized linear model

The parameter estimates from the model fit can be used to estimate LD50, the concentration at which 50% of individuals are expected to die.

$$\begin{aligned} LD50 &= -\frac{\text{intercept}}{\text{slope}} \\ &= -\frac{(0.03643)}{(-1.7445)} \\ &= 47.88 \end{aligned}$$




```
library(MASS)
dose.p(z, p=0.50)
             Dose       SE
p = 0.5: 47.8805 8.168823
```

Residual plots in `glm`

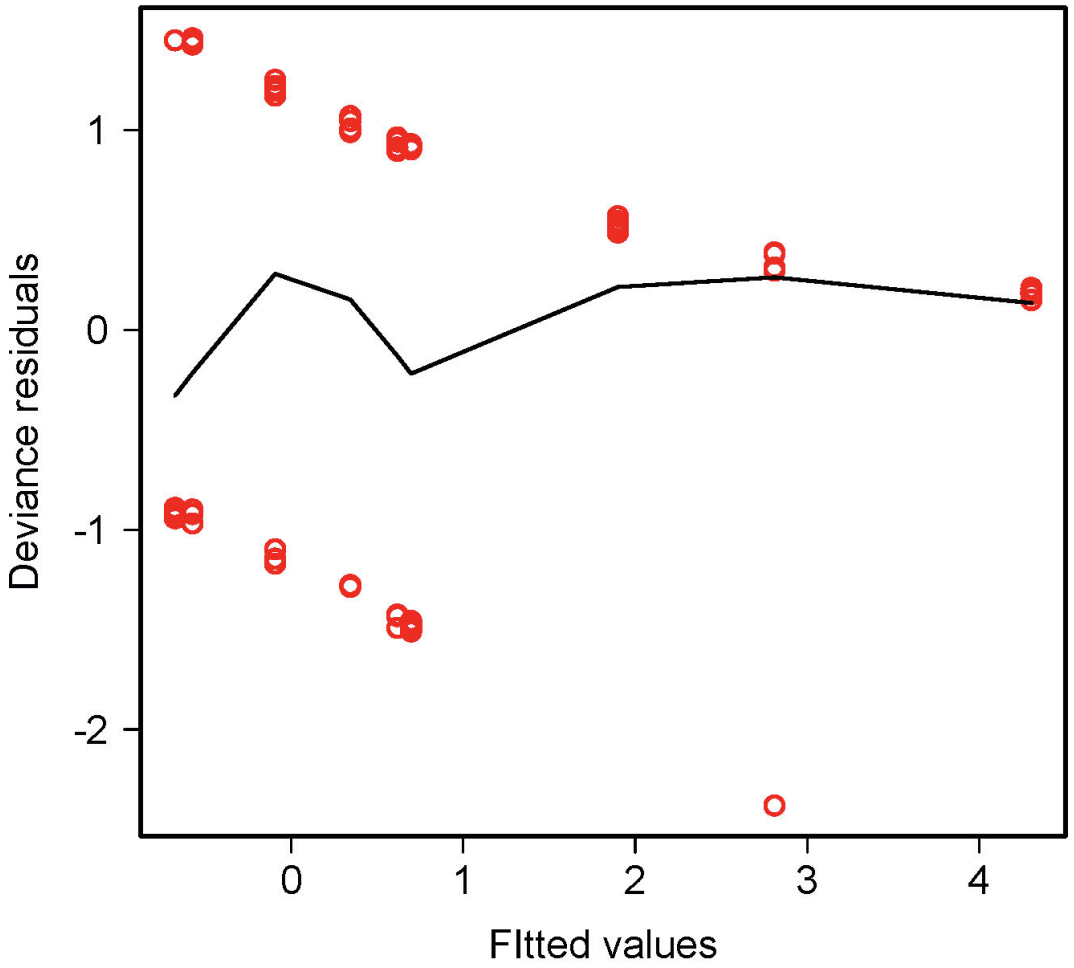
Residuals are calculated on the transformed scale (here, the logit).

1. Deviance residuals.

`plot(z)` or `plot(fitted(z), residuals(z))`

Discrete nature of the data introduces “stripes”.

