Overdispersion in binary data

Analysis of Ecological and Environmental Data

QERM 514

Mark Scheuerell

13 May 2020

Goals for today

- Understand how to evaluate goodness-of-fit for binomial data
- Understand the notion of *overdispersion* in binomial data
- Understand the options for modeling overdispersed binomial data
- Understand the pros & cons of the modeling options

Goodness-of-fit

How well does our model fit the data?

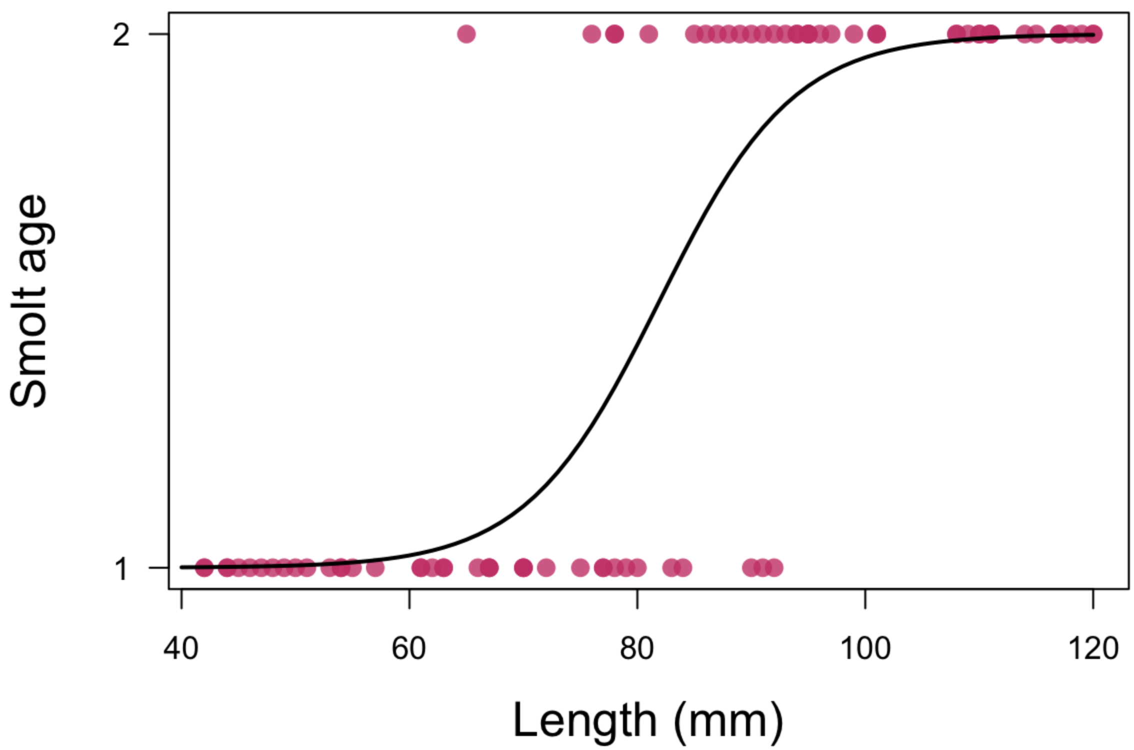
A simple check is a χ^2 test for the *standardized residuals*

$$e_i = \frac{y_i - \hat{y}_i}{\mathrm{SD}(y_i)} = \frac{y_i - \hat{y}_i}{\sqrt{(\hat{y}_i(1 - \hat{y}_i))}}$$

$$\Downarrow$$

$$\sum_{i=1}^{n} e_i \sim \chi^2_{(n-k-1)}$$

Smolt age versus length



Smolt age versus length

```
## residuals
ee <- residuals(fit_mod, type = "response")
## fitted values
y_hat <- fitted(fit_mod)
## standardized residuals
rr <- ee / (y_hat * (1 - y_hat))
## test stat
x2 <- sum(rr)
## chi^2 test
pchisq(x2, nn - length(coef(fit_mod)) - 1, lower.tail = FALSE)
```

```
## [1] 1
```

