

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:

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

- 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.
- Use `lm()` in R when analyzing fixed effects

Review: fitting a linear model in R

Use `lm()` in R when analyzing fixed effects

Simplest linear model: fit a constant (the mean)

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

Linear regression

```
z <- lm(y ~ x) # x is numeric
```

Single factor ANOVA

```
z <- lm(y ~ A) # A is categorical
```

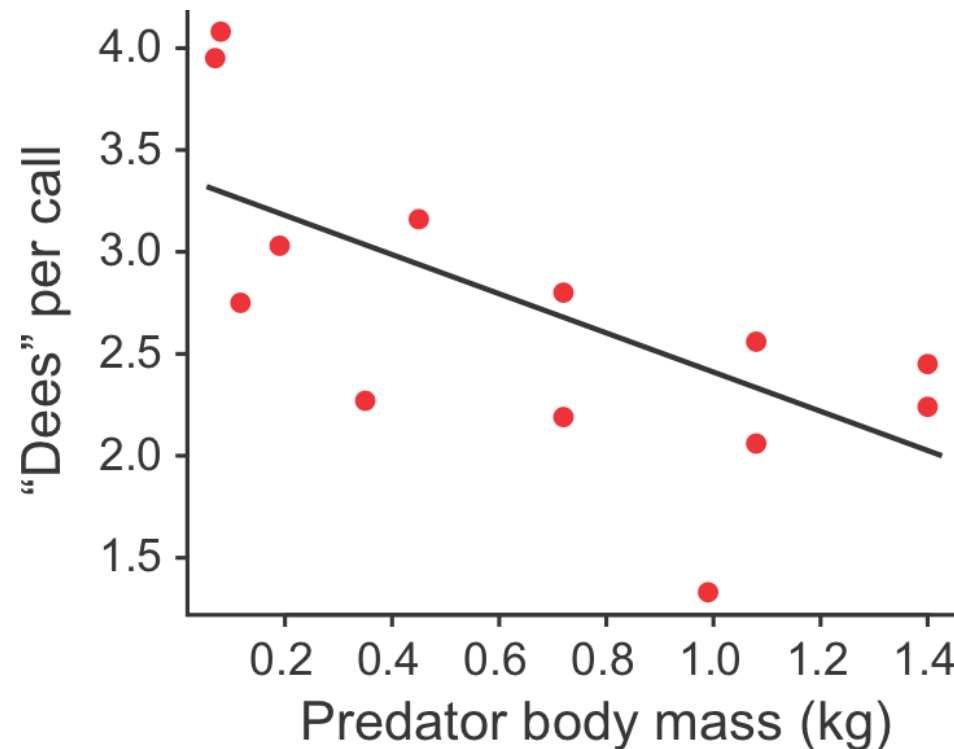
Review: what is a linear model

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

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

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

The part to the right of “=” is the linear predictor



What is a generalized linear model

A model whose predicted values are of the form

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

- The model still includes a *linear predictor* (to right of “=”)
- But now what’s being predicted is a transformation of μ , ie $g(\mu)$
- $g(\mu)$ is called the “link function,” several in common use.
- Non-normal residuals OK (probability distribution is specified)
- Unequal error variances OK (determined by probability distribution)
- 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)

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

Greek mu Greek eta

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

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

η is the predicted count on log scale, μ is the predicted count.

Use the inverse function to get the predicted count: $\mu = e^\eta$.

The two most common link functions

2) Logistic (a.k.a. “logit”)

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

Greek mu Greek eta

Used to model binary data (e.g., survived vs died)

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

η is the predicted proportion on logit scale, μ is the predicted proportion.

Use the inverse function to get the predicted proportion: $\mu = \frac{e^\eta}{1+e^\eta}$

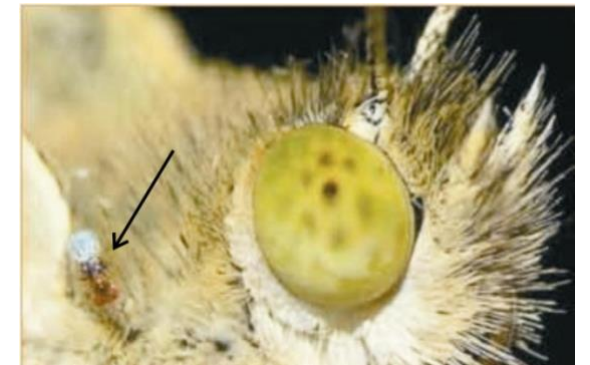
Example 1: Estimate a proportion by fitting a constant to 0-1 data

This example was used previously in likelihood lecture. My goal here is use $g_{lm}()$ to redo what we did by brute force previously.

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

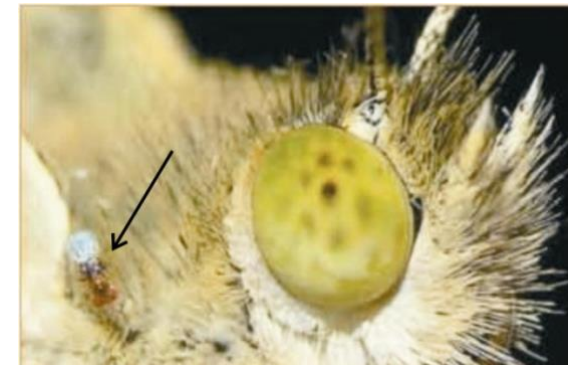
Under random sampling, the number of “successes” in n trials has a binomial distribution, with p being the probability of “success” in any one trial.

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

$Y = 23$ successes

$n = 32$ trials

Goal: estimate the proportion p

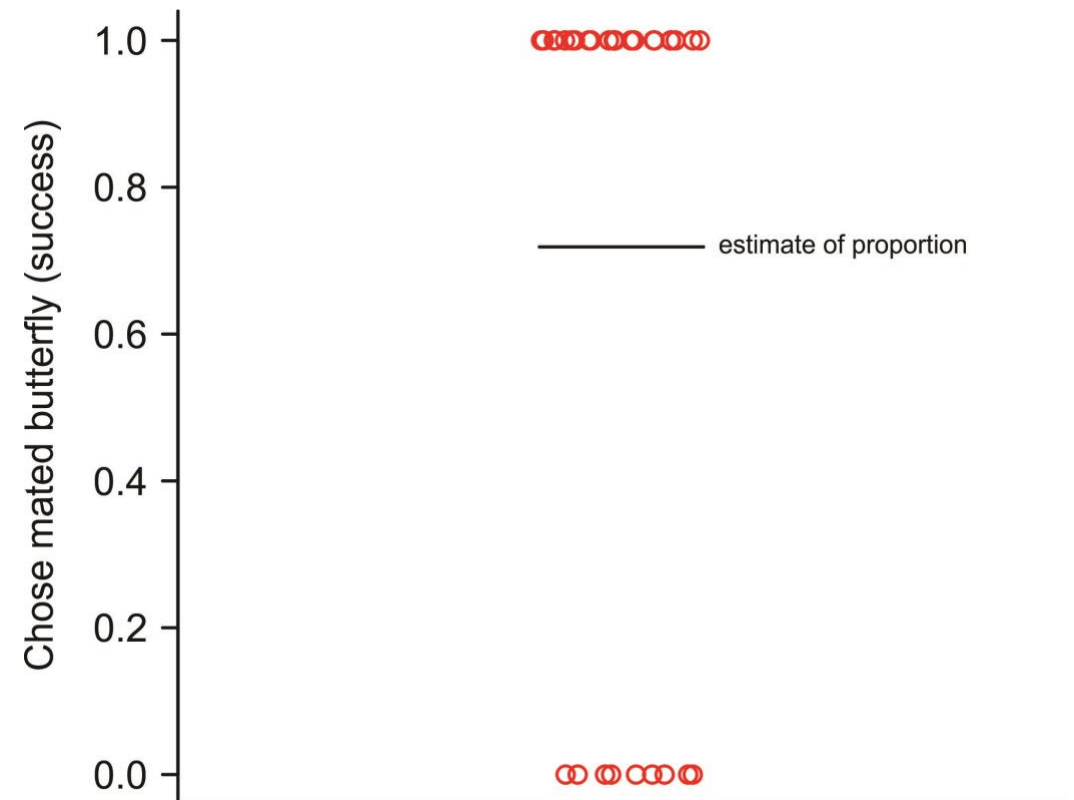


Fit a `glm()` to obtain the ML estimate of a proportion

The data are binary. Each wasp has a measurement of 1 or 0 (“success” or “failure”) for `choice`: 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 1 0 0 1 1

