Outline for today

- What is a generalized linear model

- Linear predictors and link functions

- Example: fit a constant (the proportion)

- Analysis of deviance table

- Example: fit dose-response data using logistic regression

- Example: fit count data using a log-linear model

- Advantages and assumptions of `glm`

- Modeling overdispersion (excessive variance)

- Example: Modeling contingency tables

Review: what is a linear model

A model of the following form:

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

- 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 <u>equal variance</u> at all values of the X variables.

- Uses least squares to fit model to data and to estimate parameters

- Use `lm()` in R

Review: fitting a linear model in R

Use `lm()` in R

Simplest linear model: fit a constant (the mean)

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


Linear regression

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

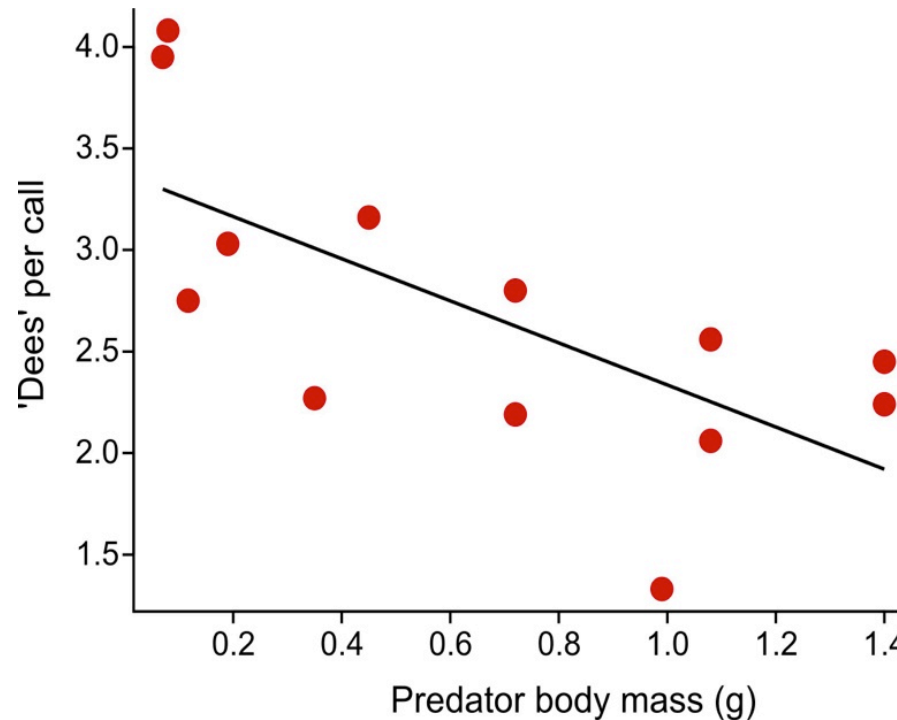
Single factor ANOVA

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

Review: what is a linear model

Eg: linear regression: $Y = \beta_0 + \beta_1 X +$ error

The predicted Y-values, symbolized here by μ, is modeled as

$$\mu = \beta_0 + \beta_1 X$$

The part to the right of "=" is the linear predictor

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 "=")
- But the predicted *Y*-values are transformed
- $g(\mu)$ is called the "link function," of which there are several types
- 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
- Fit models using `glm()` in R

The two most common link functions

1) Natural log (i.e., base *e*)

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

Usually used to model <u>count data</u> (e.g., number of mates, etc)

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

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

The two most common link functions

2) Logistic or logit

$$\log \frac{\mu}{1 - \mu} = \eta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots$$

Used to model <u>binary data</u> (e.g., survived vs died)

The link function $\log \frac{\mu}{1-\mu}$ is also known as the log-odds

The inverse function is $\mu = \dfrac{e^\eta}{1+e^\eta}$

Example 1: Fit a constant to 0-1 data (estimate a proportion)

This example was used previously in Likelihood lecture. My goal with this example is to connect what `glm()` does with what we did by brute force last week.

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?
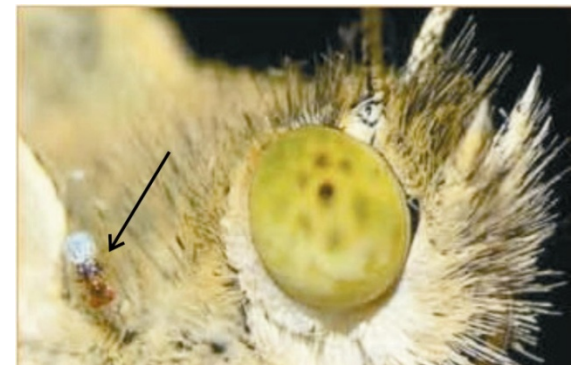
Number of wasps choosing the mated female fits a binomial distribution

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.

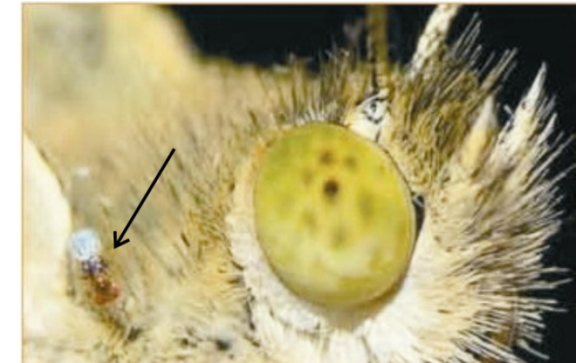To model these data, let "success" be "wasp chose mated butterfly"

Y = 23 successes
n = 32 trials

Goal: estimate p

Data are :
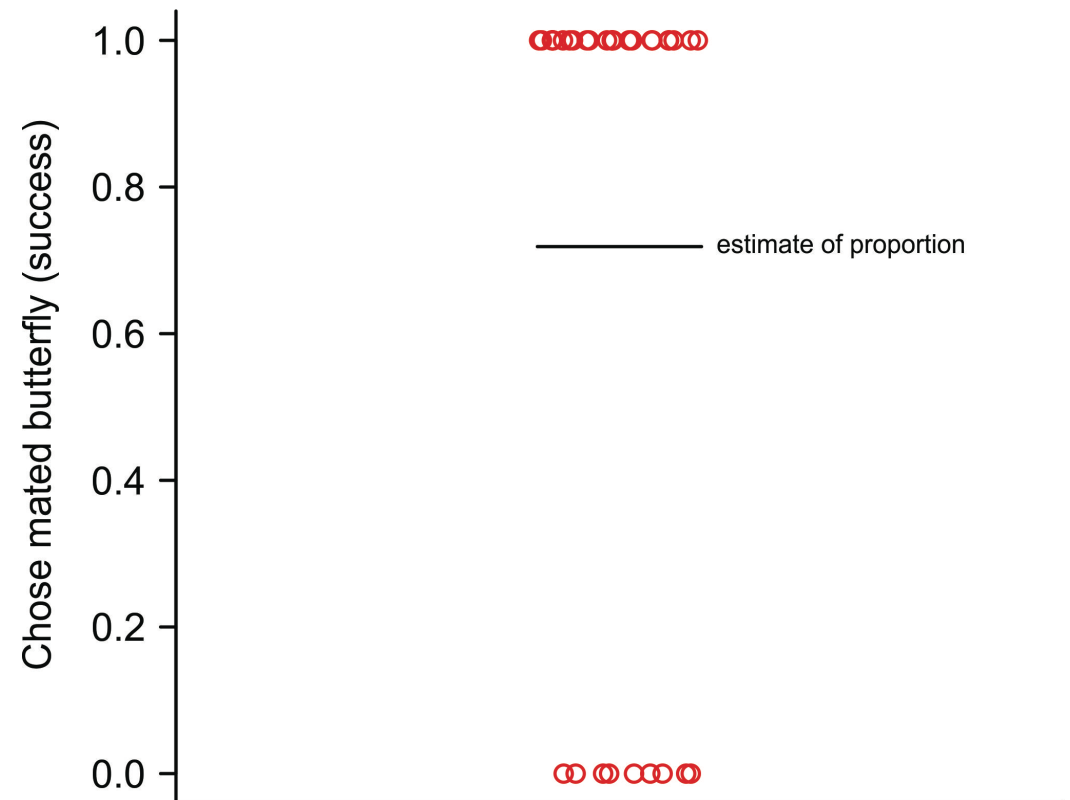1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 0 0 1 1

©BobMoul

Use `glm()` to fit a constant, and so obtain the ML estimate of p

The data are binary. Each wasp has a measurement of 1 or 0 ("success" or "failure"): 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 0 0 1 1

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

`family` specifies the error distribution (binomial) and the link function (logit).

