

BISC-869, Generalized linear models

March 11, 2020

A model of the 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 terms 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 the model to data and to estimate parameters.

Use the `lm` function in R.

Simplest linear model: fit a constant (the mean)

```
lm(y~1)
```

Linear regression

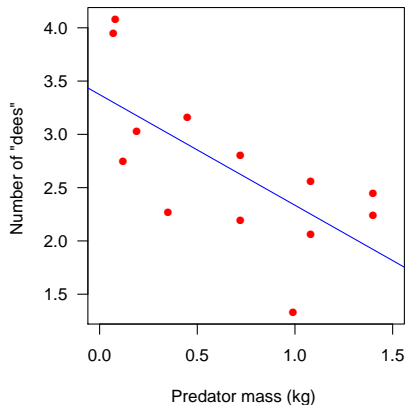
```
lm(y~x)
```

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

The predicted Y -values, denoted here as μ , are modeled as

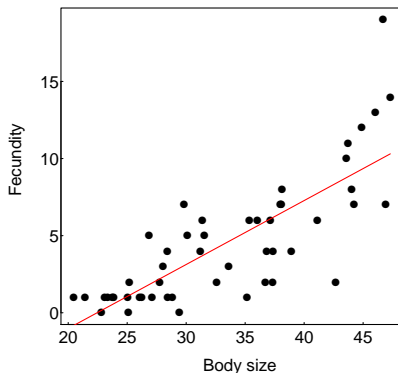
$$\mu = \beta_0 + \beta_1 X$$

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



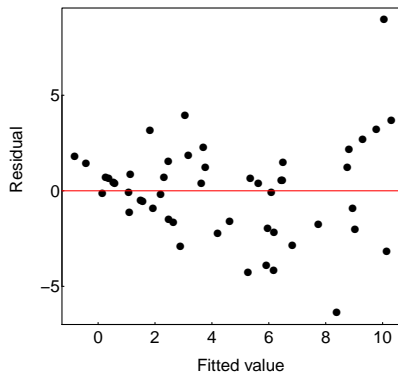
Suppose we have some (simulated) data linking body size and fecundity:

We can fit a linear model (with `lm`). What is wrong with this?



Suppose we have some (simulated) data linking body size and fecundity:

We can fit a linear model (with `lm`). What is wrong with this?



Residuals show a pattern of higher variance for larger fitted values.

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 the predicted Y-values are transformed.

$g()$ is called the “link function,” of which there are several types.

Non-normal distributions of errors are OK (these are specified by the “family” of the `glm`).

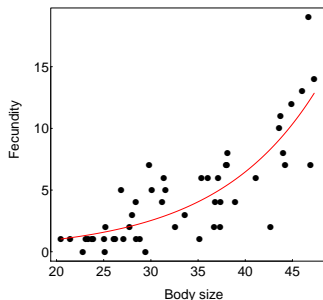
Unequal error variances OK (also specified by the “family”).

Uses maximum likelihood to estimate parameters.

Uses log-likelihood ratio tests to test parameters.

Fit models using `glm()` in R.

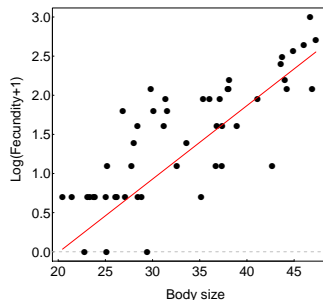
Let's return to our simulated example.



→
Transform
response via link
function (in this
case, "Log")

Fit **linear** model

←
Transform fitted
model via inverse
link (in this case
"exp")



This heuristic is not quite exactly what happens, but helps me think about what link functions are doing.

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

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

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

$g(x) = \ln(x)$ is the link function.

$g^{-1}(x) = \exp(x)$ is the inverse of link function.

The above equation could, alternatively, be written using the inverse link function as

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

2. Logistic or logit

$$\ln \left(\frac{\mu}{1 - \mu} \right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots$$

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

The link function

$$g(x) = \ln \left(\frac{x}{1 - x} \right)$$

is also known as the log-odds.

The inverse function (called “expit”) is

$$g^{-1}(x) = \text{expit}(x) = \frac{e^x}{1 + e^x}$$

The above equation could, alternatively, be written using the inverse link function as

$$\mu = \text{expit}(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots)$$



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?