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

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



Fit a `glm()` to obtain the ML estimate of a proportion

The logit is the link function appropriate for binary data.
The generalized linear model to fit a proportion:

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

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

Fitting will yield the estimate of the single coefficient, $\hat{\beta}$, which is the proportion on the logit scale. The estimate of proportion $\hat{\mu}$ is then the inverse of the logit function:

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

Use `summary()` for estimation

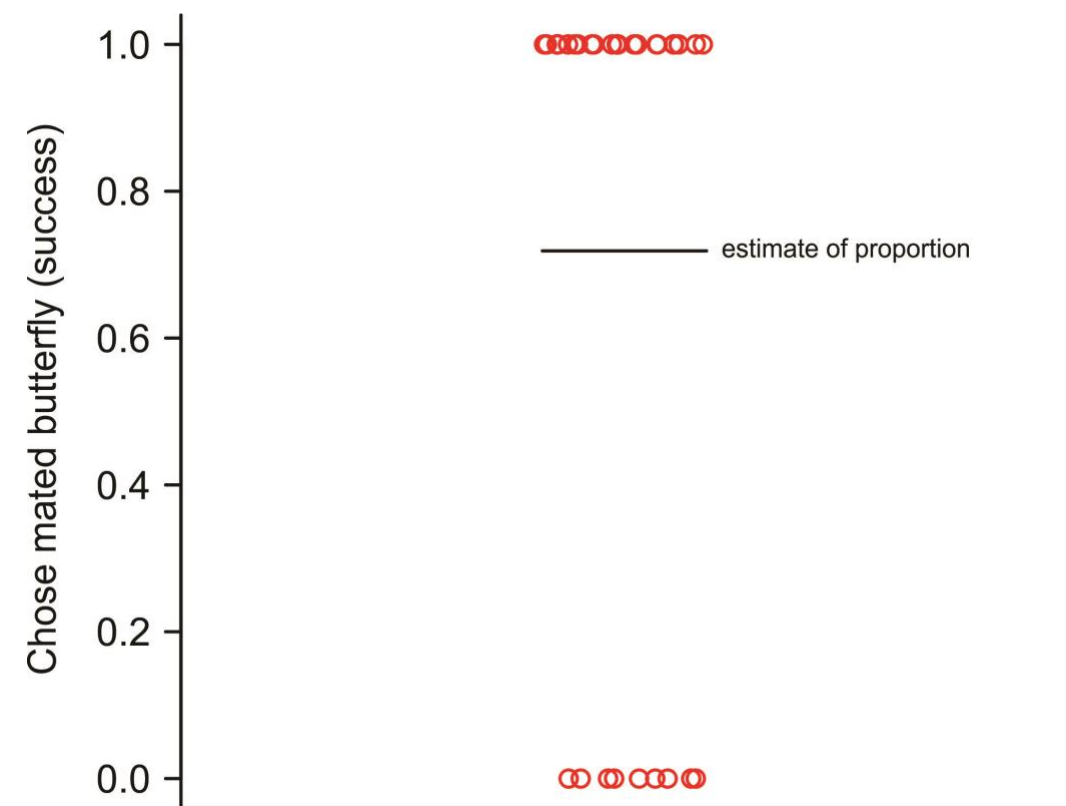
`summary(z)`

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

0.9383 is the estimate of β (the proportion on the **logit** scale). Convert back to ordinary scale (plug into inverse equation) to get the estimate of the 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. This is identical to the estimate obtained last lecture using likelihood function.



The likelihood-based method for confidence intervals is preferred

```
summary(z)
```

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

Likelihood-based 95% confidence limits:

```
myCI <- confint(z)           # limits 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 lecture for likelihood-based confidence intervals using a likelihood function (with more decimal places now).

Confidence intervals using `emmeans ()` with `glm ()` aren't likelihood-based

```
emmeans(z, specs = ~ 1)
```

1	emmean	SE	df	asympt.LCL	asympt.UCL
overall	0.938	0.393	Inf	0.168	1.71

```
emmeans(z, specs = ~ 1, type = "response")
```

1	prob	SE	df	asympt.LCL	asympt.UCL
overall	0.719	0.0795	Inf	0.542	0.847

It uses the z-value (Wald statistic) instead of the likelihood function, and is less accurate when sample size is not large.

Avoid using `summary()` for hypothesis testing

```
summary(z)
```

```
              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 less reliable than the log-likelihood ratio test.

Use `anova ()` to test hypotheses

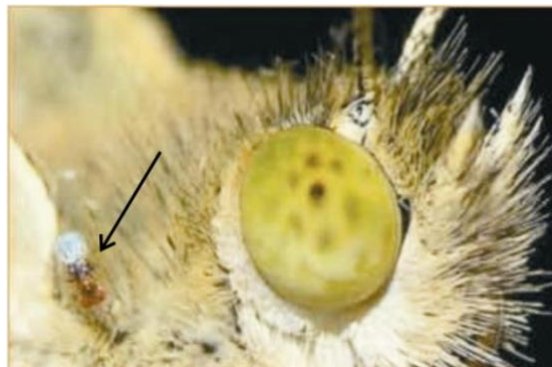
Last week we calculated the log-likelihood ratio test for these data “by hand”. Here we’ll use `glm ()` to accomplish the same task.

“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 deviance 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.

$G = 6.337$ is the identical result we obtained “by hand” using log likelihood ratio test last week.