Use `glm()` to fit a constant, and so obtain the ML estimate of *p*

Fits a model having only a constant. Use the link function appropriate for binary data:

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

μ here refers to the population proportion (p) but let's stick with μ symbol here to use consistent notation for generalized linear models.

Fitting will yield the estimate, $\hat{\beta}$.

The estimate of proportion $\hat{\mu}$ is then obtained using the inverse function:

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

Use `summary()` for estimation

```
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 constant on the **logit** scale). Convert back to ordinary scale (plug into inverse equation) to get estimate of population proportion:

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

This is the ML estimate of the population proportion. This is identical to the estimate obtained last week using likelihood function.
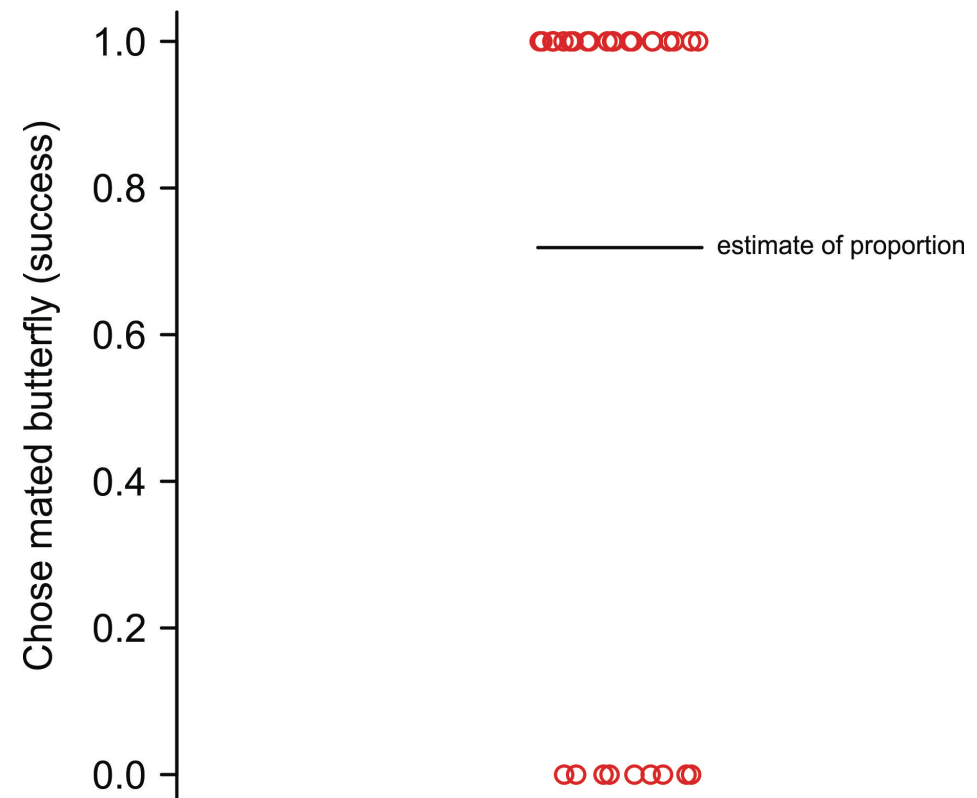
Confidence intervals

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

95% confidence limits:

```
myCI <- confint(z)          # on logit scale
exp(myCI)/(1 + exp(myCI)) # inverse logit scale

    2.5 %      97.5 %
0.5501812 0.8535933
```

$0.550 \leq p \leq 0.853$ is the same result we obtained last week for likelihood based confidence intervals using likelihood function (more decimal places this week).

Avoid using `summary()` for hypothesis testing

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

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

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

Agresti (2002, *Categorical data analysis*, 2nd ed., Wiley) says that for small to moderate sample size, the Wald test is <u>less reliable</u> than the log-likelihood ratio test.

Use `anova()` to test hypotheses

Last week we calculated the log-likelihood ratio test for these data "by hand".
Here we'll use `glm()` to accomplish the same task.

"Full" model (β estimated from data):

```
z1 <- glm(y ~ 1, family = binomial(link="logit"))
```

"Reduced" model (β set to 0 by removing intercept from model):

```
z0 <- glm(y ~ 0, family = binomial(link="logit"))
```
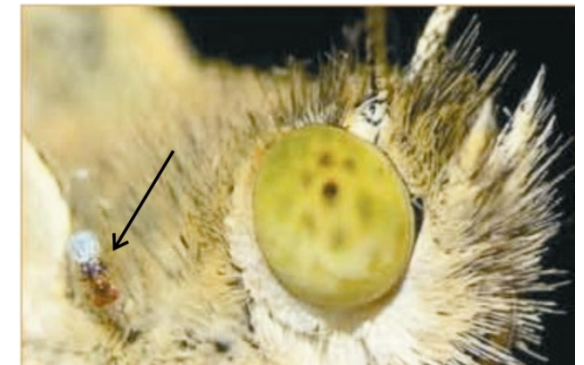
Use `anova()` to test hypotheses