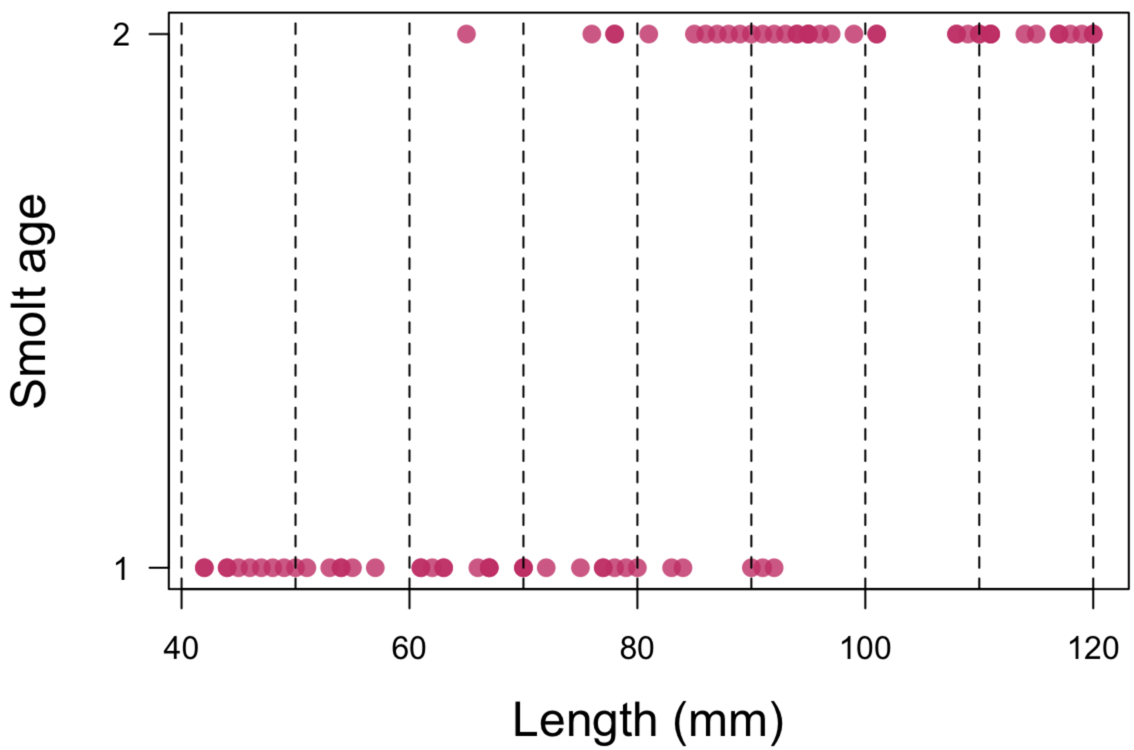
The p-value is large so we detect no lack of fit

