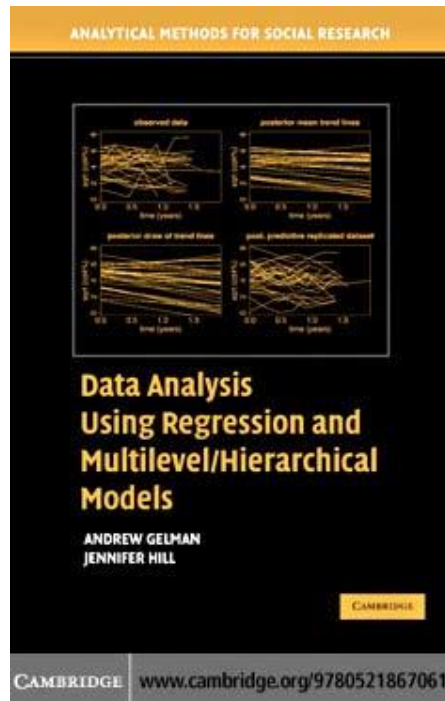


# Lecture 7

## Generalised Linear Mixed Models

# multilevel logistic regression



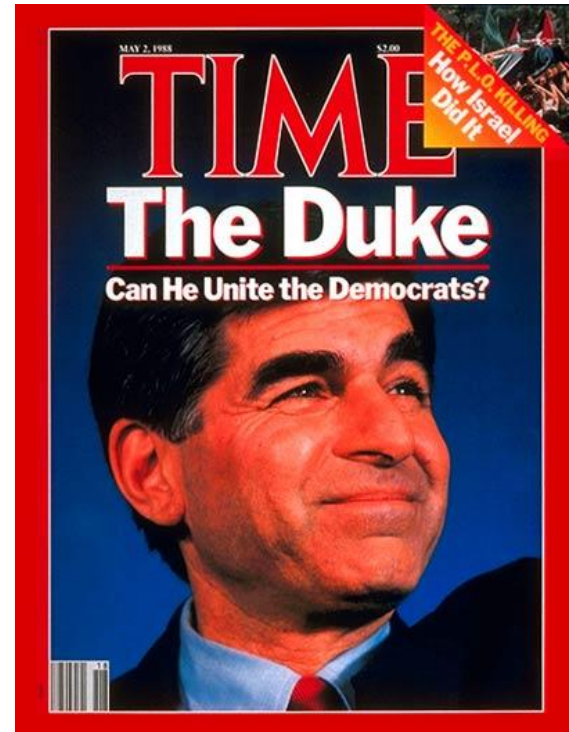
State level opinions from national polls

Gelman & Hill, Chapter 14

simplified to republicans versus democrats



George H Bush



Michael Dukakis

binary data

**Data set**

1988 election opinion polls

**binary**

1 = preferred Bush (Republican)

0 = preferred Dukakis (Democrat)

**multilevel**

state level

**predictor variables**

we will use three: race, sex and age

## Exercise 7.1

- Read in the data from the file
- Take a quick look at them
- Use a glm to answer the question:  
Were black voters and female voters  
more or less likely to prefer George  
Bush? (additive model)