```
anova(z0, z1, test  = "Chi") # Analysis of deviance
```
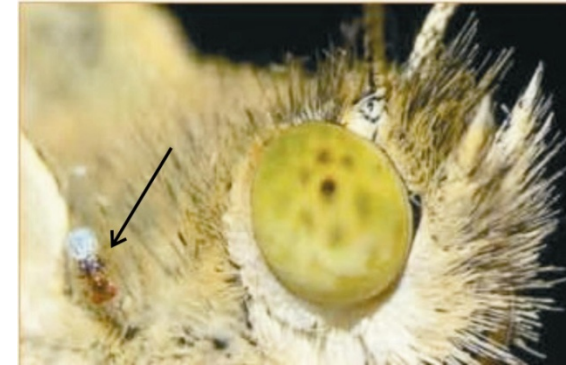
```
 Model 1: y ~ 0      # Reduced model
 Model 2: y ~ 1      # Full model
```


Analysis of deviance table:

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

The <u>deviance</u> is the log-likelihood ratio statistic (G-statistic). It has an approximate χ^2 distribution under the null hypothesis.

Residual deviance measures goodness of fit of the model to the data.

G = 6.227 is the identical result we obtained "by hand" using log likelihood ratio test last week.

Example 2: Logistic regression

One of the most common uses of generalized linear models.
Goal is to model the relationship between a proportion and an explanatory variable

Data: 72 rhesus monkeys (*Macacus rhesus*) exposed for 1 minute to aerosolized preparations of anthrax (*Bacillus anthracis*).

Goal is to estimate the relationship between dose and probability of death.
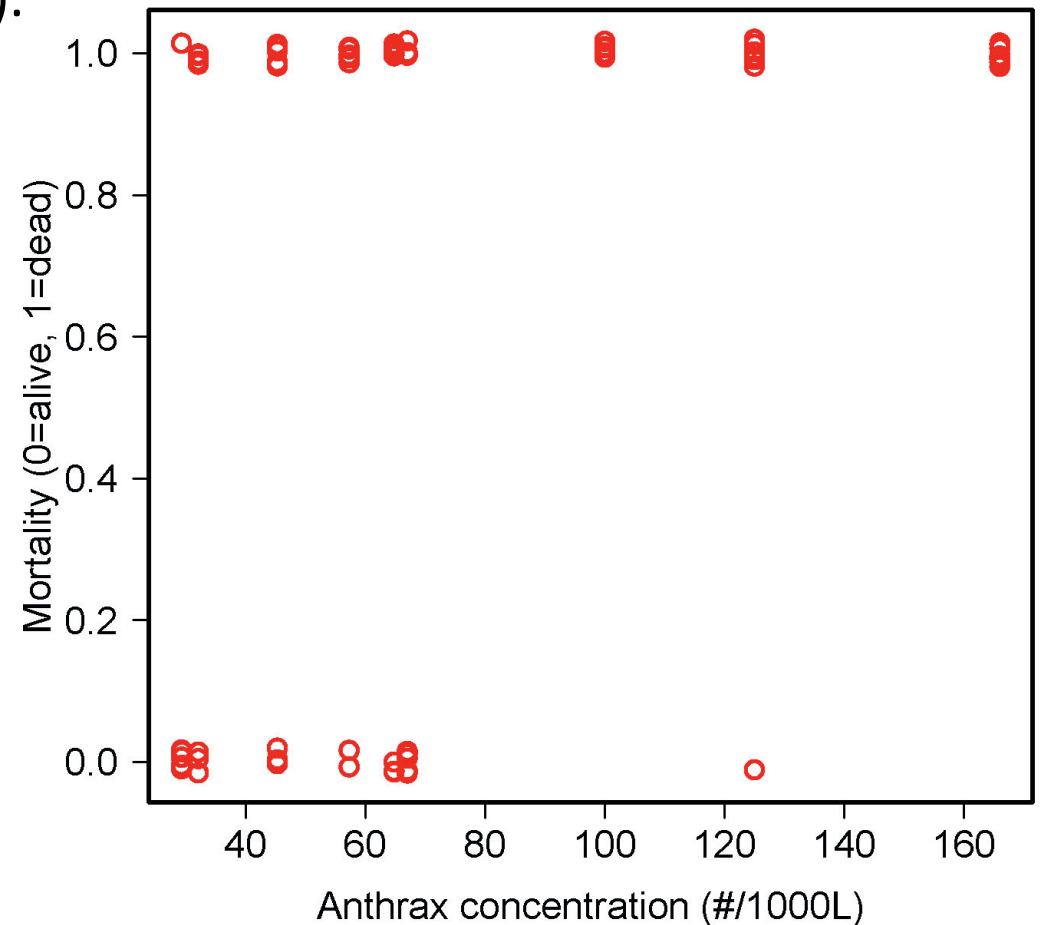
Logistic regression

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

Ordinary 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
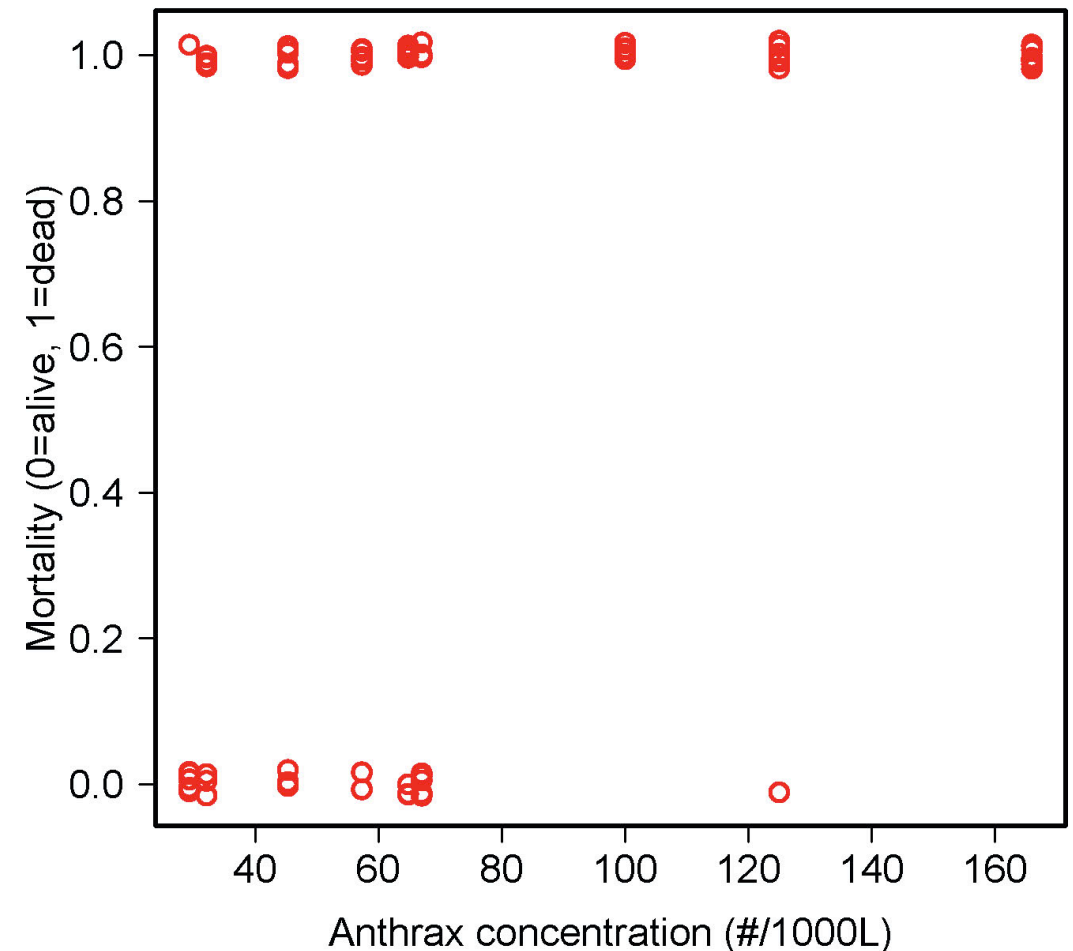
The generalized linear model

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

μ is the <u>probability</u> 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

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

`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 on the likelihood surface.
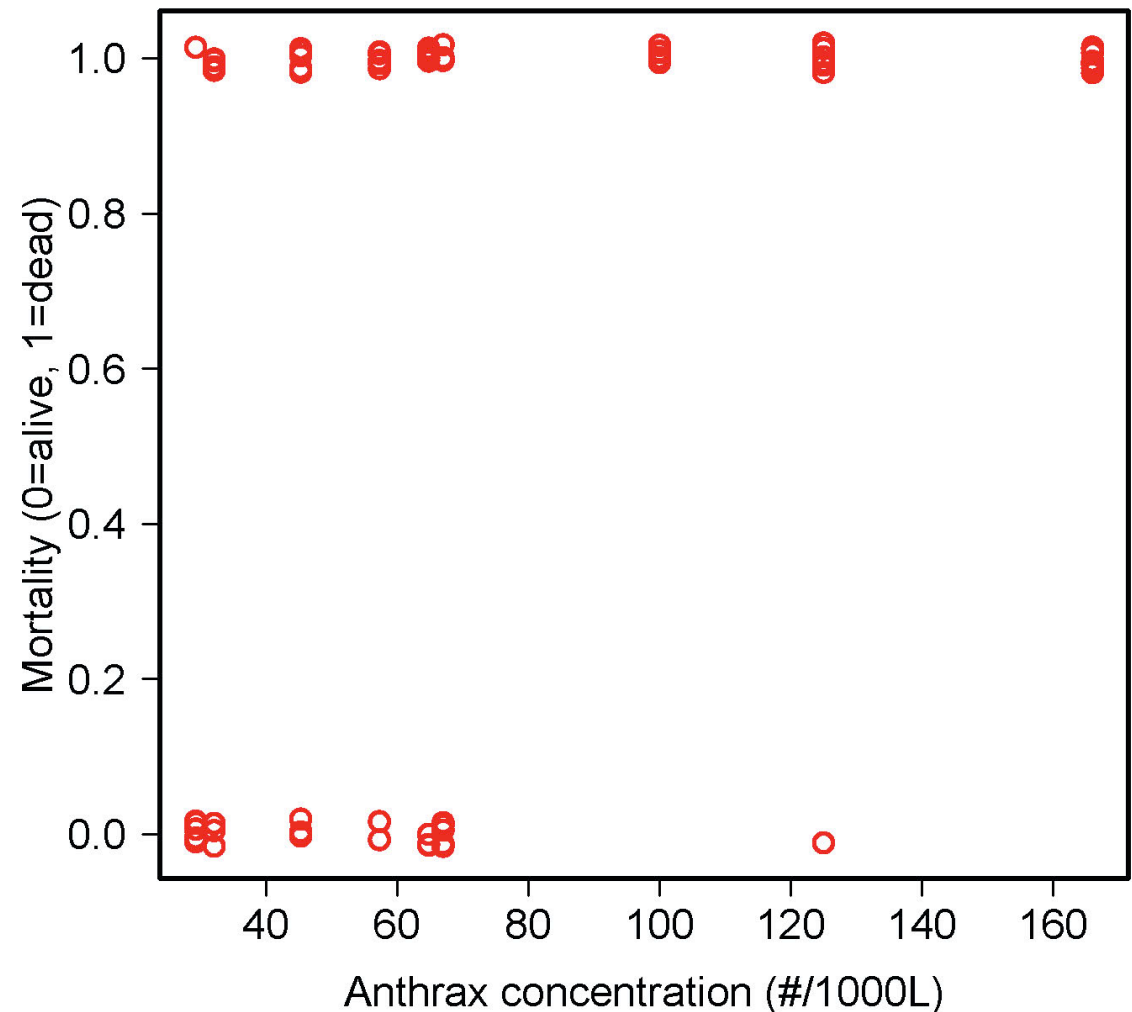
Use `summary()` for estimation

```
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) which predict $\log(\mu / 1 - \mu)$.