Binned predictions

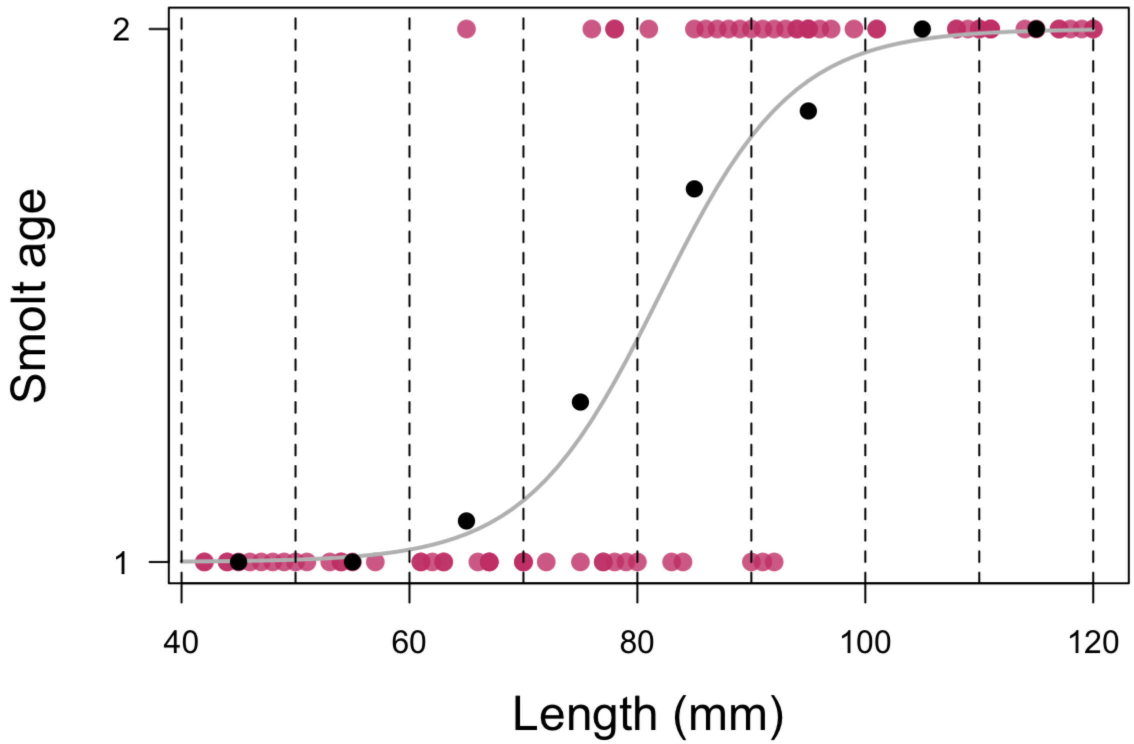
It’s hard to compare our predictions on the interval [0,1] to discrete binary outcomes {0,1}