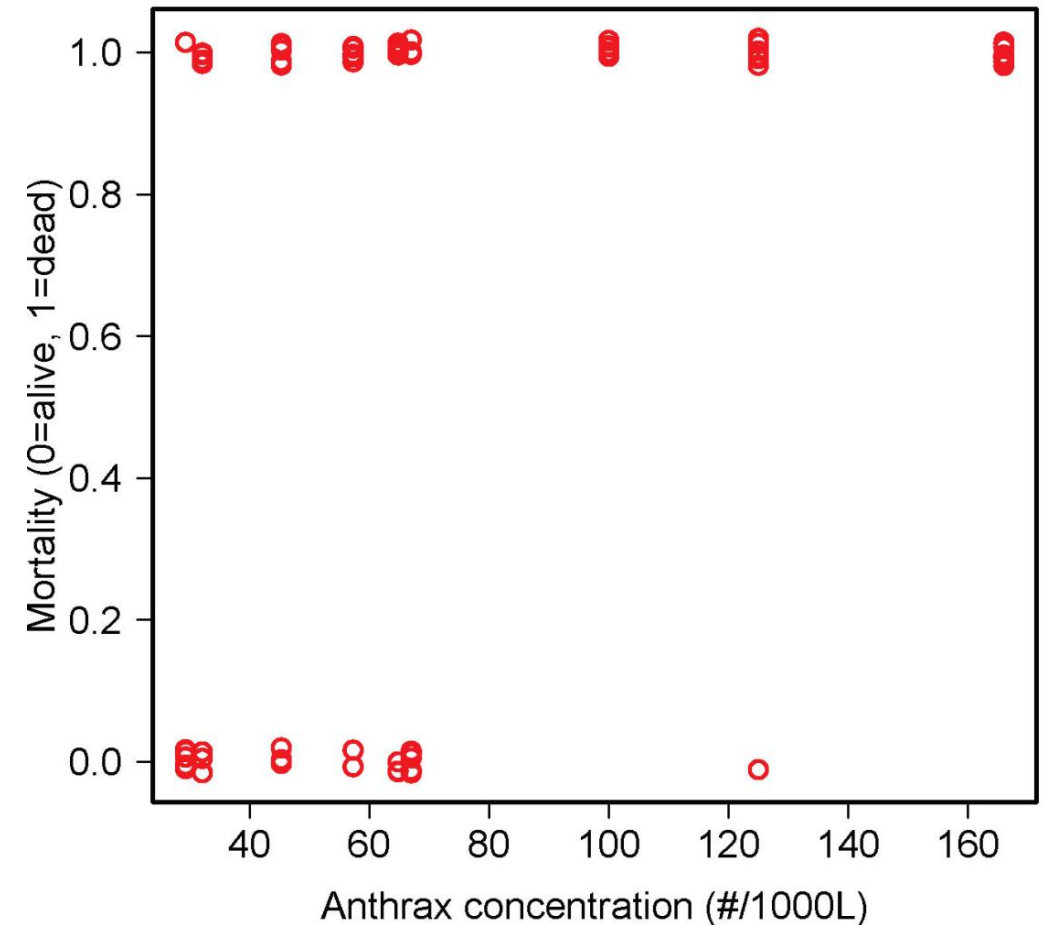
Example 2: Logistic regression

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.

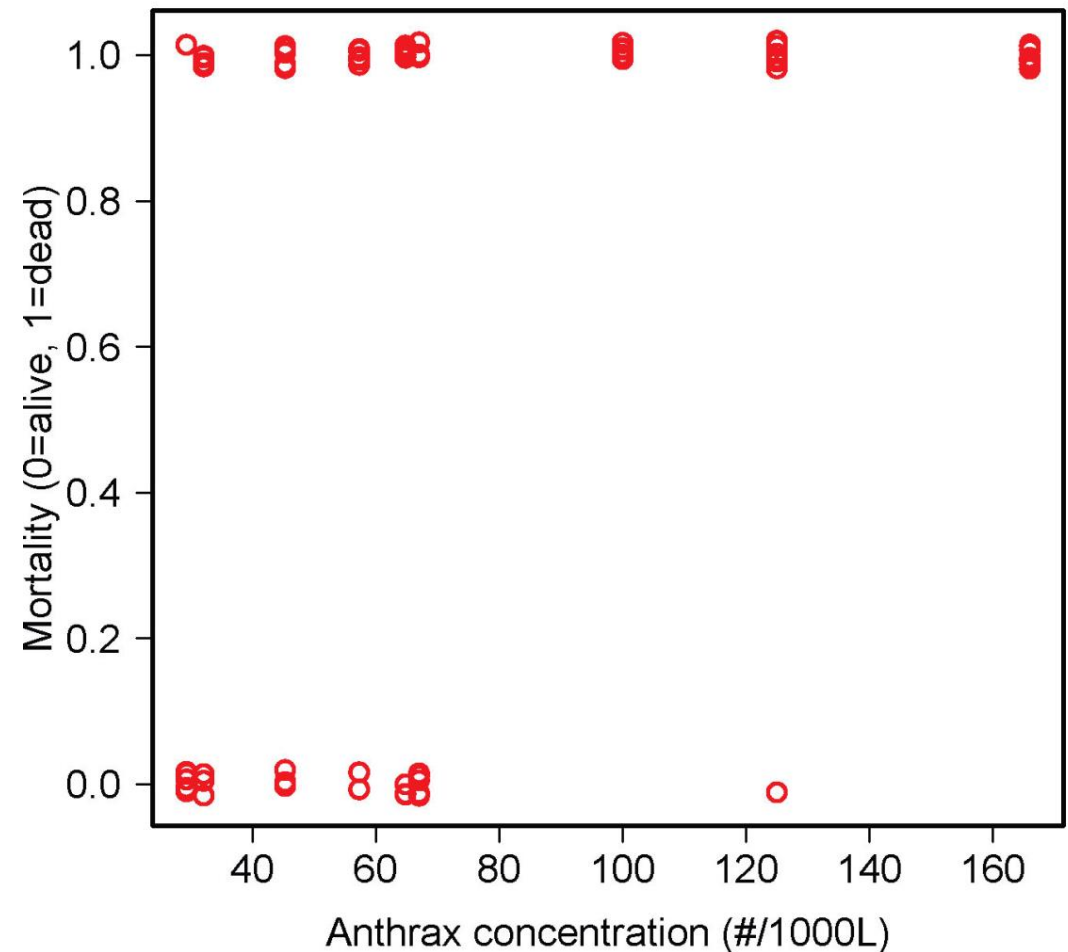


Logistic regression

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 normally distributed
- 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"))
```

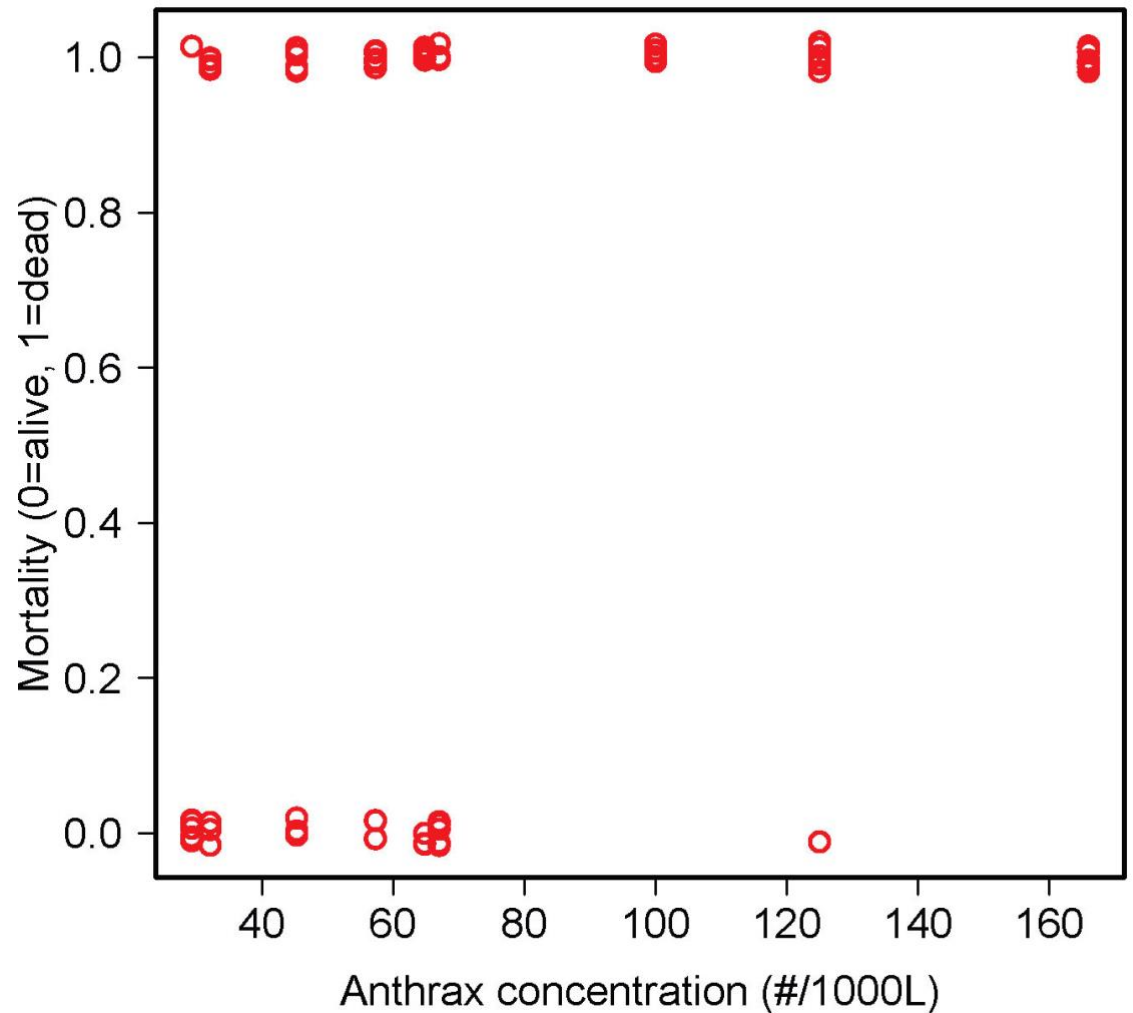
The generalized linear model

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

`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. Same idea as a grid search but more sophisticated.



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.

Use `confint()` for likelihood-based confidence intervals