The 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 : 11101110101011110101111101110011

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” and “failure”):

1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0 1 0 1 1 1 1 0 1 1 1 0 1 1 1 0 0 1 1

We can create a dataset here as follows (order doesn't matter):

```
ww <- data.frame(choice=c(rep(0,9), rep(1,23)))
```

To begin, we will fit a model with only a constant. We will use the logit link function, so our model will be:

$$\text{logit}(\mu) = \ln\left(\frac{\mu}{1-\mu}\right) = \beta$$

Here, μ refers to the population proportion, p , but we will use μ for consistency of notation.

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

The estimate of proportion is then obtained using the inverse function:

$$\hat{\mu} = \text{expit}(\hat{\beta}) = \frac{e^{\hat{\beta}}}{1 + e^{\hat{\beta}}}$$

Example: Fit a constant to 0/1 data

Formula structure is the same as when fitting a constant using `lm`.

```
out <- glm(choice~1, family=binomial(link='logit'), data=ww)
```

The `family` argument specifies the error distribution and link function.

`summary(out)`

```
Call:
glm(formula = choice ~ 1, family = binomial(link = "logit"),
    data = ww)

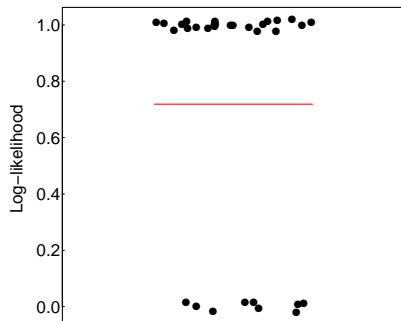
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5928 -1.5928  0.8127  0.8127  0.8127

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.9383      0.3932   2.386   0.017 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 38.024  on 31  degrees of freedom
Residual deviance: 38.024  on 31  degrees of freedom
AIC: 40.024

Number of Fisher Scoring iterations: 4
```



Why is the red line not at the value of estimate of the intercept (0.9383)?

```
summary(out)
```

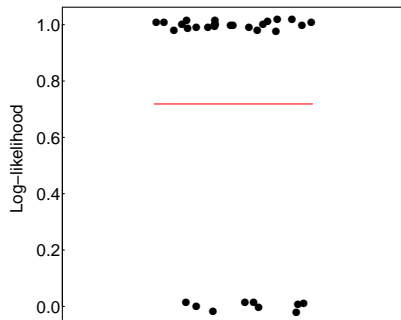
Coefficients:

	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 that has been transformed to linear scale by the **logit** function). Convert back to ordinary scale (plug into inverse equation) to get estimate of proportion:

$$\hat{\mu} = \text{expit}(\hat{\beta}) = \frac{e^{\hat{\beta}}}{1 + e^{\hat{\beta}}} = \frac{e^{0.9383}}{1 + e^{0.9383}} = 0.719$$

This is the ML estimate of the population proportion. Does it look familiar?



```
summary(out)
```

Coefficients:

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

Use `summary()` for estimation, not hypothesis testing

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. So don't use it.

95% confidence limits:

```
CI <- confint(out)
exp(CI)/(1 + exp(CI)) # inverse logit
      2.5 %      97.5 %
0.5501812  0.8535933
```

$0.550 \leq p \leq 0.853$ is the same result we obtained last week in the likelihood-based confidence interval method.

We calculated the log-likelihood ratio test for these data by hand in the likelihood lecture. Here we'll use `glm` to accomplish the same task.

“Full” model (β estimated from data):

```
out.full <- glm(choice~1, family=binomial(link='logit'))
```

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

```
out.reduced <- glm(choice~0, family=binomial(link='logit'))
```

Note: because $\text{expit}(0) = 0.5$, $\beta = 0$ tests a null hypothesis of “no preference”.

Use `anova()` to test a hypothesis about a proportion:

```
anova(out.reduced, out.full, test='Chi')
```

Analysis of Deviance Table

Model 1: choice ~ 0

Model 2: choice ~ 1

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	32	44.361			
2	31	38.024	1	6.3371	0.01182 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

Residual deviance is analogous to a residual sum of squares, and measures goodness of fit of the model to the data.

$G = 6.227$ is basically the same result we obtained with the Likelihood Ratio test last week.

Can also use `drop1` with `glms` (will do this for next example).

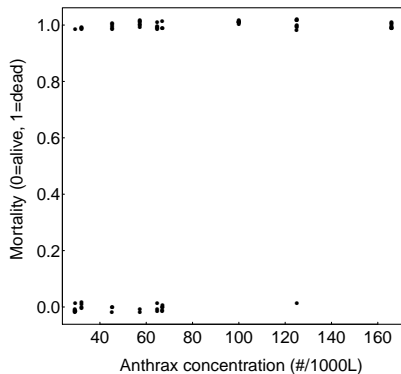
Example: Logistic regression

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*).