To help, we can compute $\hat{y}$ for *bins of data*

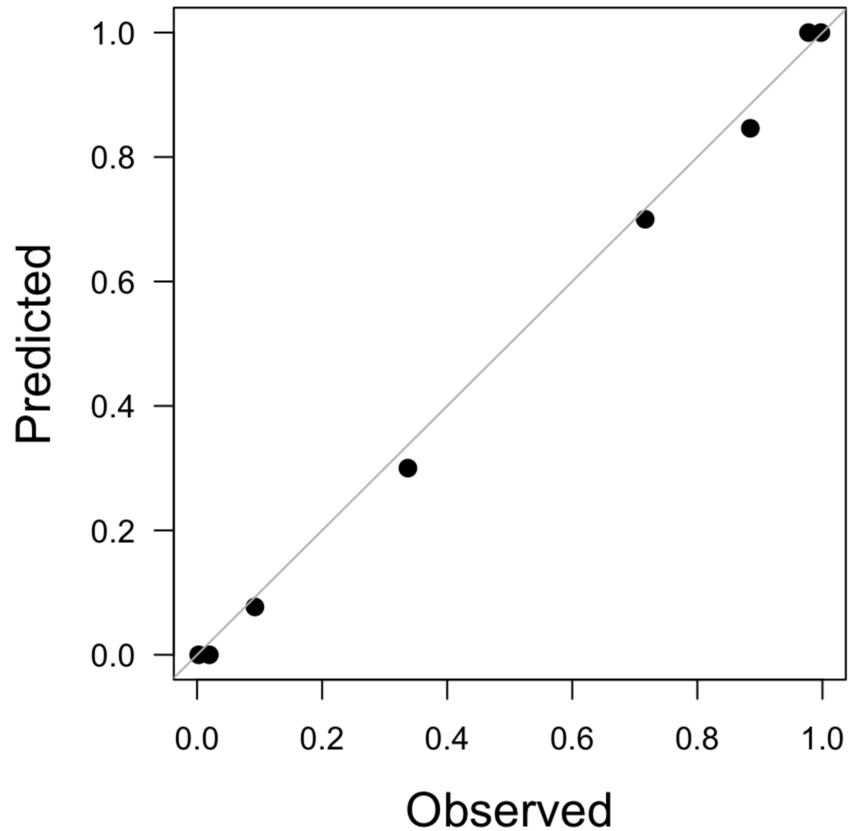
Binned predictions



Binned predictions



Binned predictions



Hosmer-Lemeshow test

We can formalize this binned comparison with the Hosmer-Lemeshow test

$$HL = \sum_{j=1}^{J} \frac{(y_j - m_j\hat{p}_J)^2}{m_j\hat{p}_J(1-\hat{p}_J)} \sim \chi^2_{(J-1)}$$

where J is the number of groups and $y_j = \sum y_{i=j}$

Hosmer-Lemeshow test

We can perform the H-L test with `generalhoslem::logitgof()`

```
## H-L test with 8 groups
generalhoslem::logitgof(obs = df$age, exp = fitted(fit_mod), g = 8)
```

```
##
##  Hosmer and Lemeshow test (binary model)
##
## data:  df$age, fitted(fit_mod)
## X-squared = 1.0998, df = 6, p-value = 0.9815
```

The p-value is large so we conclude an adequate fit

Classification scoring

Another means for evaluating goodness-of-fit is *classification scoring*

We can use our model to predict the outcome for each individual, such that

- if $p_i < 0.5$ then $\hat{y}_i = 0$
- if $p_i \geq 0.5$ then $\hat{y}_i = 1$

Classification scoring

```r
## predicted ages
pred_age <- ifelse(fitted(fit_mod) < 0.5, 1, 2)
## observed ages
obs_age = df$age + 1
## contingency table
(ct <- xtabs(~ obs_age + pred_age))
```

```
##        pred_age
## obs_age  1  2
##       1 35  5
##       2  5 35
```

```r
## correct classification
sum(diag(ct)) / nn
```

```
## [1] 0.875
```

Classification scoring

Specificity

Ability to predict age-1 when fish *do* smolt at age-1

```
##         pred_age
## obs_age  1  2
##       1 35  5
##       2  5 35
```

35 / (35 + 5) = 87.5%

Classification scoring

Sensitivity

Ability to predict age-2 when fish *do* smolt at age-2

```
##         pred_age
## obs_age  1  2
##       1 35  5
##       2  5 35
```

35 / (5 + 35) = 87.5%

Proportion of variance explained

Calculating R^2 for logistic models is not the same as linear models

Given the deviance D_M for our model and a null model D_0,

$$R^2 = \frac{1 - \exp([D_M - D_0]/n)}{1 - \exp(-D_0/n)}$$

Proportion of variance explained

Here is the R^2 for our smolt-at-age model

```
## deviances
DM <- fit_mod$deviance
D0 <- fit_mod$null.deviance
# R^2
R2 <- (1 - exp((DM - D0) / nn)) / (1 - exp(-D0 / nn))
round(R2, 2)
```

```
## [1] 0.77
```

QUESTIONS?

Lack of fit

If our model fits the data well, we expect the deviance D to be χ^2 distributed

Sometimes, however, the deviance is larger than expected

Lack of fit

What leads to a lack of fit?

- model mis-specification
- outliers
- non-linear relationship between x and η
- non-independence in the observed data

Overdispersion

Recall that the variance for a binomial of size n is given by

$$\mathrm{Var}(y) = np(1-p)$$

If $\mathrm{Var}(y) > np(1-p)$ this is called *overdispersion*

Overdispersion

Overdispersion generally arises in 2 ways related to IID errors

1. trials occur in groups & p is not constant among groups
2. trials are not independent

Overdispersion

To address overdispersion, we can include the *dispersion* parameter c, such that

$$\mathrm{Var}(y) = cnp(1-p)$$

c is also called the *variance inflation factor*

Overdispersion

We can estimate c from the deviance D as

$$\hat{c} = \frac{D}{n-k}$$

Aside: Pearson's χ^2 statistic

Pearson's χ^2 statistic is similar to the deviance

$$X^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \sim \chi^2_{(n-1)}$$

where O_i is the observed count and E_i is the expected count

Aside: Pearson's χ^2 statistic

For a binomial distribution

$$X^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

$$\Downarrow$$

$$X^2 = \sum_{i=1}^{n} \frac{(y_i - n_i\hat{p}_i)^2}{n_i\hat{p}_i(1 - \hat{p}_o)}$$