```
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	**

```
confint(z, "concentration")
```

```
      2.5 %      97.5 %  
0.01749029 0.06195451
```

No value in using the inverse function to convert these limits to the untransformed proportion scale because there is no linear slope on the proportion scale.

Predicted values on logit scale

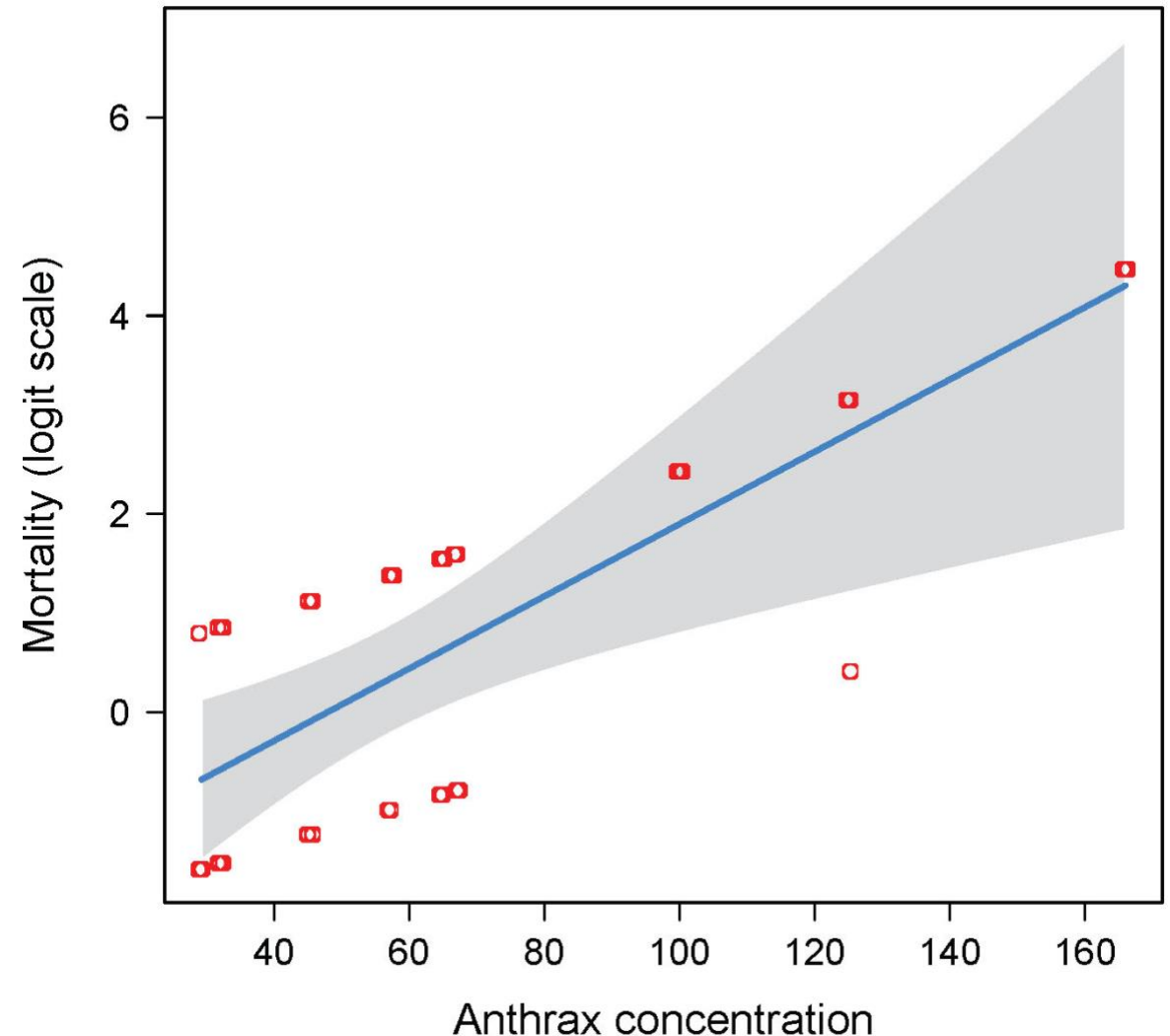
Use `predict(z)` to obtain predicted values on the logit scale

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

`visreg(z)` uses `predict()` to plot predicted values and confidence limits on the logit scale.

See that the function is a line.

The points on this scale are not the logit-transformed data. Instead, `glm()` uses residuals to create “working” values that enables fitting model to data on transformed scale.