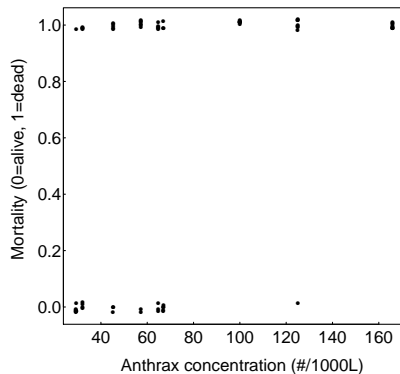
Want to estimate the relationship between dose and probability of death.



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

Ordinary *linear* regression is not appropriate because:

- For each X , the Y observations are binary, not normal.
- For each 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.



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

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

$g()$ 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.

In R:

```
out <- glm(mortality~concentration, family=binomial(link='logit'),  
           data=anthrax)
```

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

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

```
summary(out)
```

```
Call:
```

```
glm(formula = mortality ~ concentration, family = binomial(link = "logit"),  
     data = aa)
```

```
Deviance Residuals:
```

Min	1Q	Median	3Q	Max
-2.3951	-0.9161	0.3420	0.9292	1.4750

```
Coefficients:
```

	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 **

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
Null deviance: 92.982  on 71  degrees of freedom  
Residual deviance: 73.962  on 70  degrees of freedom  
AIC: 77.962
```

```
Number of Fisher Scoring iterations: 5
```

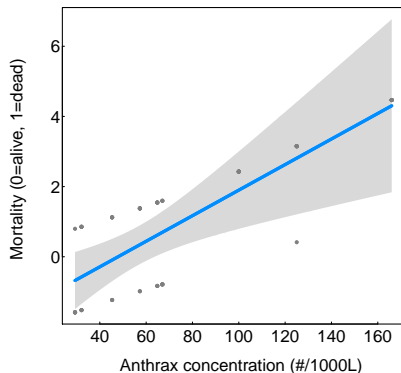
Number of Fisher Scoring iterations refers to the number of iterations used before the algorithm used by glm converged on the maximum likelihood solution.

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

$$\mu = -1.74 + 0.036x$$

`visreg(out)` uses `predict` to plot predicted values, with confidence limits in linear space.

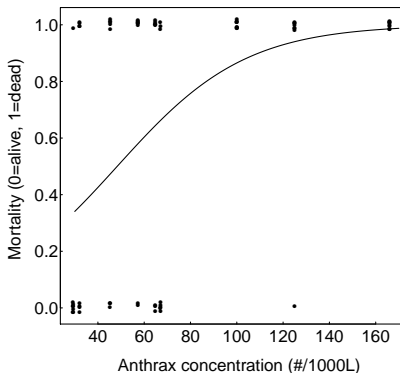
Note that the function is a line. The points on this scale are not the logit-transformed data. R creates “working” values using a transformation of the residuals from the original scale.



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

Can also plot a smooth curve using the inverse link function (in this case, `expit`):

$$\begin{aligned}\hat{\mu} &= \text{expit}(-1.74 + 0.036x) \\ &= \frac{e^{-1.74+0.036x}}{1 + e^{-1.74+0.036x}}\end{aligned}$$



In R:

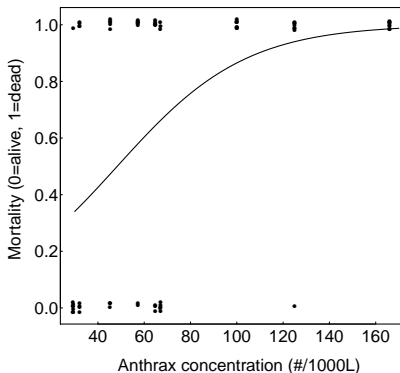
```
curve(exp(-1.74+0.036*x) / (1+exp(-1.74+0.036*x)), from=30, to=170,  
      add=TRUE)
```

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.

$$LD_{50} = -\frac{\text{intercept}}{\text{slope}} = -\frac{-1.7445}{0.03643} = 47.88$$

This can also be calculated using `dose.p` from the `MASS` package:

```
library(MASS)
dose.p(out, p=0.5)
      Dose      SE
p = 0.5: 47.8805 8.168799
```









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 (e.g., 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:	1	0	0	0	0	1
(survival)						

Glm is appropriate (individual is the replicate)

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

Glm is not appropriate (tank is the replicate)

In the second case, analyze summary statistic (fraction surviving) with `lm()`.

Mixed effects `glm` method is available in `lme4` package, but it would assume that the individuals in the same tank do not influence one another (dubious in the above hypothetical example).

Statistical independence of data points.

Correct specification of the link function for the data.

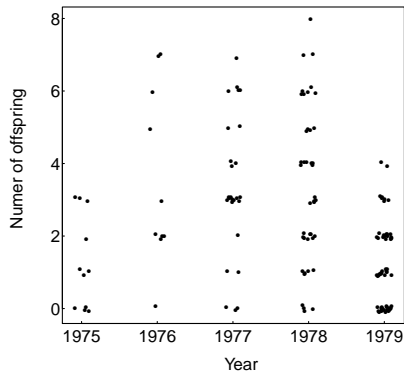
The variances of the residuals correspond to that expected from the family.

Later, I will show a method for dealing with excessive variance.

Estimate mean number of offspring fledged by female song sparrows on Mandarte Island, BC. Problem is similar to ANOVA, but ANOVA assumptions are not met.



Data are discrete counts.
Variance increases with mean.



Two solutions:

1. Transform data: $X' = \ln(X + 1)$
2. Generalized linear model. Poisson distribution might be appropriate for error distribution. So use log link function.

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

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

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

Categorical variables are modeled in R using “dummy” variables, same as with [lm](#).

```
out <- glm(noffspring~year, family=poisson(link='log'), data=ss)
summary(out)
```

Coefficients:

	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 transformed (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.

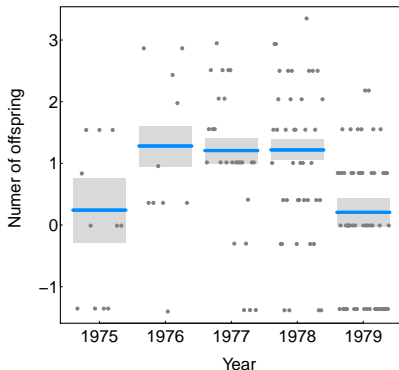
“Dispersion parameter” of 1 represents assumption that $\text{variance} = 1 \times \text{mean}$.

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

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

Notice that the “data points” on this scale are not just the transformed data - we can't calculate $\ln(0)$.

R creates “working” values using a complicated transformation of the residuals between model and data calculated on the original scale.



Predicted values on the original scale: `fitted()`

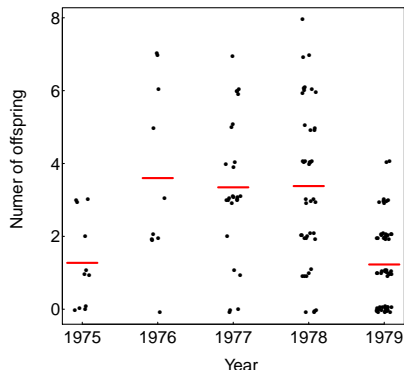
Inverse link function:

$$g^{-1}(x) = \exp(x)$$

I have plotted them here with the original data.

Note that the fitted values aren't the means of the original data.

Fitted values are the transformed values of the means that were estimated on the log scale.



Use `drop1()` 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.

```
drop1(out, test='Chisq')
```

Single term deletions

Model:

```
noffspring ~ yearF
```

	Df	Deviance	AIC	LRT	Pr(>Chi)
<none>		213.08	548.44		
yearF	4	288.66	616.01	75.575	1.506e-15 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

As with `lm`, `drop1` will only test highest order interactions for each term.

Evaluating assumptions of the glm fit

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

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

A central property of the Poisson distribution is that 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.

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(ss$noffspring, ss$year, mean)
```

1975	1976	1977	1978	1979
1.272727	3.600000	3.346154	3.380952	1.228070

```
tapply(ss$noffspring, ss$year, var)
```

1975	1976	1977	1978	1979
1.618182	6.044444	3.835385	4.680604	1.322055

Variances slightly, but not alarmingly, larger than means.

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.

Excessive variance can be accommodated in `glm` by using a different link function, one that incorporates a dispersion parameter (which must also be estimated). If the estimated dispersion parameter is $\gg 1$, 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,

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

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

```
out <- glm(noffspring~year, family=quasipoisson, data=ss)
summary(out)
```

Coefficients:

	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.

The **dispersion parameter** is reasonably close to 1 for these data, but it can be much larger than 1 for count data (in which case, you must use `family = quasipoisson`).

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

The method is commonly used to model $r \times c$ (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.

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