Overdispersion

We can estimate c as

$$\hat{c} = \frac{X^2}{n - k}$$

Effects on parameter estimates

The estimate of $\hat{\boldsymbol{\beta}}$ is *not* affected by overdispersion…

but the variance of $\hat{\boldsymbol{\beta}}$ *is* affected, such that

$$\text{Var}(\hat{\boldsymbol{\beta}}) = \hat{c}\left(\mathbf{X}^\top \hat{\mathbf{W}} \mathbf{X}\right)^{-1}$$

$$\mathbf{W} = \begin{bmatrix} y_1 & 0 & \dots & 0 \\ 0 & y_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & y_n \end{bmatrix}$$

Elk in clear cuts

Elk are known to use clear cuts for browsing

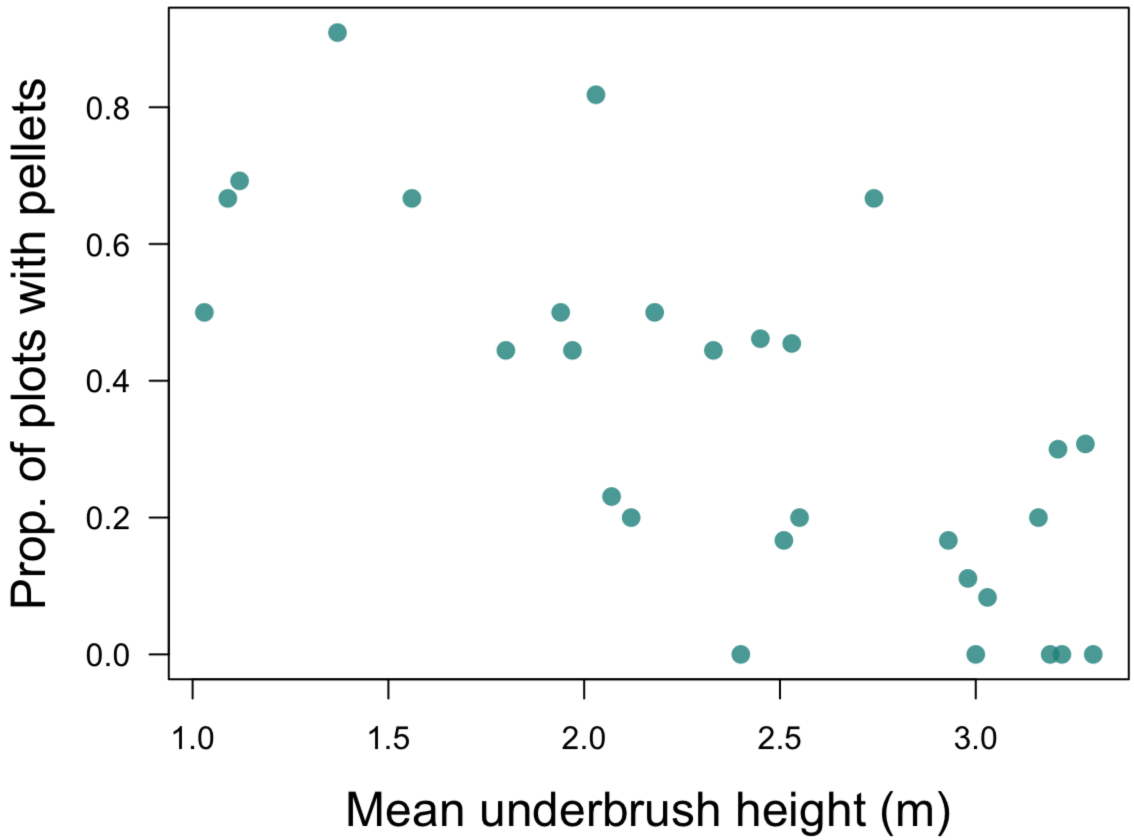
In general, the probability of finding elk decreases with height of underbrush

Elk in clear cuts

Consider an observational study to estimate the probability of finding elk as a function of underbrush height

- 29 forest sections were sampled for elk pellets along line transects
- mean height of underbrush recorded for each section
- presence/absence of pellets recorded at 9-13 points per transect

Elk in clear cuts



Elk in clear cuts

A glimpse of the pellet data

```
##    veg_height plots pellets
## 1        3.30     9       0
## 2        2.53    11       5
## 3        1.03    10       5
## 4        1.12    13       9
## 5        3.00    11       0
## 6        2.03    11       9
## 7        2.93    12       2
## 8        2.40    10       0
## 9        3.16    10       2
## 10       2.45    13       6
## 11       3.21    10       3
## 12       2.74    12       8
```

Elk in clear cuts

```r
## fit model with glm
elk_mod <- glm(cbind(pellets, plots - pellets) ~ veg_height, data = df,
               family = binomial(link = "logit"))
faraway::sumary(elk_mod)
```

```
##             Estimate Std. Error z value  Pr(>|z|)
## (Intercept)  2.40035    0.46838  5.1248 2.978e-07
## veg_height  -1.29583    0.19885 -6.5165 7.195e-11
##
## n = 29 p = 2
## Deviance = 60.28535 Null Deviance = 110.19068 (Difference = 49.90534)
```

Elk in clear cuts

```r
## original fit
faraway::sumary(elk_mod)
```

```
##              Estimate Std. Error z value  Pr(>|z|)
## (Intercept)  2.40035    0.46838  5.1248 2.978e-07
## veg_height  -1.29583    0.19885 -6.5165 7.195e-11
##
## n = 29 p = 2
## Deviance = 60.28535 Null Deviance = 110.19068 (Difference = 49.90534)
```

```r
## overdispersion parameter
c_hat <- deviance(elk_mod) / (nn- 1)
## re-scaled estimates
faraway::sumary(elk_mod, dispersion = c_hat)
```

```
##              Estimate Std. Error z value  Pr(>|z|)
## (Intercept)  2.40035    0.68726  3.4926 0.0004783
## veg_height  -1.29583    0.29178 -4.4411  8.95e-06
##
## Dispersion parameter = 2.15305
## n = 29 p = 2
## Deviance = 60.28535 Null Deviance = 110.19068 (Difference = 49.90534)
```

Quasi-AIC

For binomial models with overdispersion, we can modify AIC

$$AIC = 2k - 2\log\mathcal{L}$$

to be a *quasi*-AIC

$$QAIC = 2k - 2\frac{\log\mathcal{L}}{\hat{c}}$$

Elk in clear cuts

Model selection results

```
##                       k neg-LL   AIC deltaAIC QAIC deltaQAIC
## intercept + slope   2   61.3 126.6      0.0 60.9       0.0
## intercept only      1   86.2 174.5     47.9 82.1      21.2
```

Quasi-binomial models

When the data are overdispersed, we can relate the mean and variance of the response to the linear predictor *without* additional information about the binomial distribution

However, this creates problems when we want to make inference via hypothesis tests or CI's

Quasi-likelihood

So far we have been using likelihood methods for known distributions

Without a formal distribution for the data, we can use a *quasi-likelihood*

Quasi-likelihood

Recall that for many distributions we use a *score* (U) as part of the log-likelihood, which can be thought of as

$$U \approx \frac{(\text{observation} - \text{expectation})}{\text{scale}}$$

Quasi-likelihood

Recall that for many distributions we use a *score* (U) as part of the log-likelihood, which can be thought of as