Number of Fisher Scoring iterations 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

$$g(\widehat{\mu}) = \widehat{\eta} = -1.74 + 0.036X$$

`visreg(z)` uses `predict` to plot predicted values with confidence limits on logit scale. See that the function is a line.

The points on this scale are not the logit-transformed data. Instead, `glm()` creates "working" values to fit model to data on transformed scale.
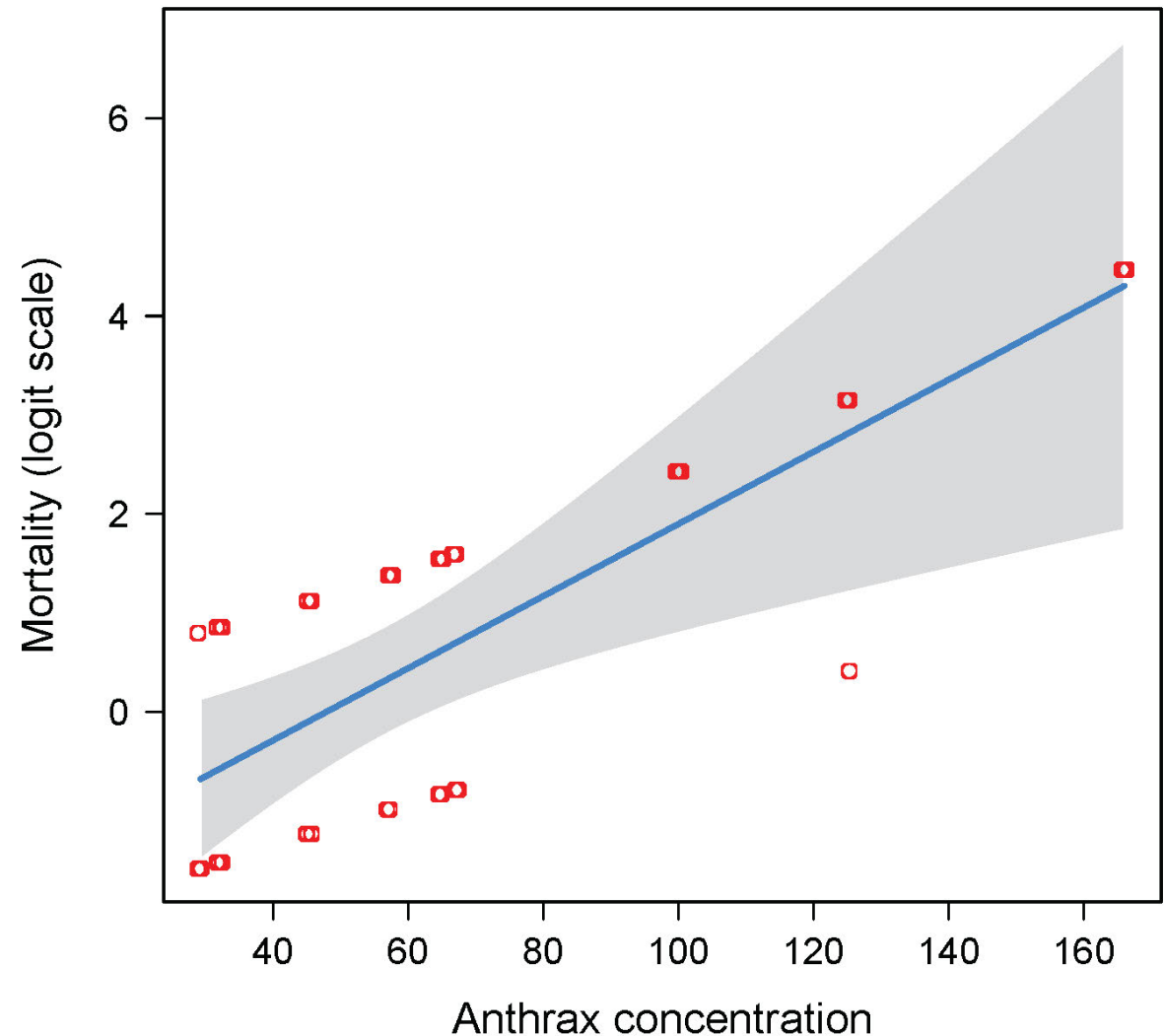
The generalized linear model

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

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

Use
`visreg(z,`
`    scale = 'response')`
to get fitted curve with
confidence bands on the
original scale.
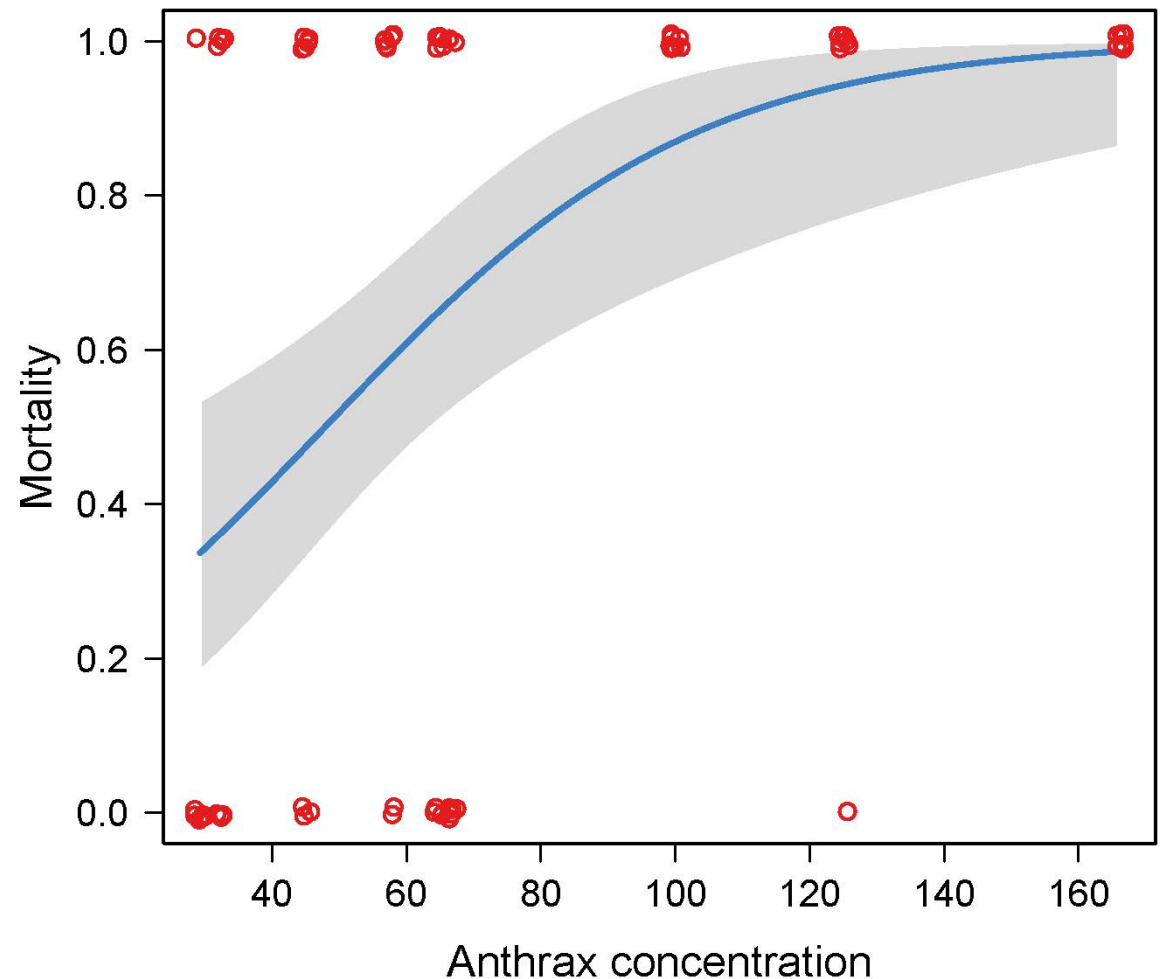
The generalized linear model

$$\text{LD}_{50} = -\frac{\text{intercept}}{\text{slope}} = -\frac{\hat{\beta}_0}{\hat{\beta}_1} = -\frac{0.03643}{-1.7445} = 47.88$$