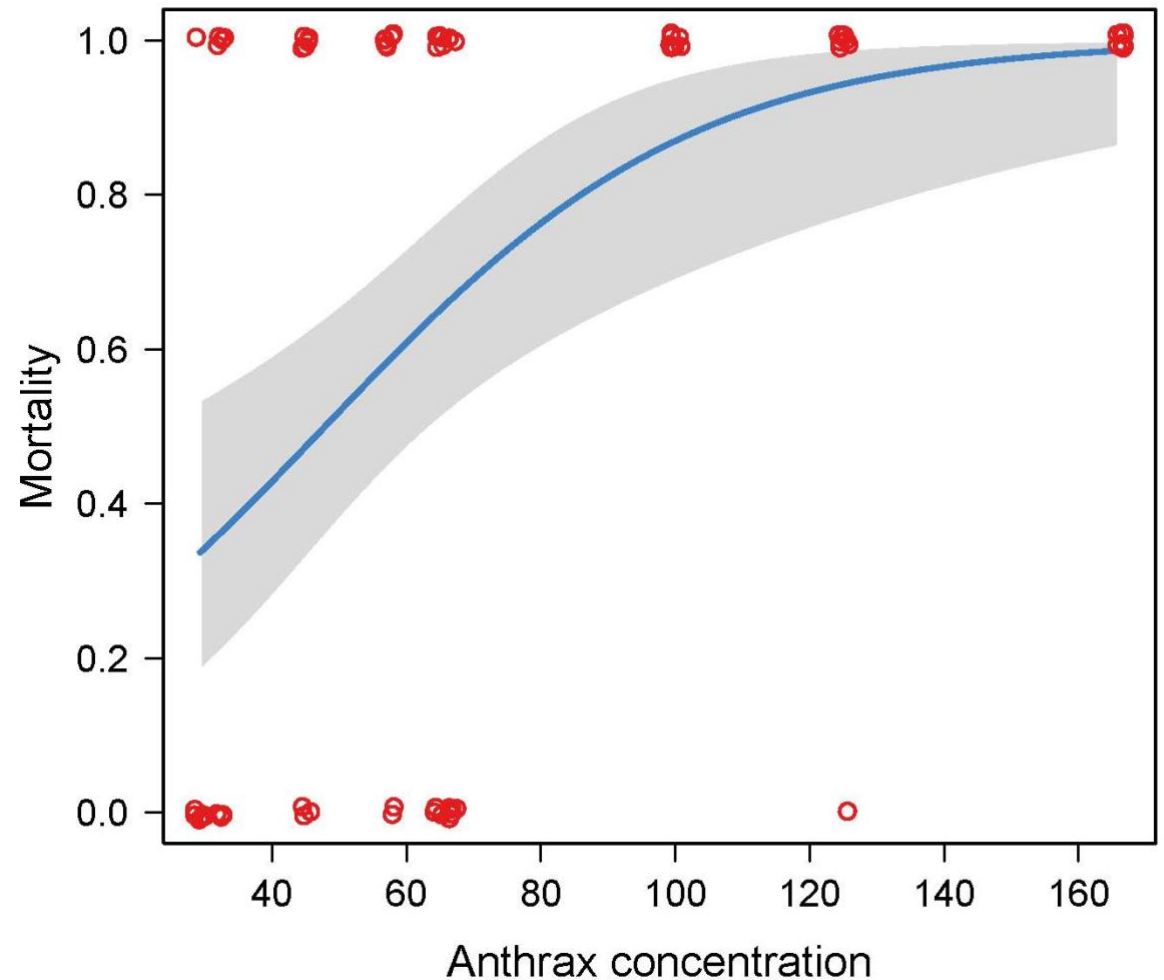


Predicted values on the original scale

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

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

`visreg(z, scale = 'response')`
gave me the fitted curve and
confidence bands on the
original proportion mortality scale.



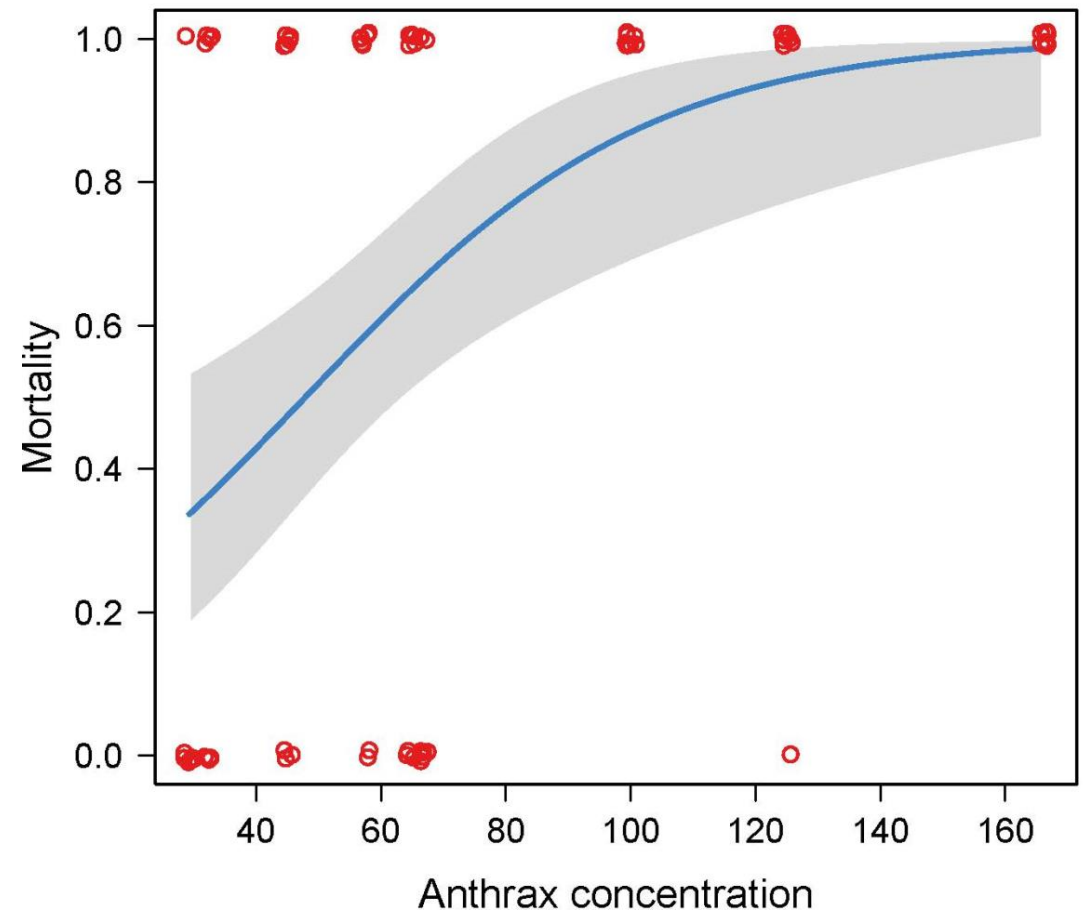
Estimate LD₅₀

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

The parameter estimates from the model fit can be used to estimate LD₅₀, 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 relationship between dose and mortality.

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

	<u>Df</u>	<u>Deviance</u>	<u>Resid</u>	<u>Df</u>	<u>Resid Dev</u>	<u>P(> Chi)</u>
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 as with `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.

Assumptions of generalized linear models







- Statistical independence of data points.
- Correct specification of the link function for the data and for the probability distribution of residuals.
- The variances of the residuals correspond to that assumed by the link function. (see below).

Analysis follows design

Hypothetical example: fish in tanks

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

Glm is appropriate (individual is the replicate)

Case 2	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)

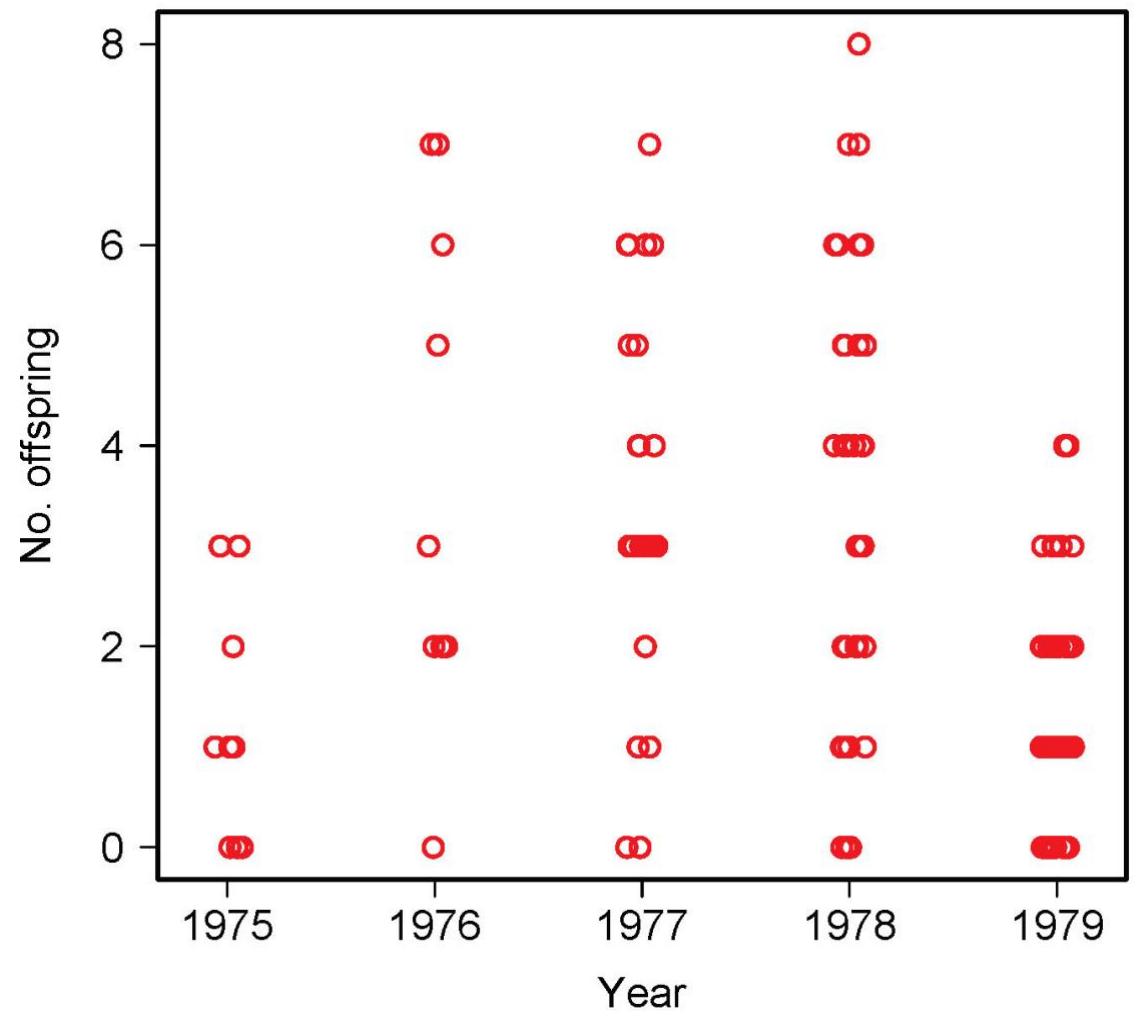
In second case, analyze the *summary statistic* per tank (fraction surviving) with `lm()`. Or, fit a generalized linear mixed models using `glmm()` in `lme4` package.