$$U = \frac{(\text{observation} - \text{expectation})}{\text{scale}}$$

For example, a normal distribution has a score of

$$U_i = \frac{(y_i - \mu)^2}{2\sigma^2}$$

Quasi-likelihood

Let’s define the following score

$$U_i = \frac{(y_i - \mu_i)^2}{\sigma^2 V(\mu_i)}$$

$$\Downarrow$$

$$\text{mean}(U) = 0$$

$$\text{Var}(U) = \frac{1}{\sigma^2 V(\mu_i)}$$

where $V(\mu)$ is a function of the covariates

Quasi-likelihood

We now define Q_i to be integral over all possible y_i and μ_i

$$Q_i = \int_{y_i}^{\mu_i} \frac{(y_i - z)^2}{\sigma^2 V(z)} dz$$

which behaves like a log-likelihood function, such that the *quasi-likelihood* for all n is

$$Q = \sum_{i=1}^{n} Q_i$$

Quasi-likelihood

We can estimate $\boldsymbol{\beta}$ by maximizing Q as with other distributions

But we need to estimate σ^2 separately as

$$\sigma^2 = \frac{X^2}{n-k}$$

where X^2 are the Pearson residuals as defined on slide #26

Elk in clear cuts

Fitting a quasi-binomial model

```
## quasi-binomial
elk_quasi <- glm(cbind(pellets, plots - pellets) ~ veg_height, data = df,
                 family = quasibinomial)
faraway::sumary(elk_quasi)
```

```
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  2.40035    0.65694  3.6538  0.001097
## veg_height  -1.29583    0.27891 -4.6461 7.884e-05
##
## Dispersion parameter = 1.96723
## n = 29 p = 2
## Deviance = 60.28535 Null Deviance = 110.19068 (Difference = 49.90534)
```

Elk in clear cuts

```
## quasi-binomial
faraway::sumary(elk_quasi)
```

```
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  2.40035    0.65694  3.6538  0.001097
## veg_height  -1.29583    0.27891 -4.6461 7.884e-05
##
## Dispersion parameter = 1.96723
## n = 29 p = 2
## Deviance = 60.28535 Null Deviance = 110.19068 (Difference = 49.90534)
```

```
## variance inflation
faraway::sumary(elk_mod, dispersion = c_hat)
```

```
##             Estimate Std. Error z value  Pr(>|z|)
## (Intercept)  2.40035    0.68726  3.4926 0.0004783
## veg_height  -1.29583    0.29178 -4.4411  8.95e-06
##
## Dispersion parameter = 2.15305
## n = 29 p = 2
## Deviance = 60.28535 Null Deviance = 110.19068 (Difference = 49.90534)
```