These aren’t the real residuals of the fitted glm model. Deviance residuals identify contributions of each point to total deviance, which measures overall fit of the model to data.



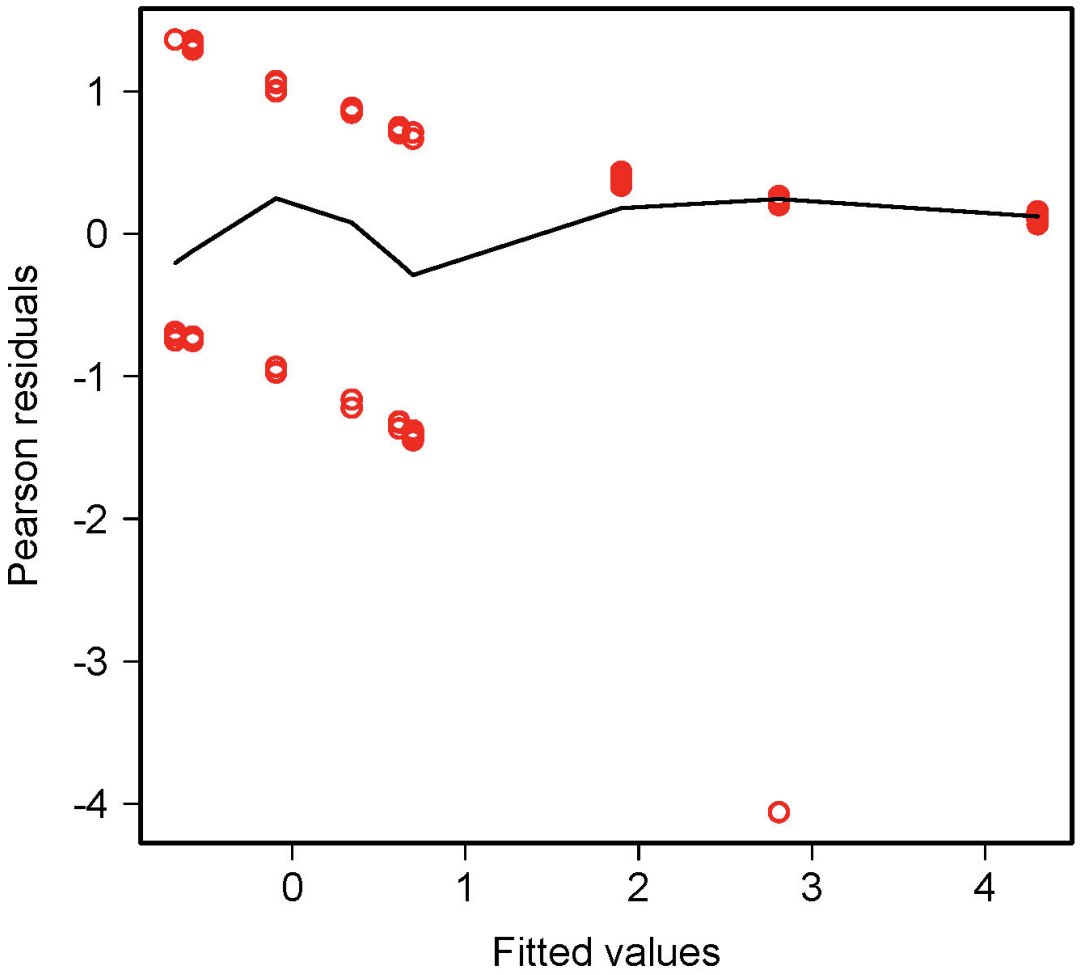
Residual plots in `glm`

2. Pearson residuals.

```
plot(fitted(z), residuals(z, type="pearson"))
```

These aren't the real residuals either.

They are a rescaled version of the real "working" residuals, corrected for their associated weights.



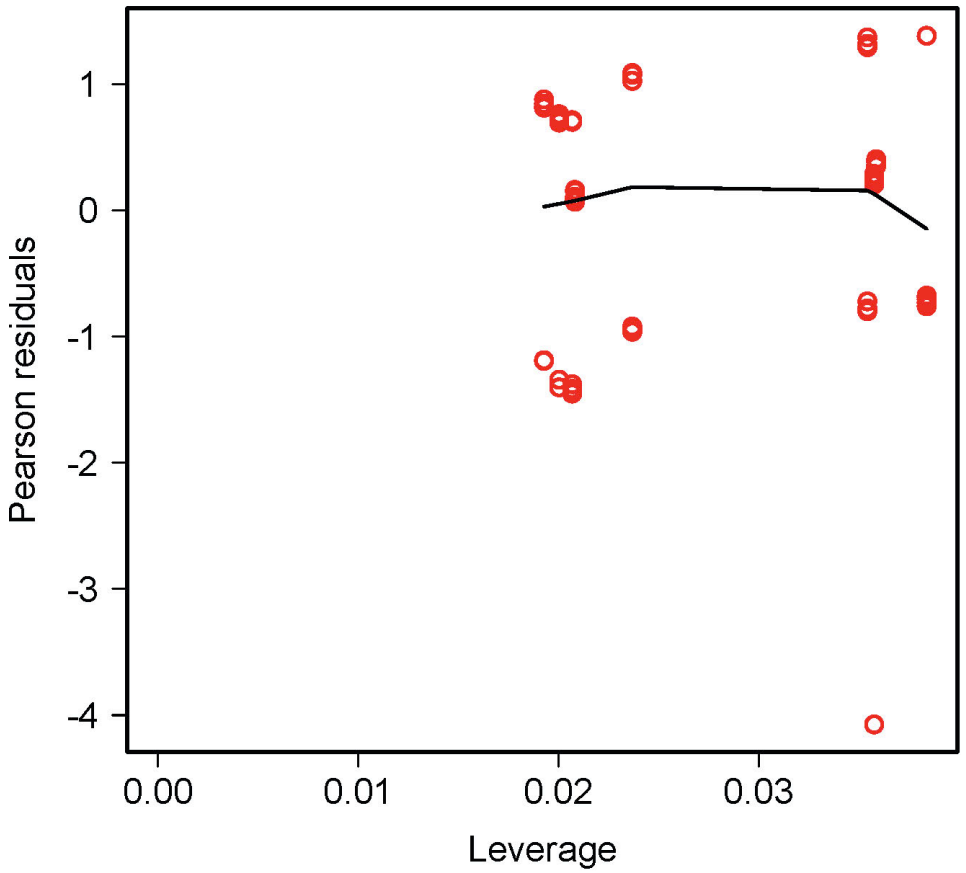
Leverage plot in `glm`

"Leverage" estimates the effect that each point has on the model parameter estimates. Obtained as `hatvalues(z)`

Leverage can range between 1/n (here, 0.013) and 1.

The graph here shows that even though one of the data points has a large residual, it does not have large leverage.

`plot(z)`



Advantages of generalized linear models

- More flexible than simply transforming variables. (A given transformation of the raw data may not accomplish both linearity and homogeneity of variance.)
- Yields more familiar measures of the response variable than data transformations. (E.g., how to interpret arcsine square root).
- Avoids the problems associated with transforming 0's and 1's. (For example, log(0), logit(0) and logit(1) can't be computed.)
- Retains the same analysis framework as linear models.

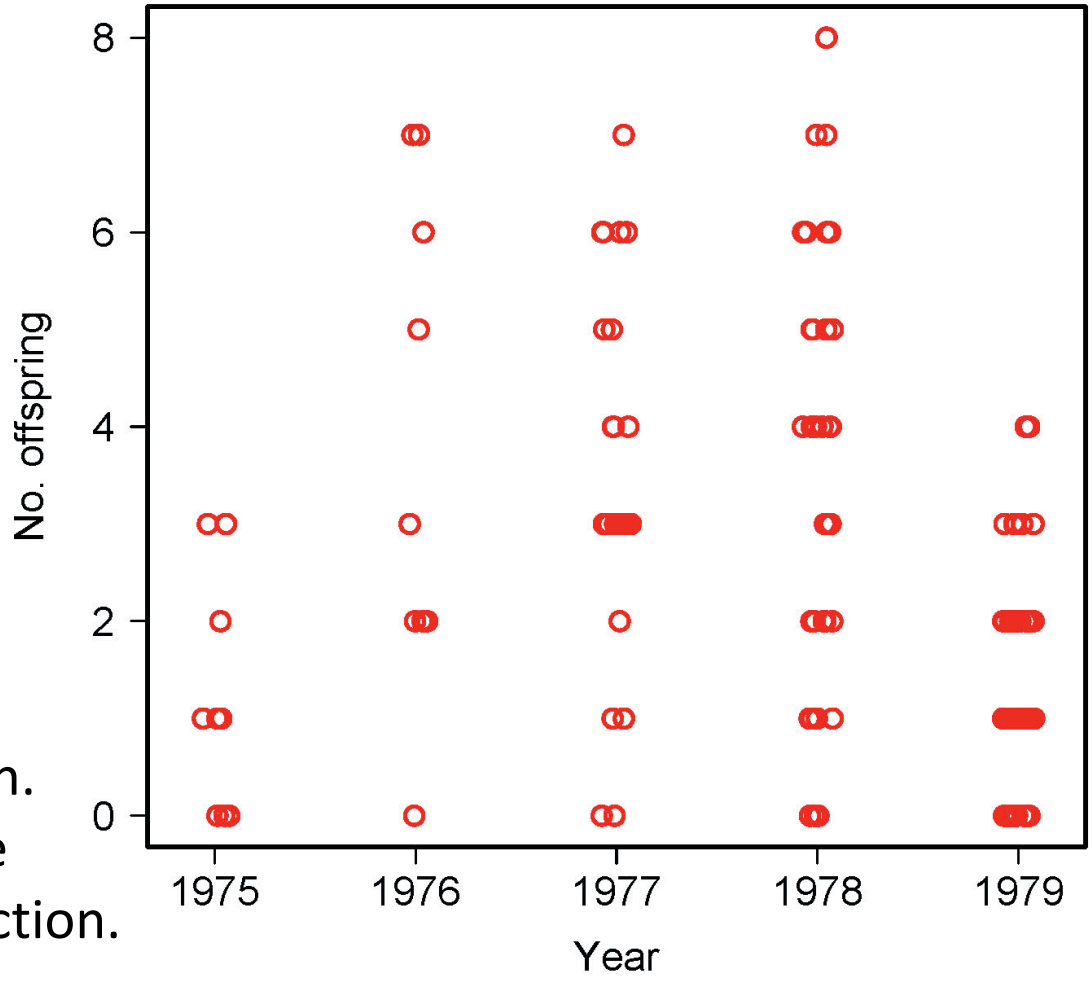
Assumptions of generalized linear models

- Statistical independence of data points.
- Correct specification of the link function for the data.
- The variances of the residuals correspond to that expected from the link function.
- Later, I will show a method for dealing with excessive variance.

Example 3: Analyzing count data with log-linear regression

Number of offspring fledged by female song sparrows on Mandarte Island, BC.

http://commons.wikimedia.org/wiki/File:Song_Sparrow-27527-2.jpg



Data are discrete counts.

Variance increases with mean.

Poisson distribution might be appropriate. Use log link function.

The generalized linear model

Log-linear regression (a.k.a. Poisson regression) uses the log link function

$$\log(\mu) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots$$

μ is the mean number of offspring, which depends on year

Year is a categorical variable (a factor in R).

Linear predictor (right side of equation) is like an ordinary ANOVA (modeled in R using "dummy" variables, same as with `lm`)

The generalized linear model

```
z <- glm(noffspring ~ year, family = poisson(link = "log"))

summary(z)
             Estimate Std. Error z value Pr(>|z|)
 (Intercept)  0.24116    0.26726   0.902 0.366872
 year1976     1.03977    0.31497   3.301 0.000963 ***
 year1977     0.96665    0.28796   3.357 0.000788 ***
 year1978     0.97700    0.28013   3.488 0.000487 ***
 year1979    -0.03572    0.29277  -0.122 0.902898

(Dispersion parameter for poisson family taken to be 1)
```

Numbers in red are the parameter estimates on the log scale.
As when analyzing a single categorical variable, the intercept refers to the mean of the first group (1975) and the rest of the coefficients are differences between each given group (year) and the first group.

“Dispersion parameter” of 1 assumes that variance=mean.