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



Example 3: Analyzing count data with log-linear regression

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

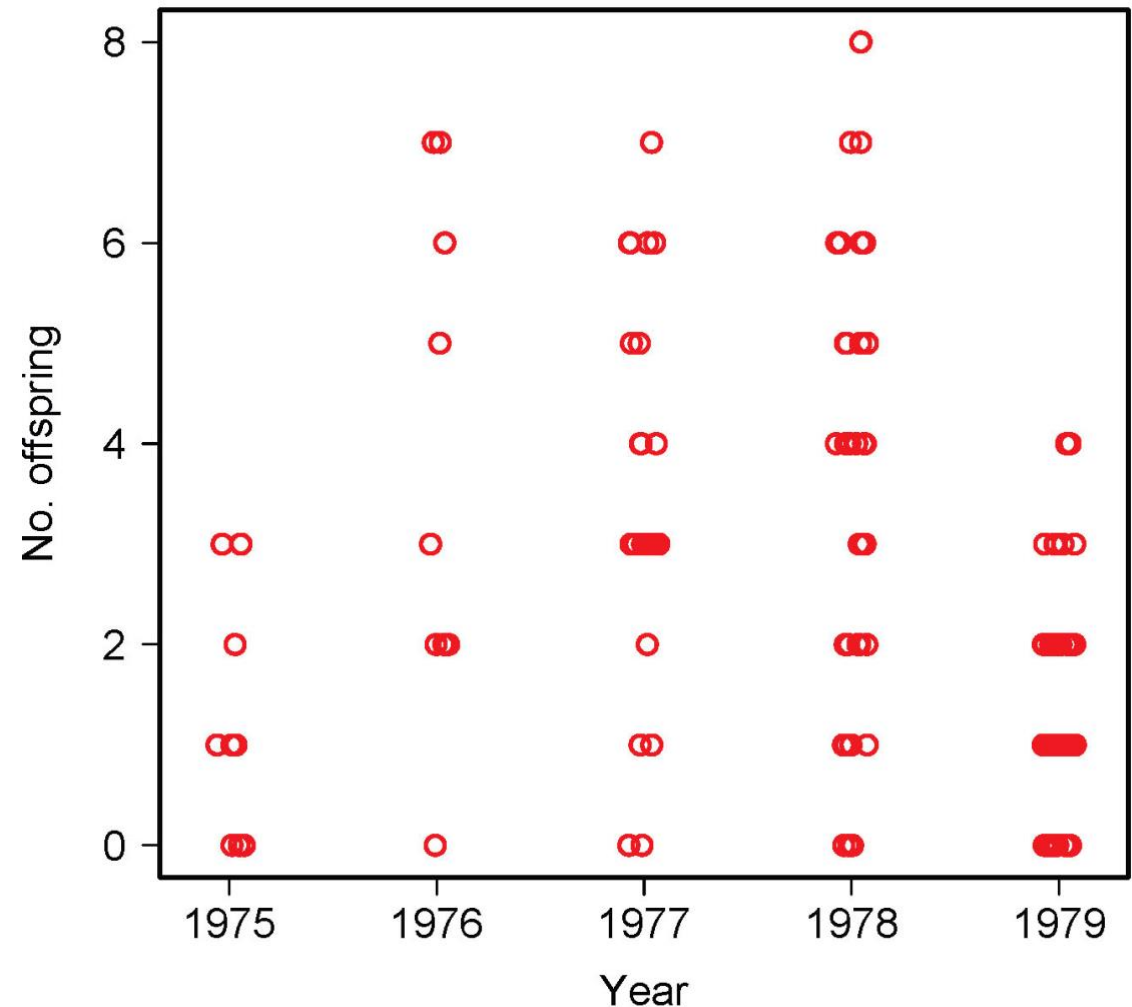
Linear model assumptions not met:

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

Two solutions:

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

2. Use generalized linear model.
Poisson distribution might be most appropriate 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

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

η is the predicted count on log scale (here, the mean), μ is the predicted count on the original 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 log scale.

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 the Poisson assumption that variance = mean (almost NEVER true – but more on this later).