The parameter estimates from the model fit can be used to estimate LD$_{50}$, the estimated concentration at which 50% of individuals are expected to die.

```
library(MASS)
dose.p(z)
```

```
              Dose           SE
p = 0.5:  47.8805  8.168823
```
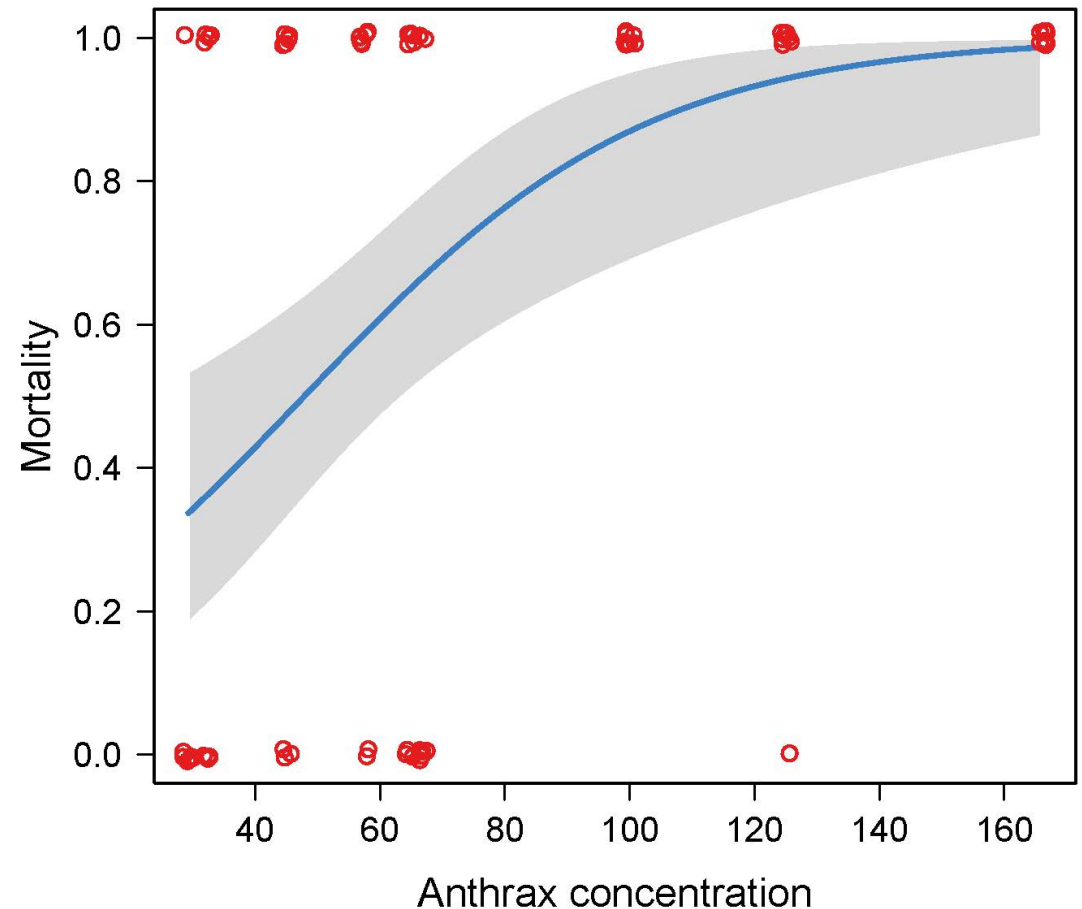
Use `anova()` to test hypotheses