The generalized linear model

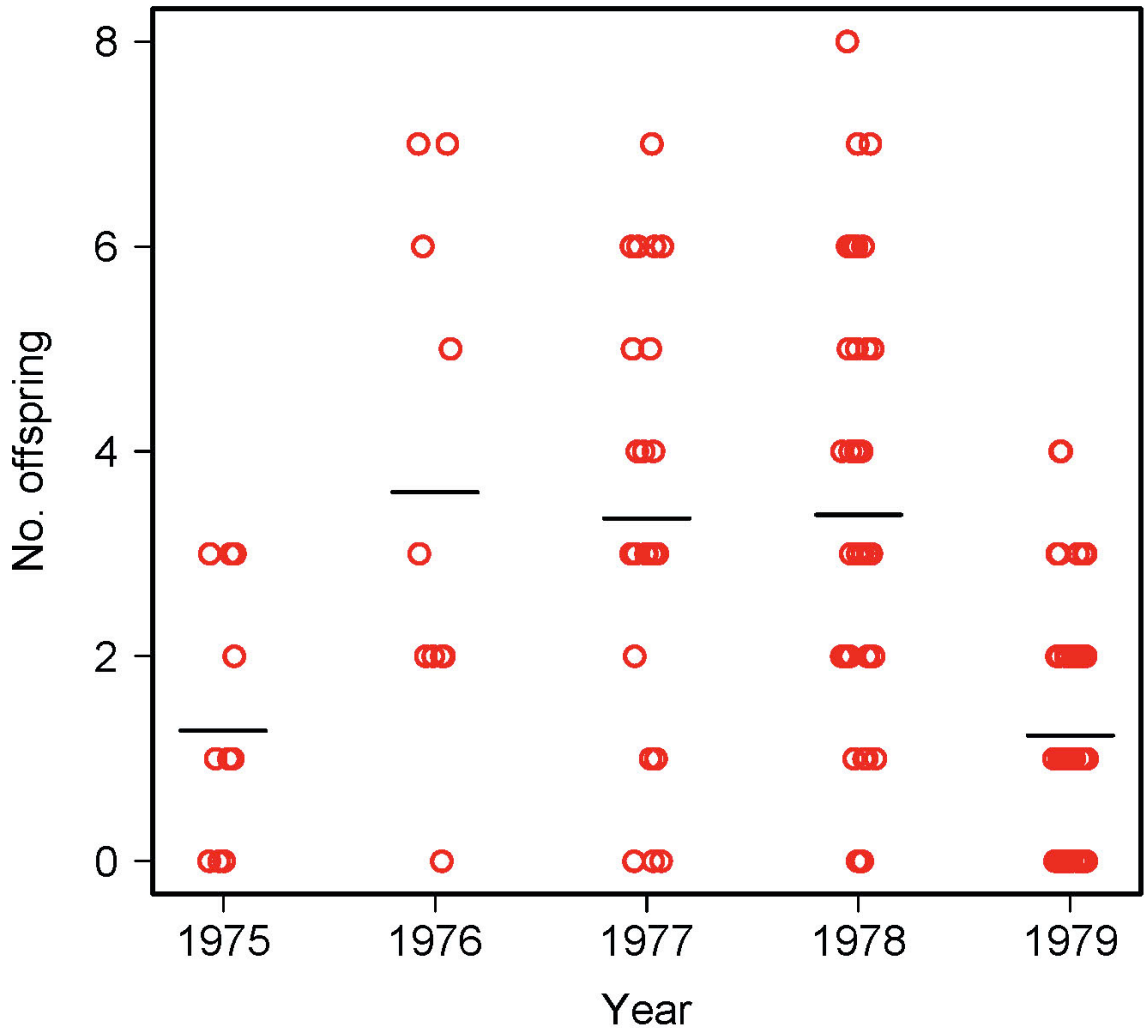
Predicted values on the log scale **`predict(z)`**

$$\hat{\eta} = 0.24116 + 1.03977(\text{year} == 1976) + \ldots$$

Predicted values on the original scale

`fitted.values(z)`

$$\hat{\mu} = \log(\hat{\eta})$$



The generalized linear model

Analysis of deviance table gives log-likelihood ratio test of the null hypothesis that there is no differences among years in mean number of offspring.

```
anova(z,test="Chi")

Terms added sequentially (first to last)

      Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                  145    288.656
year   4   75.575       141    213.081 1.506e-15 ***
```