Predicted values on the transformed scale

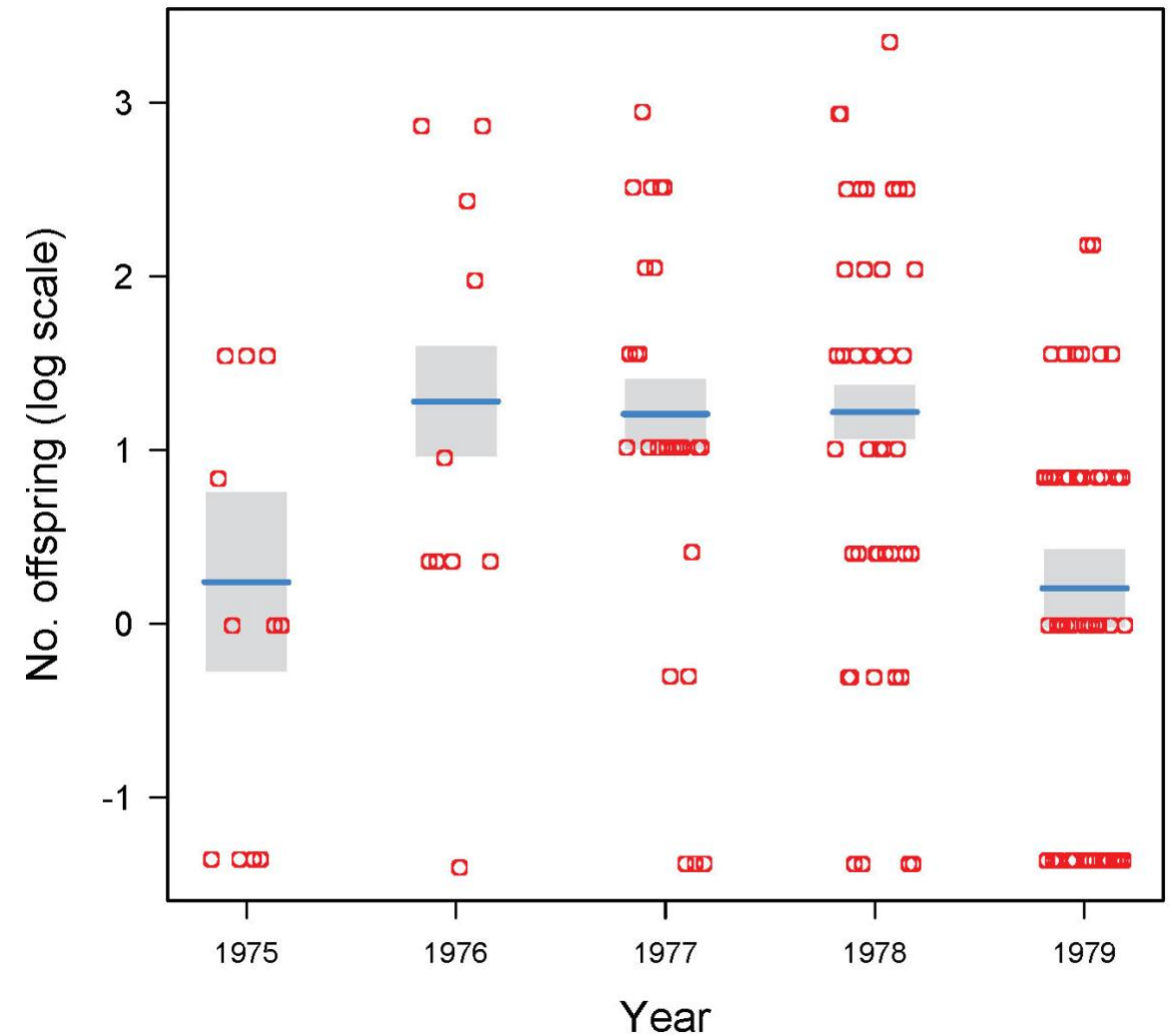
Predicted values on the log scale:

```
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).

`glm()` creates “working” values based on the residuals to fit the model to the data on the transformed scale.



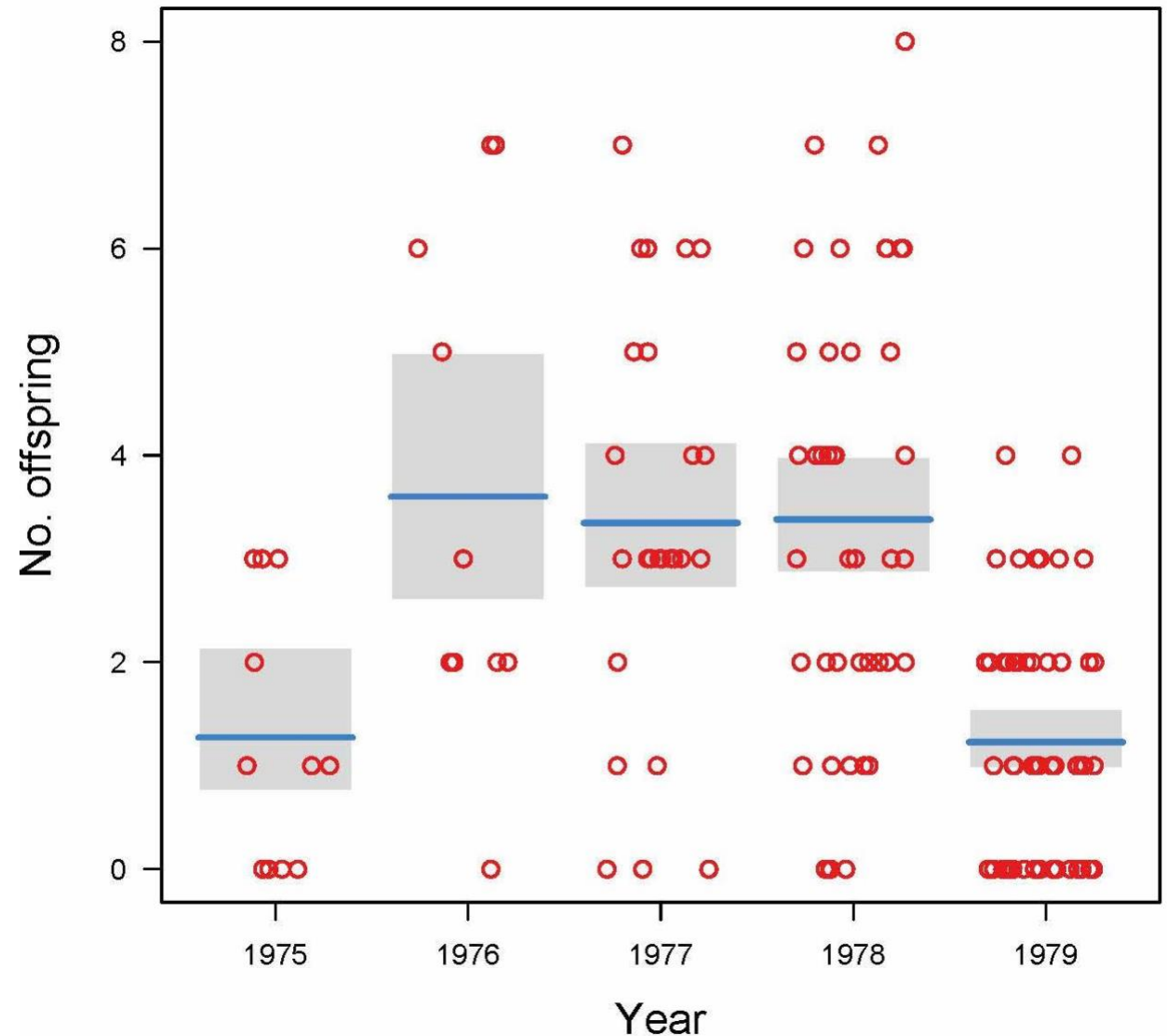
Predicted values on the original scale

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").



Confidence intervals using `emmeans()` are not likelihood-based

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

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

year	rate	SE	df	asympt.LCL	asympt.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 (this is why degrees of freedom, `df`, are shown as infinite). These confidence limits are less accurate than those produced by the likelihood-based method when sample size is not large.

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)

	<u>Df</u>	<u>Deviance</u>	<u>Resid.</u>	<u>Df</u>	<u>Resid. Dev</u>	<u>P(> Chi)</u>
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()`.

Dispersion assumption of the `glm()` method

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 (this is what it means for the `glm()` dispersion parameter = 1).

But real data in biology rarely fit this assumption.

Dispersion assumption of the `glm()` method

A central property of the Poisson distribution is that the variance and mean are equal (i.e., the `glm()` dispersion parameter = 1). But real data usually have a variance higher than the mean.

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 are generally slightly larger than means.

Modeling excessive variance

Finding excessive variance (“overdispersion”) is common when analyzing count data in biology.

Excessive variance occurs because variables not included in the model also contribute to variation in 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 allows the dispersion parameter to be estimated from the data.

The $glm()$ procedure to accomplish over (or under) dispersion uses the observed relationship between mean and variance,

$$\text{variance} = (\text{dispersion parameter}) \times \text{mean}$$

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

Modeling excessive variance

Updating the analysis of 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 **dispersion parameter** is a bit greater than 1 for these data. Typically it is much larger than 1 for count data, so I recommend using `family = quasipoisson`.

Modeling excessive variance

The point estimates are identical with those obtained using `family=poisson` instead, but the **standard errors** (and resulting confidence intervals) are wider than those based on the incorrect assumption that dispersion parameter = 1.

