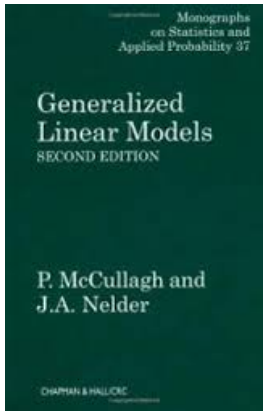
Generalized Linear Models (GLMs)



November 9 & 12, 2018

Motivation

Benefits of generalized linear models

- The residuals don't have to be normally distributed
- The response variable can be binary, integer, strictly-positive, etc...
- The variance is not assumed to be constant
- Useful for manipulative experiments or observational studies, just like linear models.

Examples

- Presence-absence studies
- Studies of survival
- Seed germination studies

Outline

Logistic regression

- The response variable is usually binary and modeled with a binomial distribution
- The probability of success is usually a logit-linear model

Poisson regression

- The response variable is a non-negative integer modeled with a Poisson distribution
- The expected count is usually modeled with a log-linear model

From linear to generalized linear

Linear model

$$\mu_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots$$
$$y_i \sim \text{Normal}(\mu_i, \sigma^2)$$

Generalized Linear model

$$g(\mu_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots$$
$$y_i \sim f(\mu_i)$$

where

g is a link function, such as the log or logit link
f is a probability distribution such as the binomial or Poisson

Alternative representations

This:

$$g(\mu_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots$$
$$y_i \sim f(\mu_i)$$

Is the same as this:

$$\mu_i = g^{-1}(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots)$$
$$y_i \sim f(\mu_i)$$

Is the same as this:

$$g(\mu_i) = \mathbf{X}\boldsymbol{\beta}$$
$$y_i \sim f(\mu_i)$$

Link functions

An inverse link function (g^{-1}) transforms values from the $(-\infty, \infty)$ scale to the scale of interest, such as $(0, 1)$ for probabilities

The link function (g) does the reverse

Link functions

Distribution	link name[1]	link equation	inverse link equation
Binomial	logit	$\log(\frac{p}{1-p})$	$\frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})}$
Poisson	log	$\log(\lambda)$	$\exp(\mathbf{X}\boldsymbol{\beta})$

Distribution	link name	link in **R**	inv link in **R**
Binomial	logit	`qlogis`	`plogis`
Poisson	log	`log`	`exp`

[1] These are the most common link functions, but others are available

Logit link example

```
beta0 <- 5
beta1 <- -0.08
elevation <- 100
(logit.p <- beta0 + beta1*elevation)

## [1] -3
```

How do we convert -3 to a probability? Use the inverse-link:

```
p <- exp(logit.p)/(1+exp(logit.p))
p

## [1] 0.04742587
```

Same as:

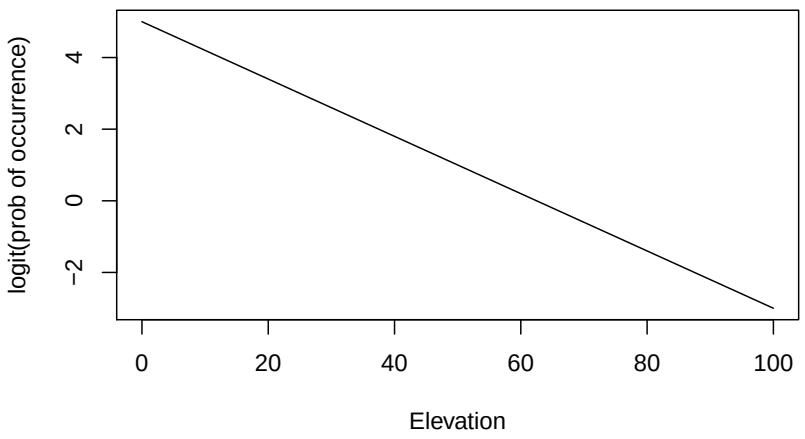
```
plogis(logit.p)

## [1] 0.04742587
```

To go back, use the link function itself:

```
log(p/(1-p))

## [1] -3

qlogis(p)

## [1] -3
```

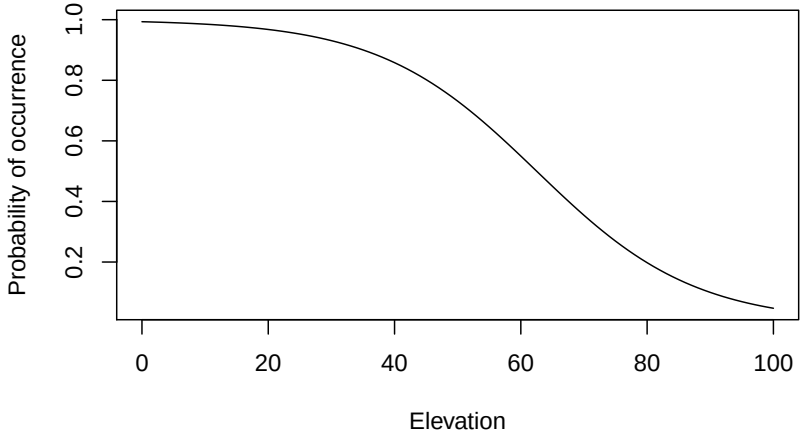
Logit link example

```
plot(function(x) 5 + -0.08*x, from=0, to=100,
     xlab="Elevation", ylab="logit(prob of occurrence)")
```



Logit link example

```
plot(function(x) plogis(5 + -0.08*x), from=0, to=100,
     xlab="Elevation", ylab="Probability of occurrence")
```



Logistic Regression

Logistic regression is a specific type of GLM in which the response variable follows a binomial distribution and the link function is the logit

It would be better to call it "binomial regression" since other link functions (e.g. the probit) can be used

Appropriate when the response is binary or a count with an upper limit

Examples:

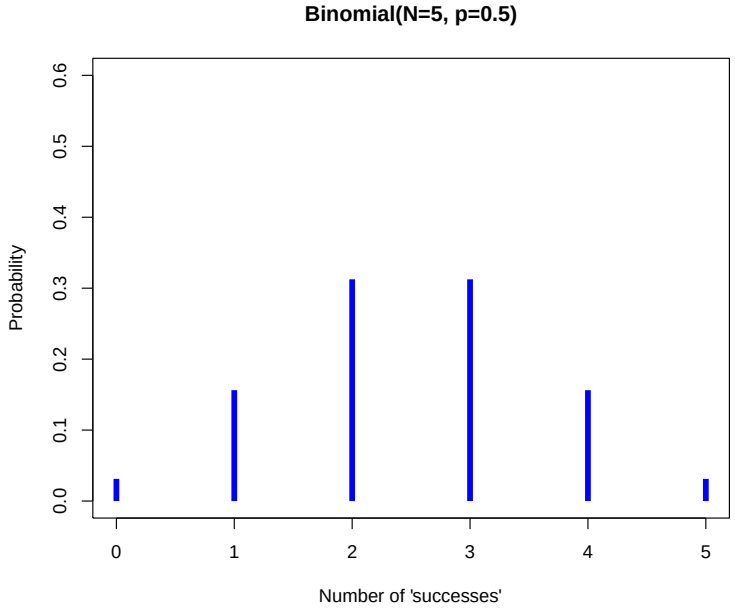
- Presence/absence studies
- Survival studies
- Disease prevalance studies

Logistic Regression

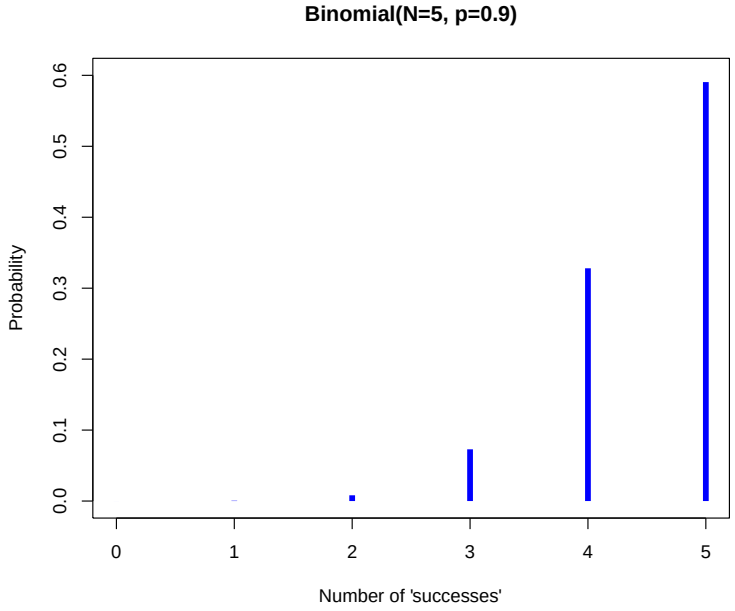
$$\text{logit}(p_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots$$
$$y_i \sim \text{Binomial}(N, p_i)$$

where:
N is the number of "trials" (e.g. coin flips)
p_i is the probability of success for sample unit i

Binomial distribution



Binomial distribution



Binomial Distribution

Properties

- The expected value of y is Np
- The variance is $Np(1-p)$

Bernoulli distribution

- The Bernoulli distribution is a binomial distribution with a single trial ($N = 1$)
- Logistic regression is usually done in this context, such that the response variable is 0/1 or No/Yes or Bad/Good, etc...

Worked example using `glm`

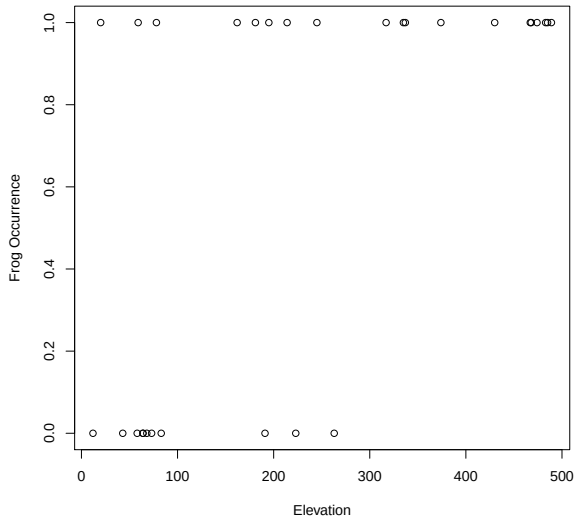
```
head(frogData, n=25)
```

```
##    presence abundance elevation habitat
## 1         0         0        58     Oak
## 2         0         3       191     Oak
## 3         0         0        43     Oak
## 4         1        15       374     Oak
## 5         1         7       337     Oak
## 6         0         0        64     Oak
## 7         1         1       195     Oak
## 8         0         1       263     Oak
## 9         1         3       181     Oak
## 10        1         3        59     Oak
## 11        1        60       489   Maple
## 12        1         9       317   Maple
## 13        0         0        12   Maple
## 14        1         4       245   Maple
## 15        1        38       474   Maple
## 16        0         0        83   Maple
## 17        1        42       467   Maple
## 18        1        52       485   Maple
## 19        1        12       335   Maple
## 20        1         1        20   Maple
## 21        1        31       430    Pine
## 22        0         1       223    Pine
## 23        0         0        68    Pine
## 24        1        47       483    Pine
## 25        1         0        78    Pine
```

First we will model the presence-absence response variable to determine if elevation and habitat affect the probability of occurrence. Then we will model abundance.

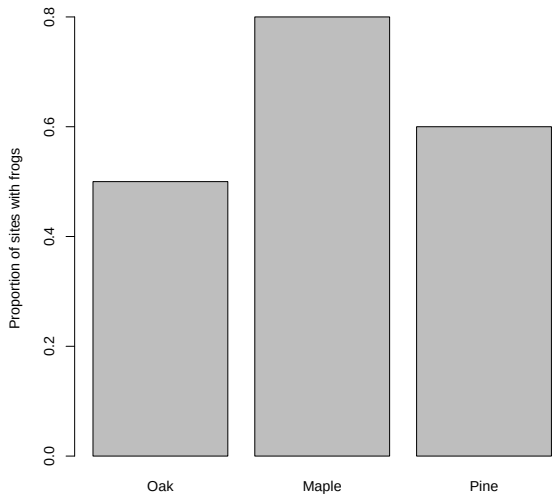
Raw data

```
plot(presence ~ elevation, frogData,
     xlab="Elevation", ylab="Frog Occurrence")
```



Raw data

```
group.prop <- tapply(frogData$presence, frogData$habitat, mean)
barplot(group.prop, ylab="Proportion of sites with frogs")
```



The function glm

```
fm1 <- glm(presence ~ habitat + elevation,
           family=binomial(link="logit"), data=frogData)
```

```
summary(fm1)
```

```
##
## Call:
## glm(formula = presence ~ habitat + elevation, family = binomial(link = "logit"),
##     data = frogData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6608  -0.7663   0.1610   0.5031   1.7773
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.053609   1.092854  -1.879   0.0602 .
## habitatMaple  1.220668   1.324680   0.921   0.3568
## habitatPine   0.281932   1.107228   0.255   0.7990
## elevation     0.011950   0.004774   2.503   0.0123 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 39.429  on 29  degrees of freedom
## Residual deviance: 25.577  on 26  degrees of freedom
## AIC: 33.577
##
## Number of Fisher Scoring iterations: 6
```


Occurrence probability and elevation

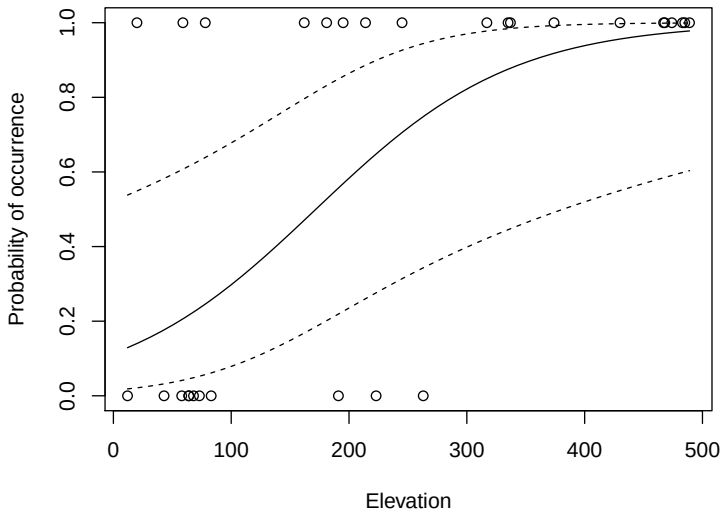
```
newdat <- data.frame(elevation=seq(12, 489, length=50),
                     habitat="Oak")
head(newdat)
```

```
##   elevation habitat
## 1  12.00000     Oak
## 2  21.73469     Oak
## 3  31.46939     Oak
## 4  41.20408     Oak
## 5  50.93878     Oak
## 6  60.67347     Oak
```

To get confidence intervals on (0,1) scale, predict on linear (link) scale and then backtransform using inverse-link

```
pred.link <- predict(fm1, newdata=newdat, se.fit=TRUE, type="link")
newdat$mu <- plogis(pred.link$fit)
newdat$lower <- plogis(pred.link$fit - 1.96*pred.link$se.fit)
newdat$upper <- plogis(pred.link$fit + 1.96*pred.link$se.fit)
```

Occurrence probability and elevation



Poisson Regression

$$\log(\lambda_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots$$
$$y_i \sim \text{Poisson}(\lambda_i)$$

where:

λ_i is the expected value of y_i

Poisson regression

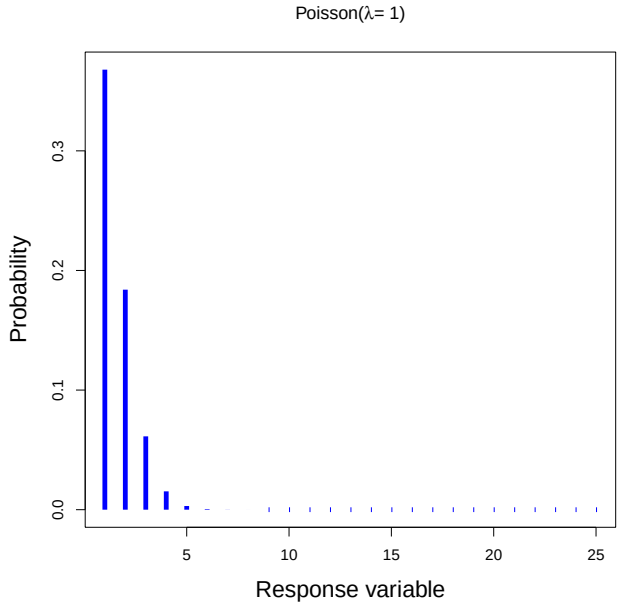
Useful for:

- Count data
- Number of events in time intervals
- Other types of integer data

Properties

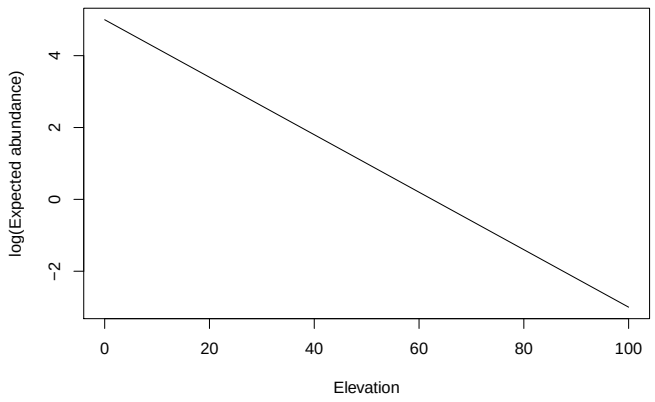
- The expected value of y (λ) is equal to the variance
- This is an assumption of the Poisson model
- Like all assumptions, it can be relaxed if you have enough data

Poisson distribution



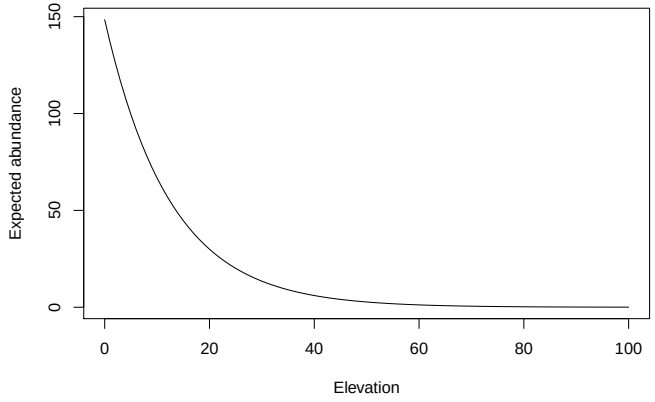
Log link example

```
plot(function(x) 5 + -0.08*x, from=0, to=100,
     xlab="Elevation", ylab="log(Expected abundance)")
```



Log link example

```
plot(function(x) exp(5 + -0.08*x), from=0, to=100,
     xlab="Elevation", ylab="Expected abundance")
```



The function glm

```
fm2 <- glm(abundance ~ habitat + elevation,
           family=poisson(link="log"), data=frogData)
```

```
summary(fm2)


##
## Call:
## glm(formula = abundance ~ habitat + elevation, family = poisson(link = "log"),
##     data = frogData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8207  -0.9818  -0.1200   0.6251   2.3868
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.284839   0.267329  -4.806 1.54e-06 ***
## habitatMaple  0.262192   0.215133   1.219    0.223
## habitatPine   0.229873   0.216865   1.060    0.289
## elevation     0.010211   0.000677  15.084  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 691.975  on 29  degrees of freedom
## Residual deviance:  28.057  on 26  degrees of freedom
## AIC: 123.21
##
## Number of Fisher Scoring iterations: 5
```

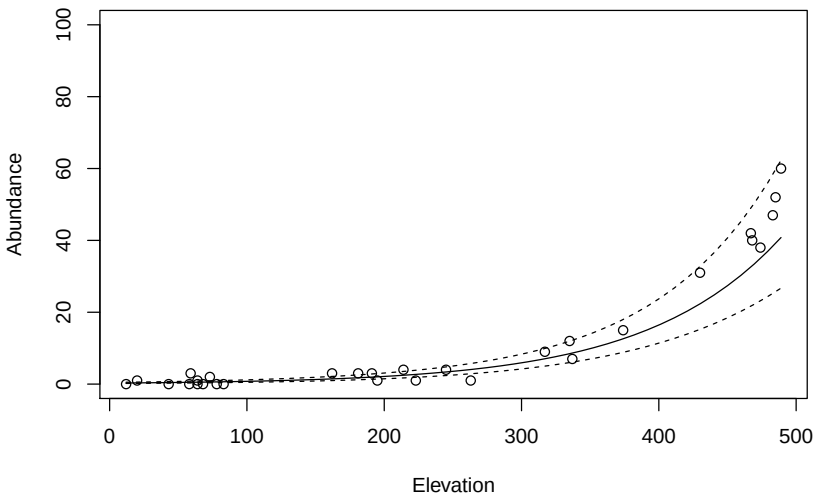

Prediction

```
newdat <- data.frame(elevation=seq(12, 489, length=50),
                     habitat="Oak")
head(newdat)

##   elevation habitat
## 1  12.00000     Oak
## 2  21.73469     Oak
## 3  31.46939     Oak
## 4  41.20408     Oak
## 5  50.93878     Oak
## 6  60.67347     Oak
```

To get confidence intervals on $(0,\infty)$ scale, predict on linear (link) scale and then backtransform using inverse-link

```
pred.link <- predict(fm2, newdata=newdat, se.fit=TRUE, type="link")
newdat$mu <- exp(pred.link$fit)
newdat$lower <- exp(pred.link$fit - 1.96*pred.link$se.fit)
newdat$upper <- exp(pred.link$fit + 1.96*pred.link$se.fit)
```

Prediction



Assessing model fit

The most common problem in Poisson regression is overdispersion.

Overdispersion is the situation in which there is more variability in the data than predicted by the model.

Overdispersion cannot be assessed by simply comparing the mean and variance of the response variable.

For example, the presence of many zeros is not necessarily indicative of overdispersion.

Overdispersion can be assessed using a goodness-of-fit test.

Goodness-of-fit

The fit of a Poisson regression can be assessed using a χ^2 test.

The test statistic is the residual deviance:

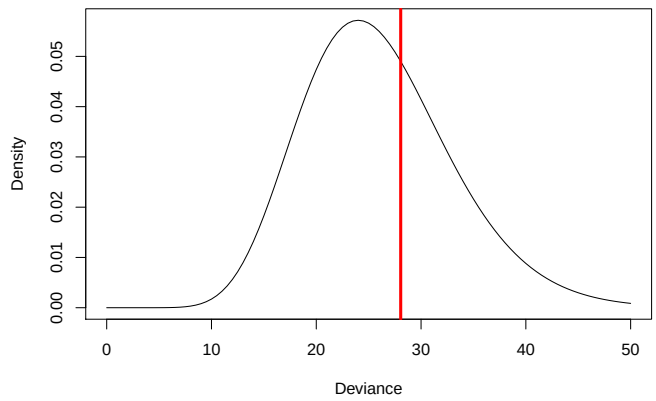
$$D = 2\left\{\sum y_i \log\left(\frac{y_i}{\hat{\lambda}_i}\right) - (y_i - \hat{\lambda}_i)\right\}$$

If the null hypothesis is true (ie, the model fits the data), D should follow χ^2 distribution with $N - K$ degrees-of-freedom.

```
N <- nrow(frogData)                    # sample size
K <- length(coef(fm2))                 # number of parameters
df.resid <- N-K                        # degrees-of-freedom
Dev <- deviance(fm2)                   # residual deviance
p.value <- 1-pchisq(Dev, df=df.resid)  # p-value
p.value                                # fail to reject H0

## [1] 0.3556428
```


χ^2 distribution and residual deviance

```
curve(dchisq(x, df=df.resid), from=0, to=50, xlab="Deviance", ylab="Density")
abline(v=Dev, lwd=3, col="red")
```



The red line is the residual deviance. We fail to reject the null hypothesis, and we conclude that the Poisson model fits the data

What if model doesn't fit the data?

Alternatives to the Poisson distribution

- Negative binomial
- Zero-inflated Poisson

Negative binomial distribution