The generalized linear model

As with `lm`, we can also fit the "means model" to get estimates of group means (here on the log scale).

```
z1 <- glm(noffspring ~ year-1, family=poisson(link="log"))

summary(z1)
           Estimate Std. Error z value Pr(>|z|)
 year1975   0.24116    0.26726   0.902   0.3669
 year1976   1.28093    0.16667   7.686 1.52e-14 ***
 year1977   1.20781    0.10721  11.266  < 2e-16 ***
 year1978   1.21816    0.08392  14.516  < 2e-16 ***
 year1979   0.20544    0.11952   1.719   0.0856
```

Back-transform to get estimates of means on original scale

```
exp(coef(summary(z1))[,1])

 year1975 year1976 year1977 year1978 year1979
 1.272727 3.600000 3.346154 3.380952 1.228070
```

Likelihood based confidence intervals

We can also get likelihood-based confidence intervals for the group means (command part of the MASS library)

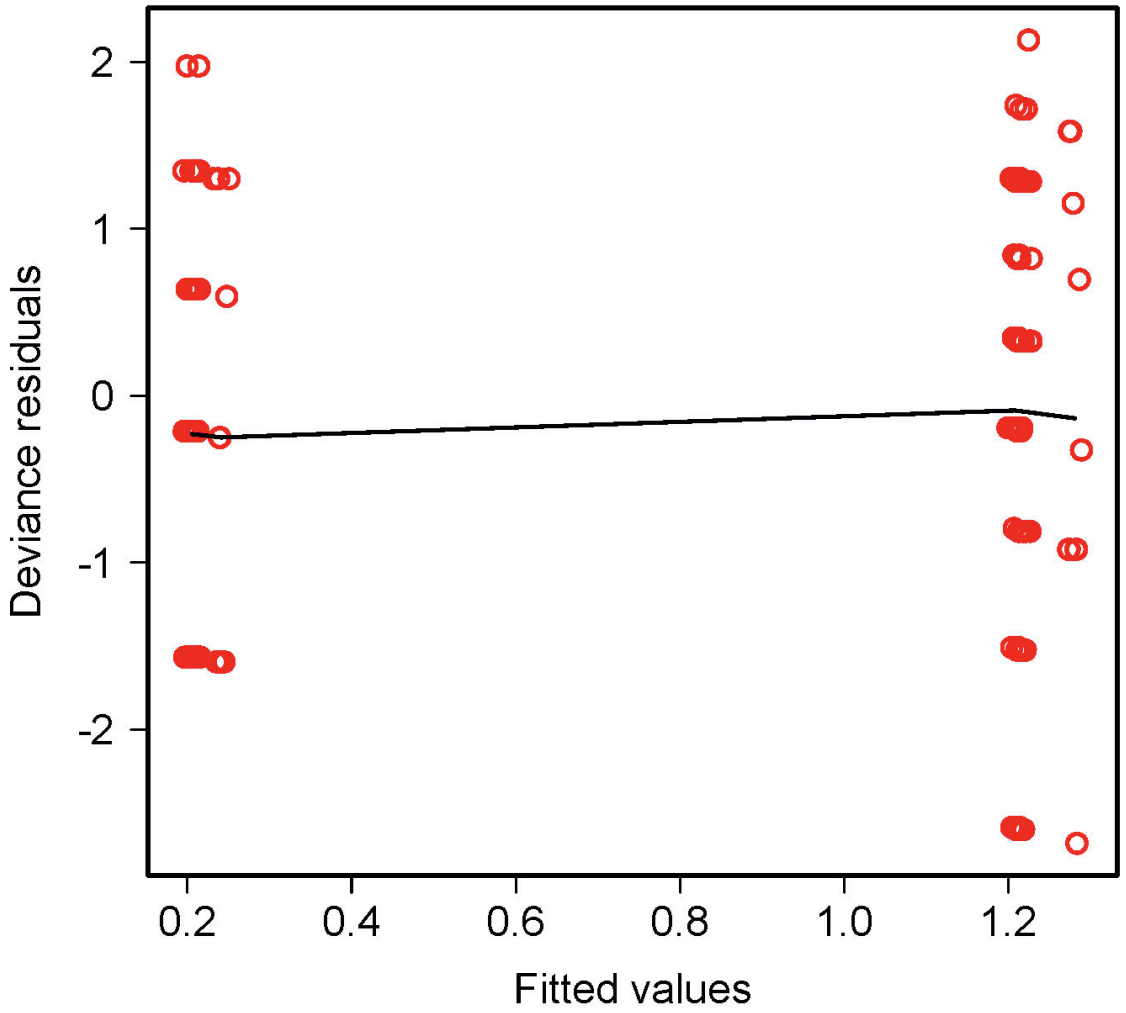
```
library(MASS)
confint(z1) # log scale
                2.5 %    97.5 %
 year1975 -0.33273518 0.7228995
 year1976  0.93546683 1.5907293
 year1977  0.99005130 1.4108270
 year1978  1.04904048 1.3782407
 year1979 -0.03833633 0.4308985

exp(confint(z1)) # back-transformed to original scale
              2.5 %   97.5 %
 year1975 0.7169600 2.060399
 year1976 2.5484028 4.907326
 year1977 2.6913725 4.099344
 year1978 2.8549105 3.967915
 year1979 0.9623892 1.538639
```