```
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		

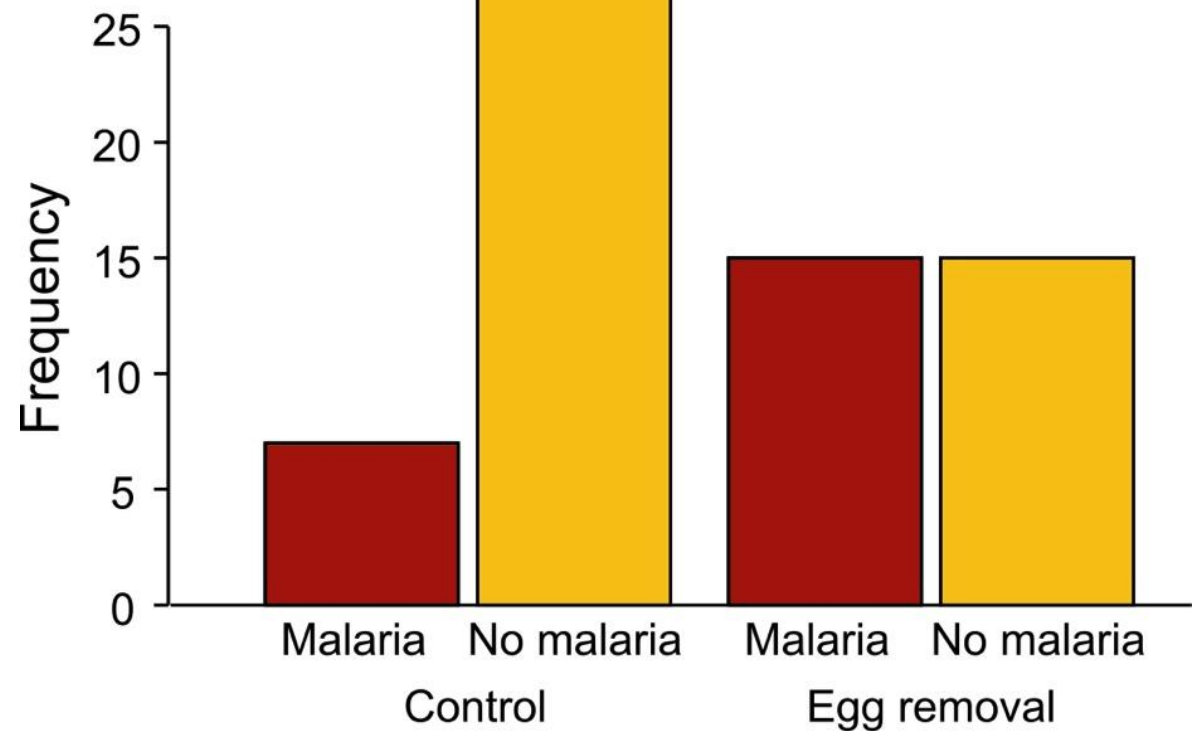
Dispersion parameter for poisson family taken to be 1

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

2x2 contingency table converted to a flat table:

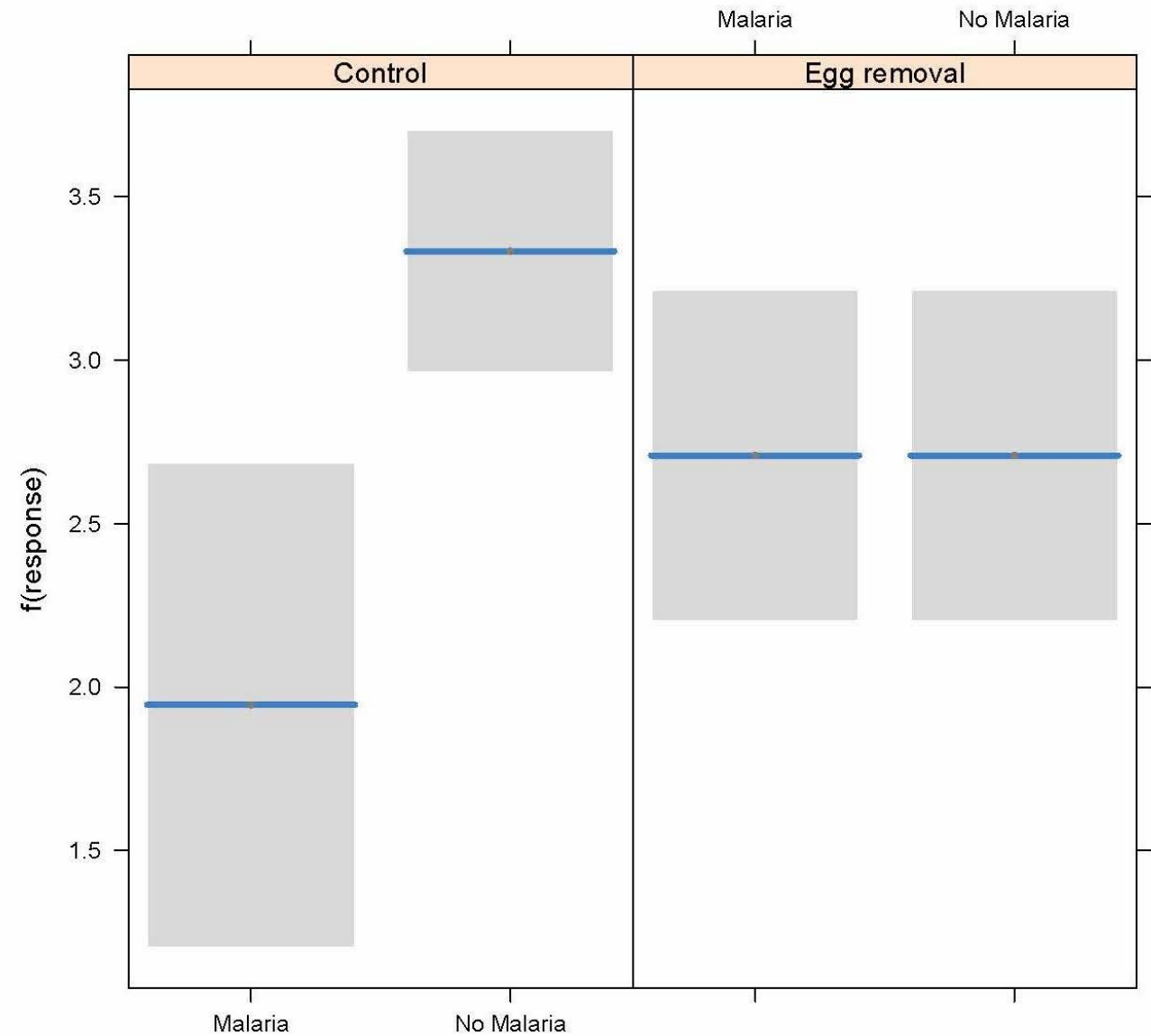
<u>Response</u>	<u>Treatment</u>	<u>Frequency</u>
Malaria	Control	7
No Malaria	Control	28
Malaria	Egg removal	15
No Malaria	Egg removal	15

```
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. If the standard assumptions of data are met (e.g., independent random trials) then `family = poisson` is appropriate.

Example 4: Modeling contingency tables

```
visreg(z, xvar="Response", by="Treatment")
```



Use `anova ()` to test hypotheses

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

	Df	Deviance	Resid.	Df	Resid. Dev	<i>P</i>
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.

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 which can be assessed separately.

`glm()` can handle data having other probability distributions than the ones used in my examples, including exponential, gamma and negative binomial 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: Kyle & Laura-Anne

Moderators: