# Class 6: Generalised linear mixed models

Andrew Parnell
andrew.parnell@mu.ie

Maynooth University
National University of Ireland Maynooth

# Learning outcomes

- Understand the basics of a glmm
- See a few different examples of glmms
- Understand how to fit basic glmms in `lme4`

# Revision: generalised linear models

- Recall that the normal linear model has residuals which are normally distributed and a response variable that is conditionally normally distributed
- e.g. $y_i = \alpha + \beta x_i + \epsilon_i$ with $\epsilon_i \sim N(0, \sigma^2)$ and $y_i | x_i \sim N(\alpha + \beta x_i, \sigma^2)$
- In a generalised linear model the residuals often don't exist and the response variable has a non-normal conditional distribution, e.g. Binomial, Poisson or Gamma (among many many others)
- The key to a glm is that the mean of the conditional distribution is transformed in a clever way via a *link* function, and the transformed mean is given a standard linear relationship with the covariates

# Glm example

- Example 1: Binomial

$$y_i|x_i \sim Bin(N, p_i);\ logit(p_i) = \alpha + \beta x_i$$

- Example 2: Poisson

$$y_i|x_i \sim Po(\lambda_i);\ \log(\lambda_i) = \alpha + \beta x_i$$

- Example 3: Negative Binomial

$$y_i|x_i \sim NegBin(\phi, p_i);\ logit(p_i) = \alpha + \beta x_i$$

# Some important notes about glms

- Sometimes you have the choice as to how you collect your data (e.g. collecting precise numbers as opposed to yes/no values). When you have this choice, it is almost always preferable to collect the precise values as these will give you more precise results later on. Estimating the values of $\alpha$ and $\beta$ in e.g. a Binomial glm is often much less precise than estimating them in a normal linear model
- There are lots of different link functions with no strong guidance as to which one you should choose. For example some people use *probit* instead of logit for Binomial models, and some people don't use any link function at all
- In the frequentist world, the fitting method becomes even more complicated when dealing with glms, often using something called Iteratively Re-weighted Least Squares (IRLS) which you might see referred to by `lme4`

# Example: the swiss willow tit data

```
swt = read.csv('../data/swt.csv')
head(swt)
```

```
##   rep.1 rep.2 rep.3 c.2 c.3 elev forest dur.1 day.2 day.3 length alt
## 1     0     0     0   0   0  420      3   240    58    73    6.2 Low
## 2     0     0     0   0   0  450     21   160    39    62    5.1 Low
## 3     0     0     0   0   0 1050     32   120    47    74    4.3 Med
## 4     0     0     0   0   0 1110     35   180    44    71    5.4 Med
## 5     0     0     0   0   0  510      2   210    56    73    3.6 Low
## 6     0     0     0   0   0  630     60   150    56    73    6.1 Low
```

# A first glm

- Suppose we want to fit a model on the sum $y_i =$ `rep.1 + rep.2 + rep.3`:

$$y_i \sim Bin(N_i, p_i), logit(p_i) = \alpha + \beta(x_i - \bar{x})$$

  where $x_i$ is the percentage of forest cover
- There are no random effects in this (yet) so we currently have just a glm
- Remember that the relationship between $x_i$ and $p_i$ (the probability of observing a bird) is not linear. People usually use $\exp(\beta)$ as a measure of the proportional increase in the probability associated with a unit increase in $x$

# Fitting the glm

```
summary(glm(cbind(y, N) ~ x, family = binomial(link = logit)))
```

```
##
## Call:
## glm(formula = cbind(y, N) ~ x, family = binomial(link = logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8968  -1.0906  -0.8140   0.4749   2.1429
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.188254   0.179000 -12.225  < 2e-16 ***
## x            0.020322   0.003289   6.178 6.49e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 332.73  on 236  degrees of freedom
## Residual deviance: 292.22  on 235  degrees of freedom
## AIC: 463.89
##
## Number of Fisher Scoring iterations: 4
```