Evaluating assumptions of the `glm` fit

Residual plot (deviance residuals)

Can look strange because of discrete data.

Variances look roughly homogeneous.



Evaluating assumptions of the `glm` fit

Do the variances of the residuals correspond to that expected from the link function?

The log link assumes that the *Y* are Poisson distributed at each *X*

A central property of the Poisson distribution is that the variance and mean are equal (dispersion parameter = 1).

```
tapply(noffspring,year,mean)
tapply(noffspring,year,var)
```

```
     1975     1976     1977     1978     1979
 1.272727 3.600000 3.346154 3.380952 1.228070 means
 1.618182 6.044444 3.835385 4.680604 1.322055 variances
```

Variances slightly, but not alarmingly, larger than means.

(Note: the binomial link function also assumes a strict mean-variance relationship, specified by binomial dist'n (dispersion parameter = 1).)

Modeling excessive variance

Finding excessive variance (“overdispersion”) is typical when analyzing count data. Excessive variance occurs because other variables not included in the model affect the response variable.

In the workshop we will analyze an example where the problem is more severe than in the case of the song sparrow data here.

Modeling excessive variance

Excessive variance can be accommodated in glm by using a different link function, one that incorporates a dispersion parameter (which must also be estimated). A dispersion parameter >> 1 implies excessive variance.

Procedure is based on the relationship between mean and variance rather than an explicit probability distribution for the data. In the case of count data,

variance = (dispersion parameter) * (mean)

Method generates “quasi-likelihood” estimates that behave like maximum likelihood estimates.

Modeling excessive variance

Lets try it with the song sparrow data, and use the “means model”

```
z2 <- glm(noffspring ~ year-1, family=quasipoisson, data=x)

 summary(z)
          Estimate Std. Error t value Pr(>|t|)
 year1975   0.2412     0.2965   0.813    0.417
 year1976   1.2809     0.1849   6.928 1.40e-10 ***
 year1977   1.2078     0.1189  10.155  < 2e-16 ***
 year1978   1.2182     0.0931  13.085  < 2e-16 ***
 year1979   0.2054     0.1326   1.549    0.124

 Dispersion parameter for quasipoisson family taken to be
 1.230689
```

The point estimates are identical with those obtained earlier, but the standard error (and resulting confidence intervals) are wider.

The dispersion parameter is reasonably close to 1 for these data, so original approach using poisson link probably ok.

Other uses of generalized linear models

We have used glm to model binary frequency data, and count data.

The method is also commonly used to model r x c (and higher order) contingency tables, in which cell counts depend on two (or more) categorical variables each of which may have more than two categories or groups.

Finally, glm can handle data having other probability distributions than the ones used in my examples, including normal and gamma distributions.

Discussion paper for next week:

Whittingham et al (2006) Why do we still use stepwise modelling?

Download from "**assignments**" tab on course web site.

Presenters: Allison & …..

Moderators: ….. & …..