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="Chisq")
```

	Df	Deviance	Resid Df	Resid Dev	P(>|Chi|)
NULL			71	92.982	
concentration	1	19.020	70	73.962	1.293e-05

As with `lm()`, terms are tested using model comparison (always a "full" vs "reduced" model). Default program of action is to fit terms sequentially ("Type 1 sums of squares"), just like `lm()`.

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.

- Avoids the problems associated with transforming 0's and 1's. For example, the logit transformation of 0 or 1 can't be computed.

- Retains the same analysis framework as linear models.

When `glm()` is appropriate and when it is not

treatment:	A	B	B	A	B	A
response: (survival)	1	0	0	0	0	1

Glm is appropriate (individual is the replicate)

treatment:	A	B	B	A	B	A
response: (survival)	1,1,1,0	0,0,0,1	0,0,0,0	0,1,0,1	0,1,0,0	1,1,1,1

Glm is not appropriate (tank is the replicate)

In second case, analyze summary statistic (fraction surviving) with `lm()`. Fitting generalized linear mixed models is possible in `lme4` package using `glmm()` method.

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 assumed by the link function. (A method to deal with violation of this assumption is shown below).

Example 3: Analyzing count data with log-linear regression

Estimate mean number of offspring fledged by female song sparrows on Mandarte Island, BC.

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

Linear model assumptions not met:

Data are discrete counts (non-normal).
Variance increases with mean.
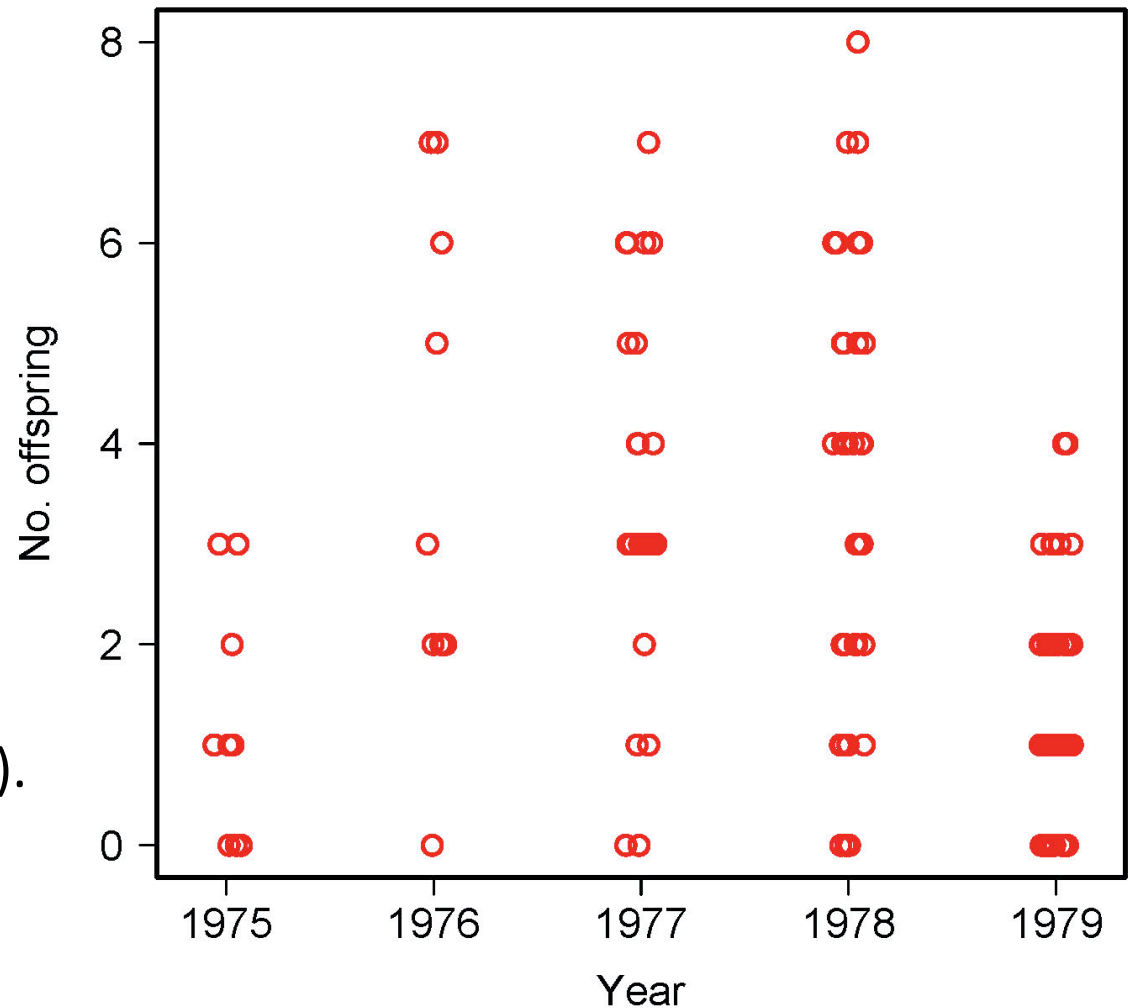
Example 3: Analyzing count data with log-linear regression

Estimate mean number of offspring fledged by female song sparrows on Mandarte Island, BC.

Linear model assumptions not met:

Data are discrete counts (non-normal). Variance increases with mean.

Two solutions:

1. Transform data: $X' = \log(X + 1)$

2. Generalized linear model. Poisson distribution might be appropriate for error distribution. So try log link function.
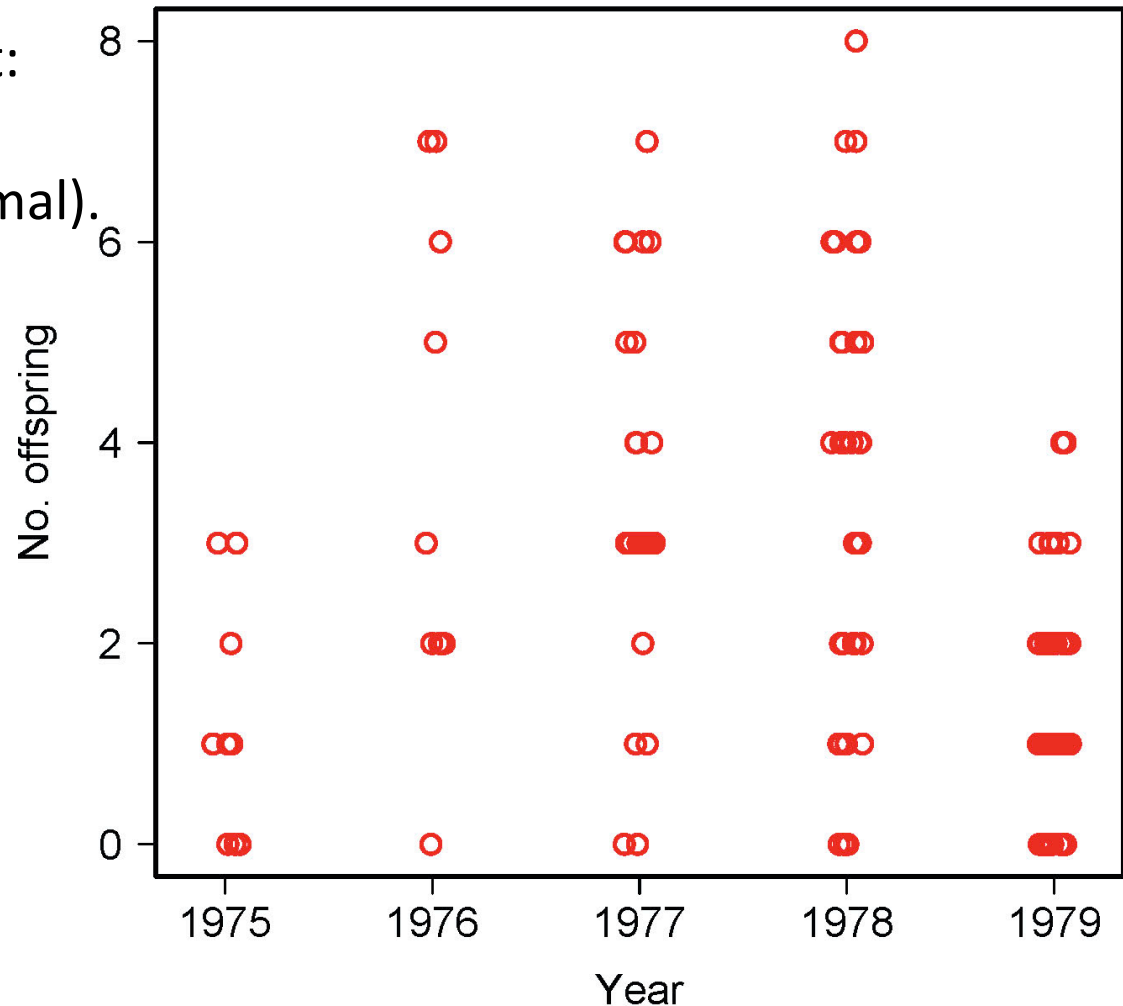
Example 3: Analyzing count data with log-linear regression

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

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

η is the response variable on the log scale (here, mean of each group on log scale).

Year is a categorical variable. So is analogous to single factor ANOVA.

Categorical variables are modeled in R using "dummy" indicator variables, as with `lm()`.

Use `summary()` for estimation

```
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 <u>log scale</u>.
Intercept refers to 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 states assumption that in every year, variance = mean.

Predicted values on the transformed scale

Predicted values on the transformed (log) scale: `predict(z)`

`visreg(z)` uses `predict` to plot the predicted values, with confidence limits, on this transformed scale.

See that the "data points" on this scale are not just the transformed data (we can't take log of 0). Instead, `glm()` creates "working" values to fit the model to the data on the transformed scale.
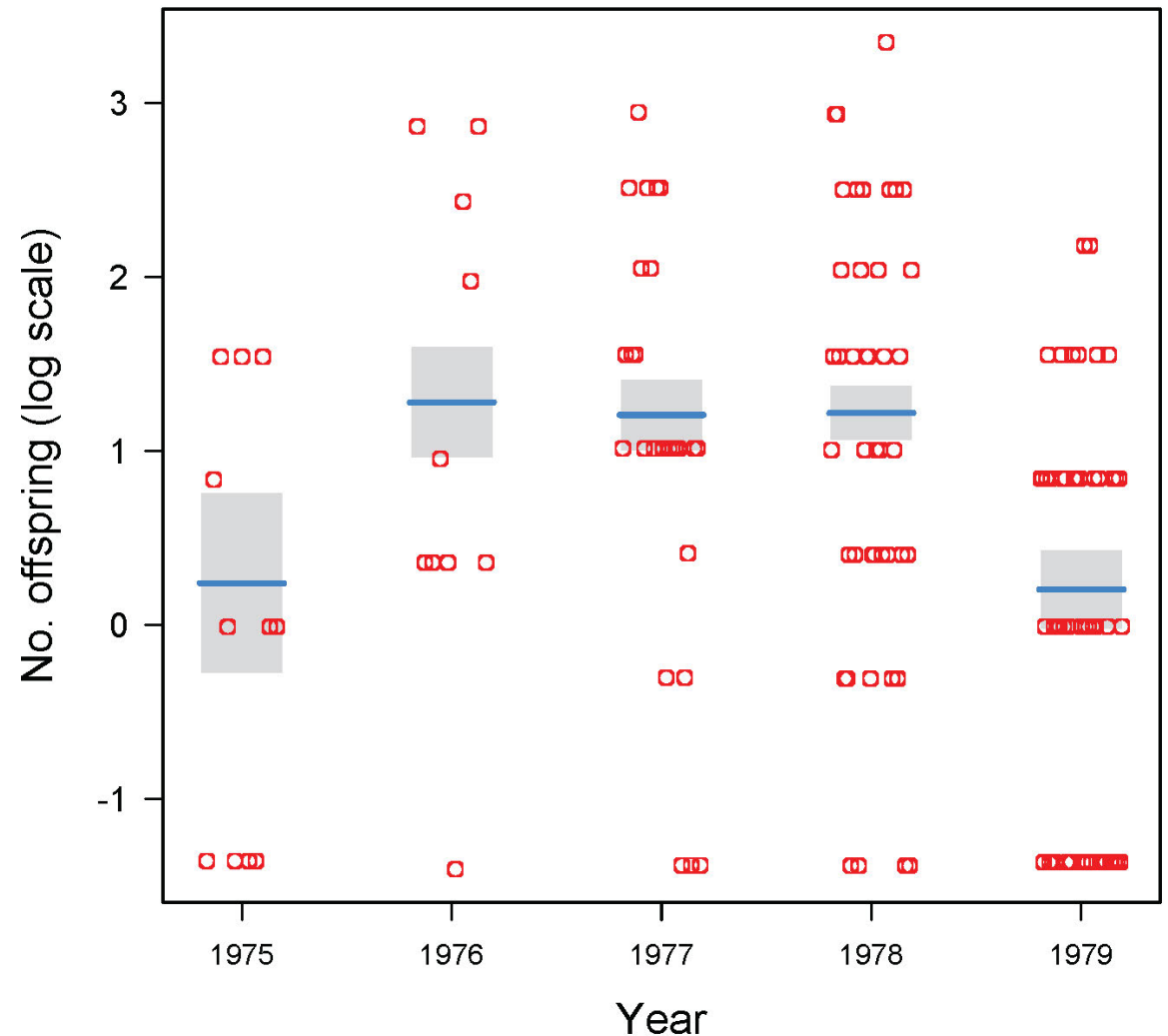
Predicted values on the original scale

Predicted values on original scale:
`fitted.values(z)`

$$\hat{\mu} = e^{\hat{\eta}}$$

I have plotted them here using `visreg()` and superimposed the original data points.

Note that the fitted values aren't the means of the original data. Fitted values are the transformed means estimated on the log scale (geometric means).
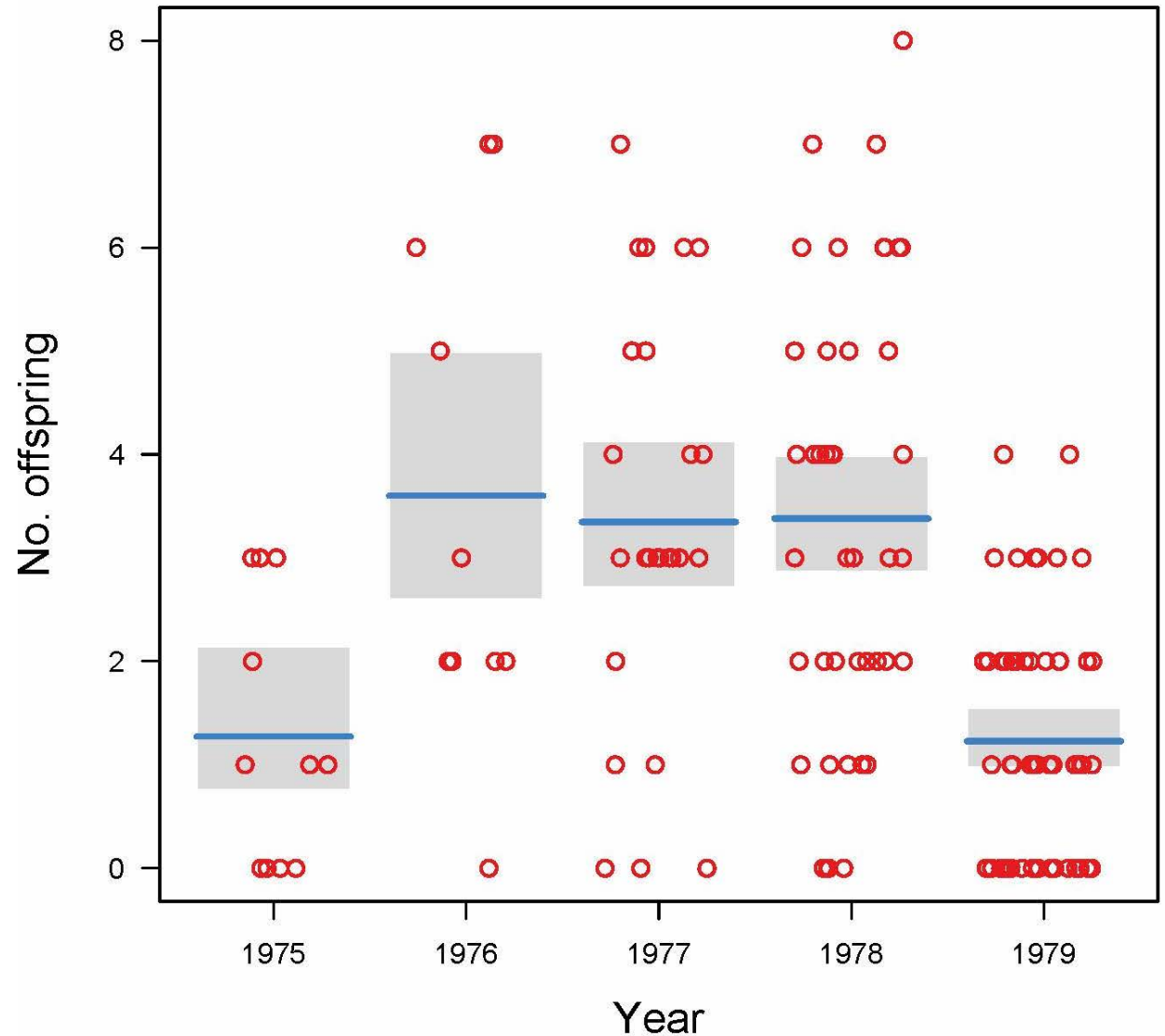
Use `emmeans()` to obtain model predicted values on original scale

```
emmeans(z, "year", type = "response")
```

Yields model-fitted predicted values and approximate 95% confidence intervals.

year	rate	SE	df	asymp.LCL	asymp.UCL
1975	1.272727	0.3401496	Inf	0.7537770	2.148957
1976	3.600000	0.6000000	Inf	2.5967825	4.990791
1977	3.346154	0.3587453	Inf	2.7119865	4.128614
1978	3.380952	0.2837232	Inf	2.8681893	3.985385
1979	1.228070	0.1467822	Inf	0.9715952	1.552248

Uses a large-sample approximation (degrees of freedom (df) are shown as infinite), and confidence limits might not be accurate for small sample sizes.

Use `anova()` to test hypotheses

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="Chisq")
```

```
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	***

As with `lm()`, terms are tested using model comparison (always a "full" vs "reduced" model). Default program of action is to fit terms sequentially ("Type 1 sums of squares"), as with `lm()`.

Evaluating assumptions of the `glm()` fit

Do the variances of the residuals correspond to those assumed by the chosen link function?

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

A key property of the Poisson distribution is that within each treatment group the variance and mean are equal (i.e., the `glm()` dispersion parameter = 1).

Similarly, when analyzing binary data, the logit link function also assumes a strict mean-variance relationship, specified by binomial distribution, when dispersion parameter = 1. This is rarely violated with 0/1 data, so I focus on log-linear model here.

Evaluating assumptions of the `glm()` fit

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

Let's check the sparrow data:

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

```
      1975       1976       1977       1978       1979
 1.272727   3.600000   3.346154   3.380952   1.228070   # mean
 1.618182   6.044444   3.835385   4.680604   1.322055   # variance
```

Variances slightly, but not alarmingly, larger than means.

Modeling excessive variance

Finding excessive variance ("overdispersion") is common when analyzing count data. Excessive variance occurs because variables not included in the model also 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 with `glm()` by using a different link function, one that incorporates a dispersion parameter (which must be estimated from the data). If the estimated dispersion parameter is >> 1 then there is likely excessive variance.

The `glm()` procedure to accomplish over (or under) dispersion uses the observed 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

```
z <- glm(noffspring ~ year, family = quasipoisson)

summary(z)
           Estimate Std. Error t value Pr(>|t|)
Intercept)   0.24116    0.29649   0.813  0.41736
year1976     1.03977    0.34942   2.976  0.00344 **
year1977     0.96665    0.31946   3.026  0.00295 **
year1978     0.97700    0.31076   3.144  0.00203 **
year1979    -0.03572    0.32479  -0.110  0.91259

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

The dispersion parameter is reasonably close to 1 for these data. But typically it is much larger than 1 for count data, so use `family = quasipoisson`.

Modeling excessive variance

Lets try it with the song sparrow data

```
z <- glm(noffspring ~ year, family = quasipoisson)
```

```
summary(z)
           Estimate Std. Error t value Pr(>|t|)
Intercept)   0.24116    0.29649   0.813   0.41736
year1976     1.03977    0.34942   2.976   0.00344  **
year1977     0.96665    0.31946   3.026   0.00295  **
year1978     0.97700    0.31076   3.144   0.00203  **
year1979    -0.03572    0.32479  -0.110   0.91259
```

Dispersion parameter for quasipoisson family taken to be 1.230689

The point estimates are identical with those obtained using `family=poisson` instead, but the standard errors (and resulting confidence intervals) are wider.

Example 4: Modeling contingency tables

Example: Incidence of malaria in female great tits in relation to experimental treatment. n = 65 birds.

Grouped bar graph

Explanatory variable = outer groups; response variable = inner groups
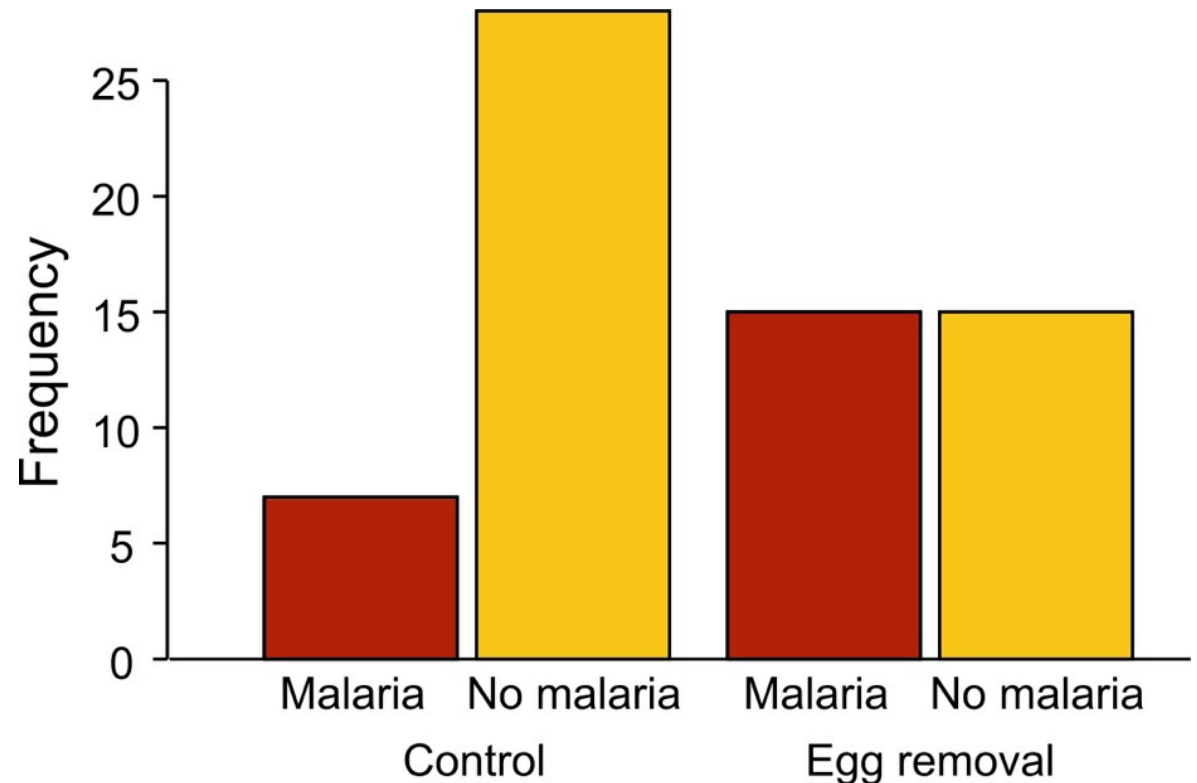
Example 4: Modeling contingency tables

Response	Treatment	Frequency
Malaria	Control	7
No Malaria	Control	28
Malaria	Egg removal	15
No Malaria	Egg removal	15

```
z <- glm(Frequency ~ Treatment + Response, data = mydata,
         family = poisson(link = "log"))
```

The counts are modeled in an analogous way to modeling of means in an ordinary linear model.

Example 4: Modeling contingency tables

```
visreg(z, xvar="Response", by="Treatment", scale="response")
```
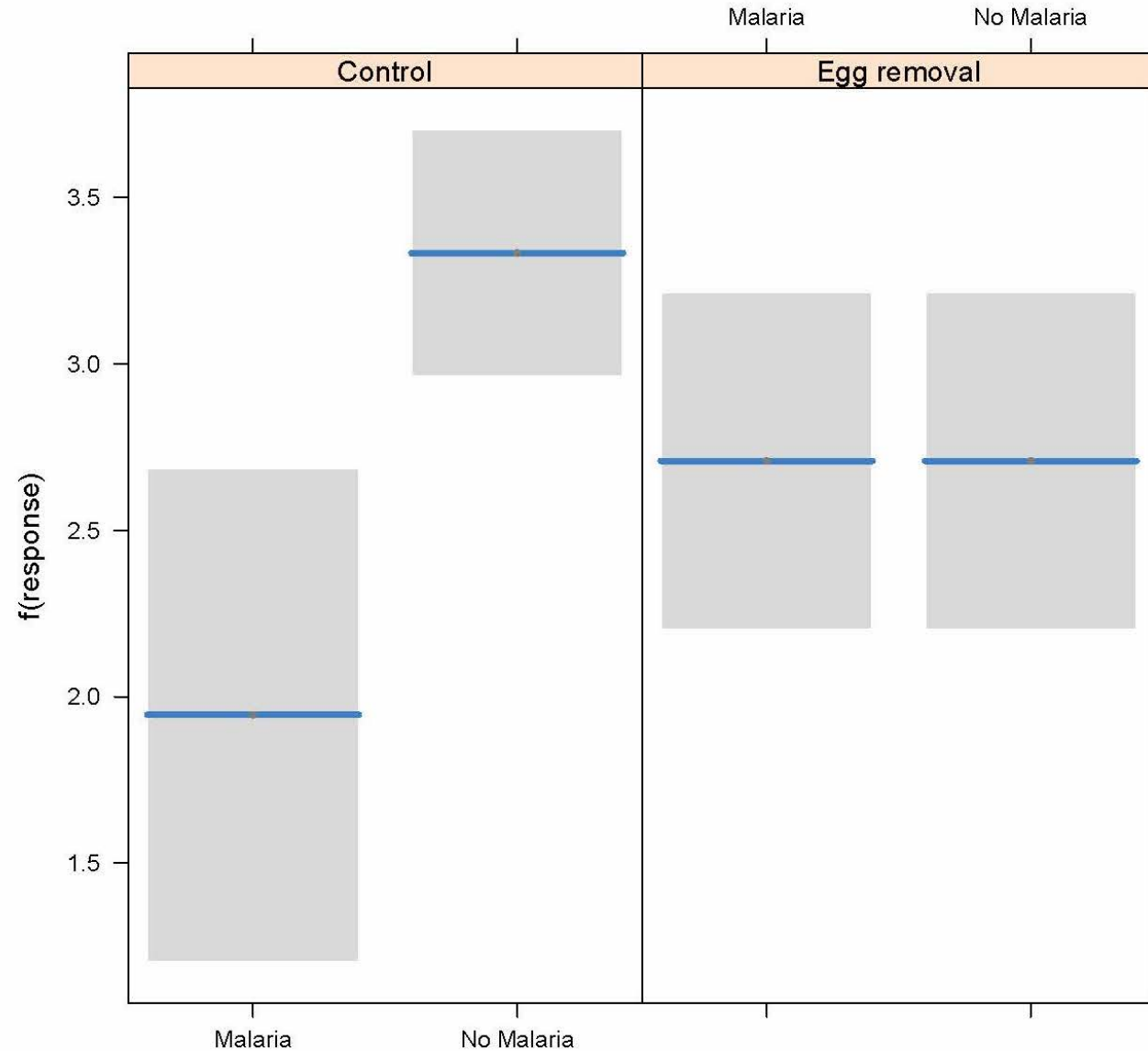
Use `anova()` to test hypotheses

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

	Df	Deviance	Resid. Df	Resid. Dev	P
NULL	3	13.8771			
Treatment	1	0.3850	2	13.4921	0.534942
Response	1	6.9079	1	6.5843	0.008582
Treatment:Response	1	6.5843	0	0.0000	0.010288

The interaction between Treatment and Response is the test of association.

Ignore the main effects.

Other uses of generalized linear models

The method is especially useful when analyzing higher-order contingency tables, where there may be two- and three-way interactions. Each of the interactions can be assessed separately.

`glm()` can handle data having other probability distributions than the ones used in my examples, including exponential 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:

Moderators: Gabriel & Keerthi