# Changing to a glmm

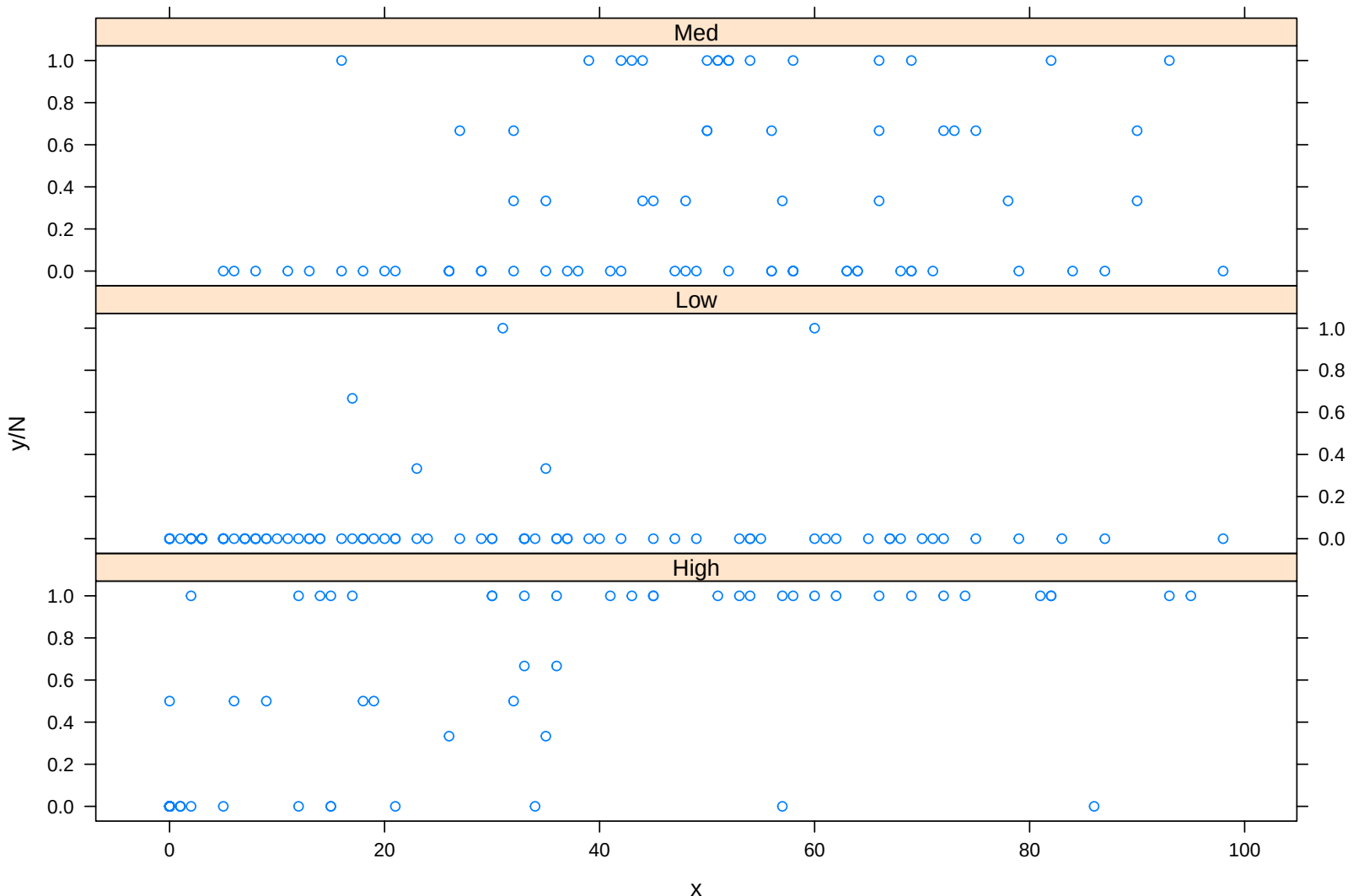
- Now extend the model to have a random intercept by altitude

$$y_{ij} \sim Bin(N_{ij}, p_{ij}), logit(p_{ij}) = \alpha_j + \beta(x_i - \bar{x})$$

  with $\alpha_j \sim N(\mu_\alpha, \sigma^2_\alpha)$.
- Now $y_{ij}$ is the count for observation $i$ at altitude $j$. Other parameters defined similarly
- This means that there will be three different $\alpha_j$ values for altitude low, medium and high

# Plot of the data

```
xyplot(y/N ~ x|alt, swt, type='p',
       layout=c(1,3), index.cond = function(x,y)max(y))
```



# Fitting the glmm

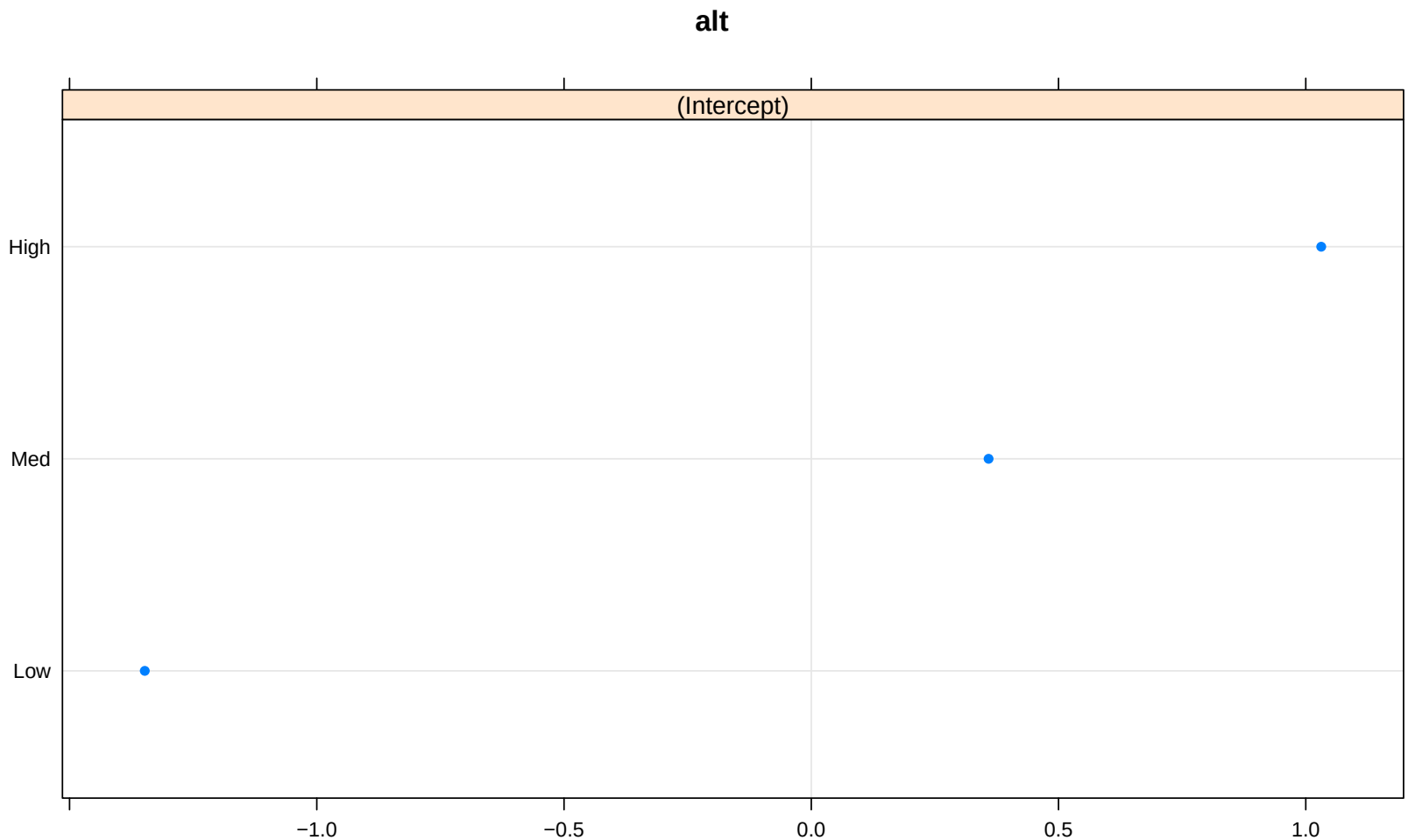
```
glmm_1 = glmer(cbind(y, N) ~ x + (1 | alt),
               family = binomial, data = swt)
summary(glmm_1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(y, N) ~ x + (1 | alt)
##    Data: swt
##
##      AIC      BIC   logLik deviance df.resid
##    397.1    407.5   -195.5    391.1      234
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8988 -0.7074 -0.3283  0.2221  5.8152
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  alt    (Intercept) 1.072    1.036
## Number of obs: 237, groups:  alt, 3
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.416608   0.631343  -3.828 0.000129 ***
## x            0.018246   0.003536   5.160 2.47e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr)
## x -0.257
```

# Look at the random effects

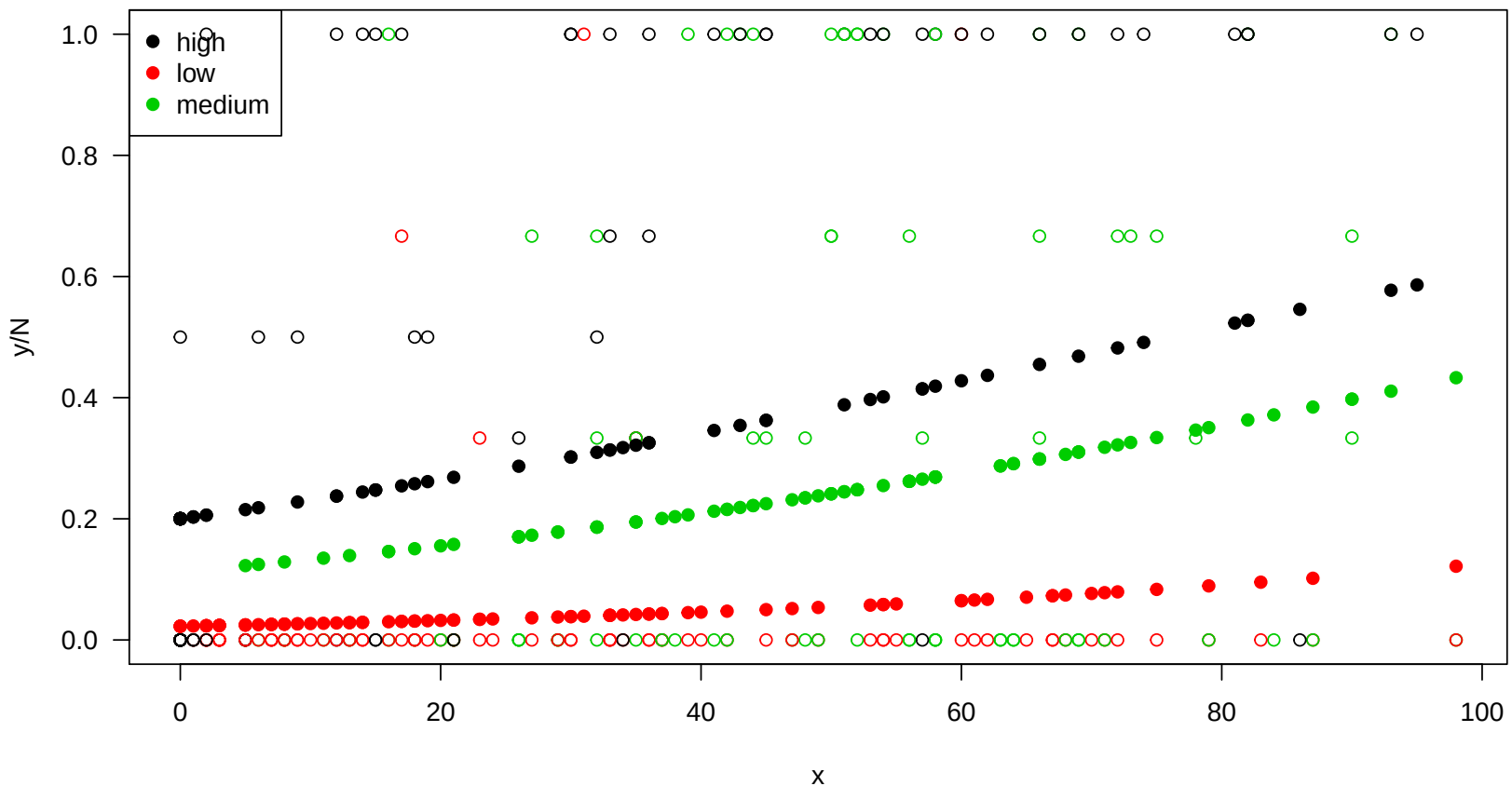
```
library(lattice)
dotplot(ranef(glmm_1))
```

```
## $alt
```



# Plot the probabilities

```
p_est = predict(glmm_1, type = 'response')
plot(x, y/N, col = swt$alt, las = 1)
points(x, p_est, col = swt$alt, pch = 19)
legend('topleft', c('high', 'low', 'medium'), pch = 19, col
```



# A model with varying intercepts and slopes

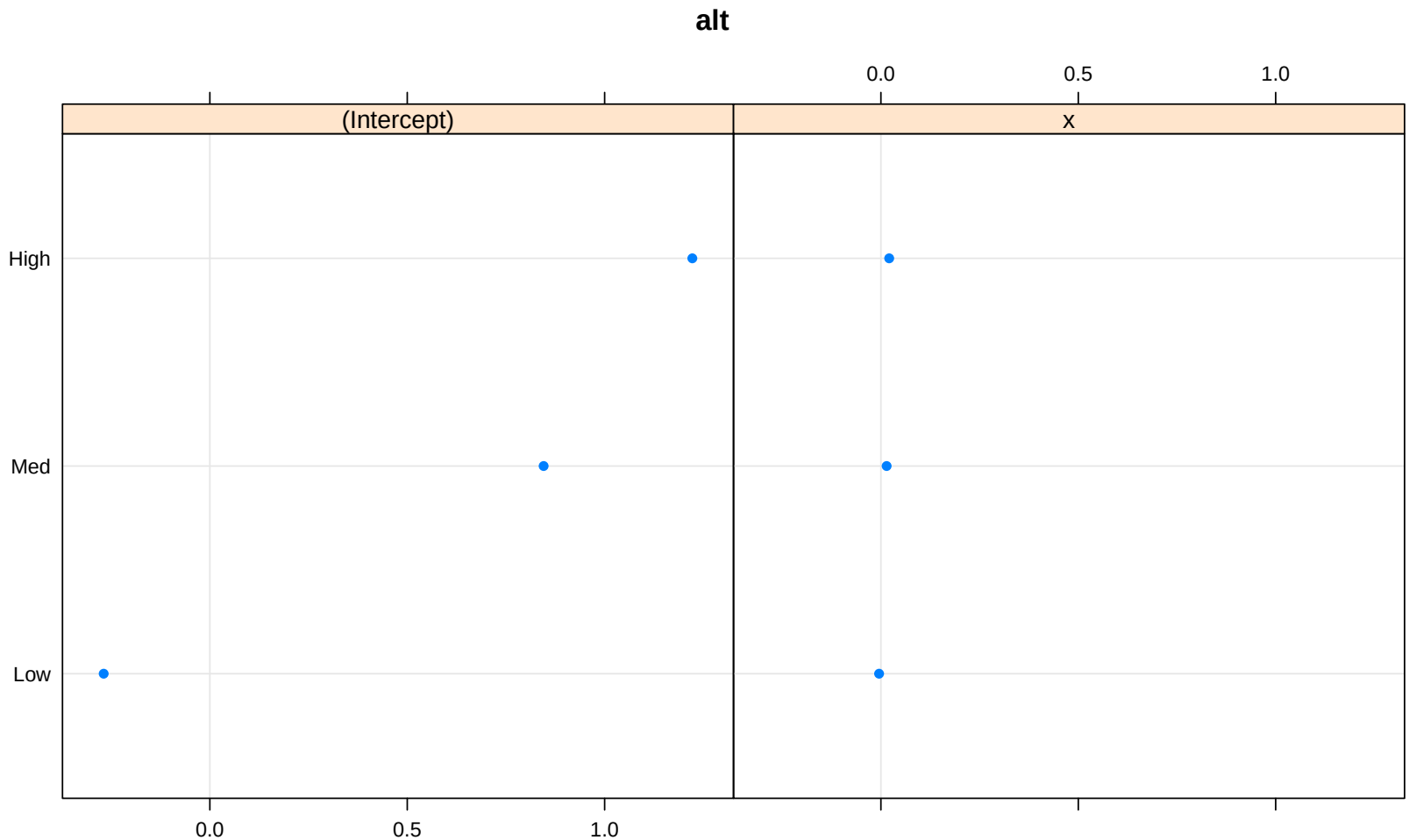
```
glmm_2 = glmer(cbind(y, N) ~ (x | alt),
               family = binomial, data = swt)
summary(glmm_2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(y, N) ~ (x | alt)
##    Data: swt
##
##      AIC      BIC   logLik deviance df.resid
##    401.2    415.1   -196.6    393.2      233
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0280 -0.6716 -0.3727  0.1687  6.0027
##
## Random effects:
##  Groups Name        Variance  Std.Dev. Corr
##  alt    (Intercept) 0.7797798 0.88305
##         x           0.0002299 0.01516  1.00
## Number of obs: 237, groups:  alt, 3
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.7114     0.5904  -4.593 4.38e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Plot the new random effects

```
dotplot(ranef(glmm_2))
```

```
## $alt
```



# Compare the two binomial models

```
anova(glmm_1, glmm_2)
```

```
## Data: swt
## Models:
## glmm_1: cbind(y, N) ~ x + (1 | alt)
## glmm_2: cbind(y, N) ~ (x | alt)
##         Df    AIC    BIC  logLik deviance Chisq Chi Df Pr
## glmm_1   3 397.07 407.48 -195.54   391.07
## glmm_2   4 401.22 415.09 -196.61   393.22     0      1
```

# A Poisson model - glm set-up

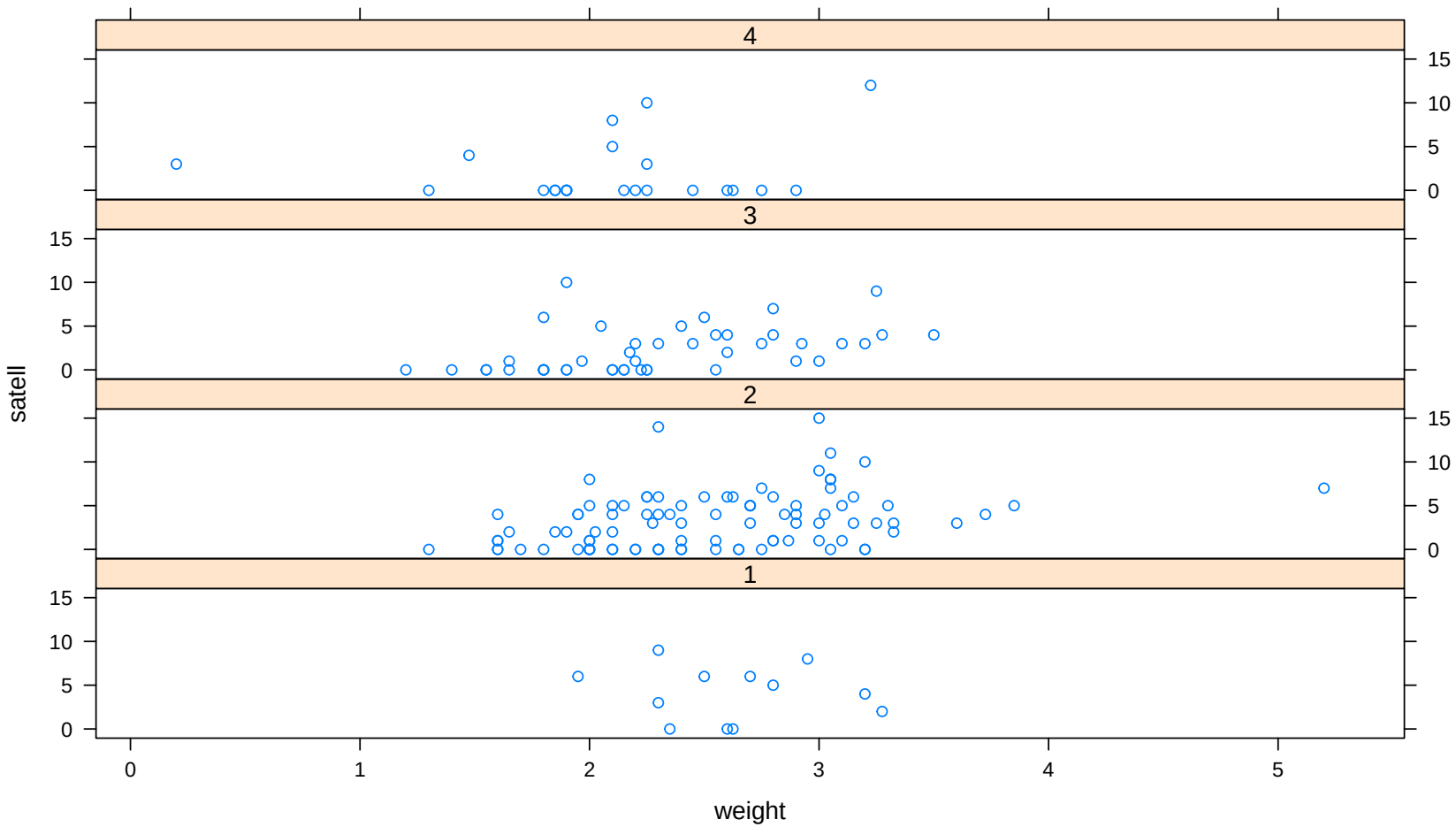
- If we wanted a Poisson glm we would set it up as:

$$y_i \sim Po(\lambda_i);\ \log(\lambda_i) = \alpha + \beta x_i$$

- log is the link function here. Again people often use $\exp(\beta)$ as an estimate of how the rate parameter $\lambda$ is affected by $x$
- The Poisson is a really un-realistic model. Remember it assumes that the mean and the variance of $y$ are the same. This almost never occurs in real data

# Poisson glm - example data

```
horseshoe = read.csv('../data/horseshoe.csv')
xyplot(satell ~ weight|as.factor(color), horseshoe,
       type='p',layout=c(1,4))
```



# Fit Poisson glm

```
summary(glm(satell ~ weight, data = horseshoe,family = poisson(link = log)))
```

```
##
## Call:
## glm(formula = satell ~ weight, family = poisson(link = log),
##     data = horseshoe)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9182  -2.0169  -0.5926   1.0290   4.9755
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.36997    0.17811  -2.077   0.0378 *
## weight       0.56837    0.06496   8.750   <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: 632.79  on 172  degrees of freedom
## Residual deviance: 564.35  on 171  degrees of freedom
## AIC: 923.65
##
## Number of Fisher Scoring iterations: 5
```

# A Poisson glmm

```
glmm_3 = glmer(satell ~ weight + (1 | color),
               family = poisson, data = horseshoe)
summary(glmm_3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: satell ~ weight + (1 | color)
##    Data: horseshoe
##
##      AIC      BIC   logLik deviance df.resid
##    924.6    934.1   -459.3    918.6      170
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0931 -1.4290 -0.4841  1.0286  6.9111
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  color  (Intercept) 0.01212  0.1101
## Number of obs: 173, groups:  color, 4
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.3249     0.1920  -1.692   0.0907 .
## weight        0.5453     0.0683   7.983 1.42e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr)
## weight -0.917
```

# A different type of Poisson model

- As previously stated the Poisson is a bit unrealistic (because of mean = variance assumption)
- Random effects can be added in to model overdispersion:

$$y_i \sim Po(\lambda_i),\ \log(\lambda_i) = \alpha + \beta x_i + \epsilon_i$$

with $\epsilon_i \sim N(0, \sigma^2)$

- This is just adding an *individual-level* random effect (or a residual term) and is the same as:

$$\log(\lambda_i) \sim N(\alpha + \beta x_i, \sigma^2)$$

# Poisson over-dispersion model

```r
horseshoe$obs <- 1:nrow(horseshoe)
glmm_4 = glmer(satell ~ weight + (1 | obs),
               family = poisson, data = horseshoe,
               control = glmerControl(optimizer = "Nelder_Mead"))
summary(glmm_4)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: satell ~ weight + (1 | obs)
##    Data: horseshoe
## Control: glmerControl(optimizer = "Nelder_Mead")
##
##      AIC      BIC   logLik deviance df.resid
##    769.5    778.9   -381.7    763.5      170
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.05174 -0.78335 -0.00272  0.34225  1.39463
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  obs    (Intercept) 0.9822   0.9911
## Number of obs: 173, groups:  obs, 173
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.4538     0.4352  -3.340 0.000837 ***
## weight        0.8254     0.1624   5.083 3.71e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr)
```

# Model results

```
anova(glmm_3, glmm_4)
```

```
## Data: horseshoe
## Models:
## glmm_3: satell ~ weight + (1 | color)
## glmm_4: satell ~ weight + (1 | obs)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df P
## glmm_3  3 924.60 934.06 -459.30   918.60
## glmm_4  3 769.48 778.94 -381.74   763.48 155.12      0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

Poisson OD model a far better fit

# Some notes on glmm

- Really just scratching the surface. Many many options for distributions and link functions
- Basic idea (and computational approach) is exactly the same as for `lmer`
- Individual level random effects are often useful in glmms as they can represent over-dispersion. They essentially just add a residual effect into the linked mean

# Summary

- We have seen a Binomial and a Poisson generalised linear mixed model (glmm)
- Very simple to fit in `lme4` using the `glmer` function. Exactly the same formula approach
- Over-dispersion a useful trick for getting good-fitting models