Beta-binomial models

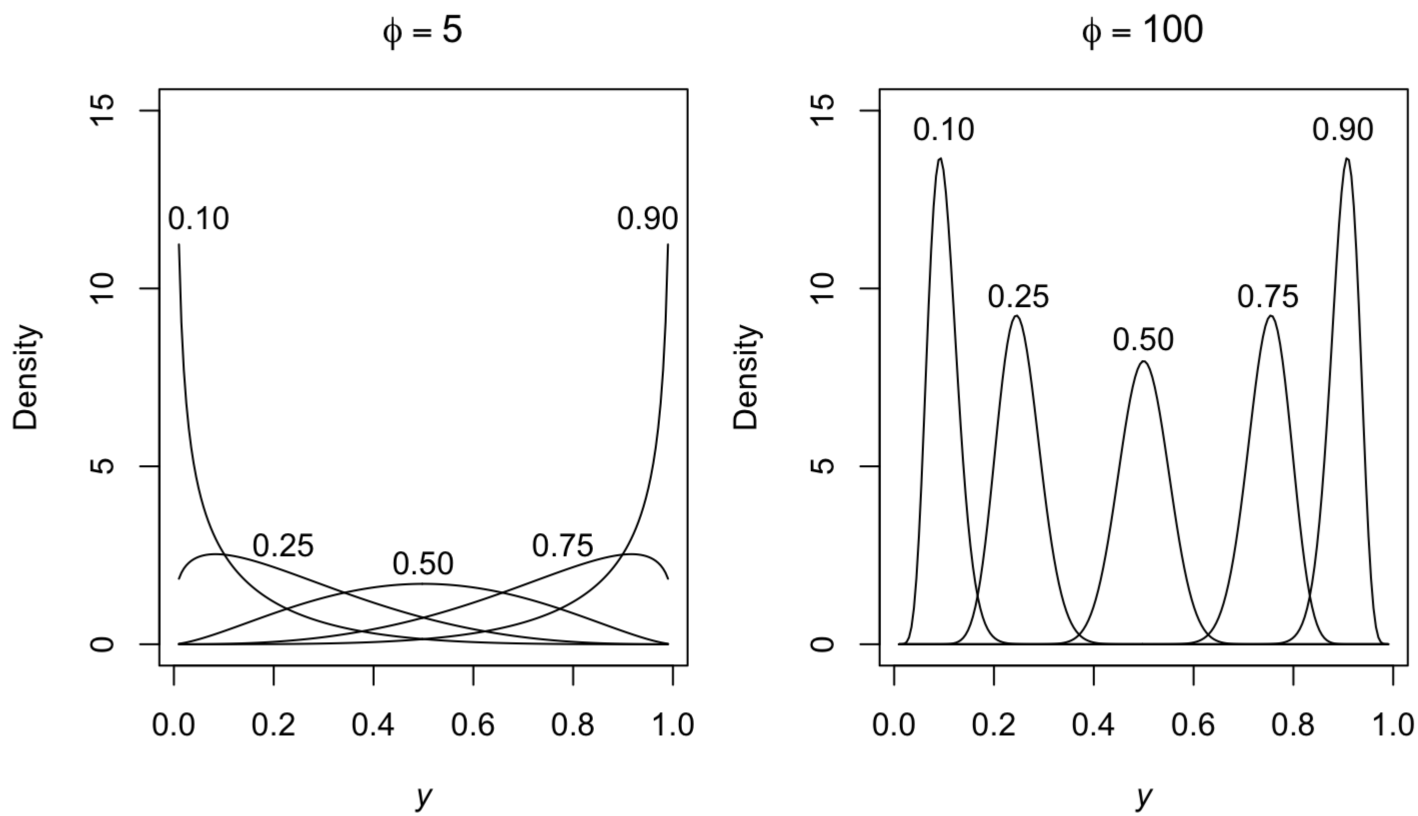
Another option for binomial data is the beta distribution

$$f(y; \mu, \phi) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma((1-\mu)\phi)} y^{\mu\phi - 1}(1-y)^{(1-\mu)\phi - 1}$$

with

$$\text{mean}(y) = \mu$$
$$\text{Var}(y) = \frac{\mu(1-\mu)}{1+\phi}$$

Beta-binomial models



Beta-binomial models

We can use `gam()` from the **mgcv** package to fit beta-binomial models

```
## load mgcv
library(mgcv)
## `gam()` needs proportions for the response
df$prop <- df$pellets / df$plots
## weight by num of plots per section
wts <- df$plots / sum(df$plots)
## fit model
elk_betabin <- gam(prop ~ veg_height, weights = wts, data = df,
                   family = betar(link = "logit"))
```

Beta-binomial models

```
## inspect beta-binomial fit
summary(elk_betabin)
```

```
##
## Family: Beta regression(0.947)
## Link function: logit
##
## Formula:
## prop ~ veg_height
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.496      4.381   0.570    0.569
## veg_height    -1.539      1.714  -0.898    0.369
##
##
## R-sq.(adj) =   0.48   Deviance explained = -38.5%
## -REML = -6.2244  Scale est. = 1         n = 29
```

Summary

There are several ways to model overdispersed binomial data, each with its own pros and cons

Model	Pros	Cons
binomial	Easy	Underestimates variance
binomial with VIF	Easy; estimate of variance	Ad hoc
quasi-binomial	Easy; estimate of variance	No distribution for inference
beta-binomial	Strong foundation	Somewhat hard to implement