# Assumptions of GLMs again

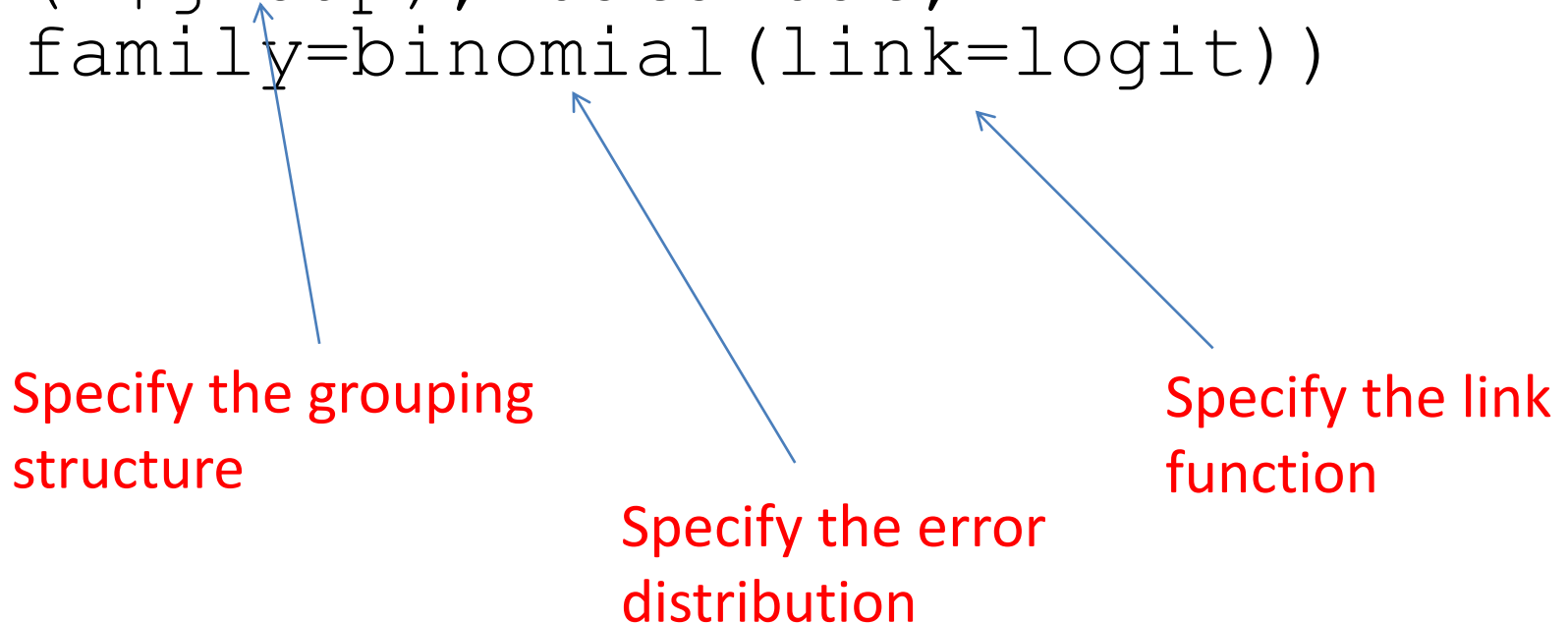
- (and things to do)
  1. The model makes biological sense (think critically!)
  2. Additivity (consider transformations, interactions, GLMs)
  3. Linearity (plot data, transformations, GLMs)
  4. Independence of errors (blocking, mixed effects models)
  5. Homoscedasticity – equal variance of errors (GLMs, weighted least squares)
  6. Allow Gaussian or non-Gaussian distributions for the residuals (e.g. Binomial, Poisson, Gamma).

# GLMMs

- Allow incorporation of both fixed and random effects into GLM structure
- Use *glmer* function

```
> mod <- glmer(y ~ x1 + x2 +  
  (1|group), data=dat,  
  family=binomial(link=logit))
```

Specify the grouping  
structure



The diagram consists of three blue arrows pointing upwards from red text labels to specific parts of the R code. The first arrow points from 'Specify the grouping structure' to '(1|group)'. The second arrow points from 'Specify the error distribution' to 'family=binomial'. The third arrow points from 'Specify the link function' to 'link=logit'.

Specify the error  
distribution

Specify the link  
function

# Logit links

- Useful for Binomial models
- Good link functions = response goes from  $-\infty$  to  $+\infty$
- $\text{logit}(p) = \log[p/(1-p)]$
- $\text{logit}(0) = -\infty$ ;  $\text{logit}(1) = \infty$
- Can get  $\text{logit}(p)$  in R with  
`q=logis (p)`
- The inverse is obtained with  
`p=plogis (q)`



## Exercise 7.2

- Use ggplot to understand how black female voters vote in different states
- Now use a glmer to test the same relationships:

Were black voters and female voters more or less likely to prefer George Bush?

# Interpreting the output

Generalized linear mixed model fit by maximum  
likelihood (Laplace Approximation) [glmerMod]

Family: binomial ( logit )  
Formula: bush ~ female + black + (1 | state)  
Data: polls

AIC	BIC	logLik	deviance	df.resid
2664.7	2687.1	-1328.3	2656.7	2009

Random effects:

Groups	Name	Variance	Std.Dev.
state	(Intercept)	0.1687	0.4108

Number of obs: 2013, groups: state, 48

# Interpreting estimates

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.4377	0.1015	4.31	1.6e-05
female1	-0.0931	0.0951	-0.98	0.33
black1	-1.7397	0.2095	-8.30	< 2e-16

Correlation of Fixed Effects:

	(Intr)	female1
female1	-0.552	
black1	-0.118	-0.005

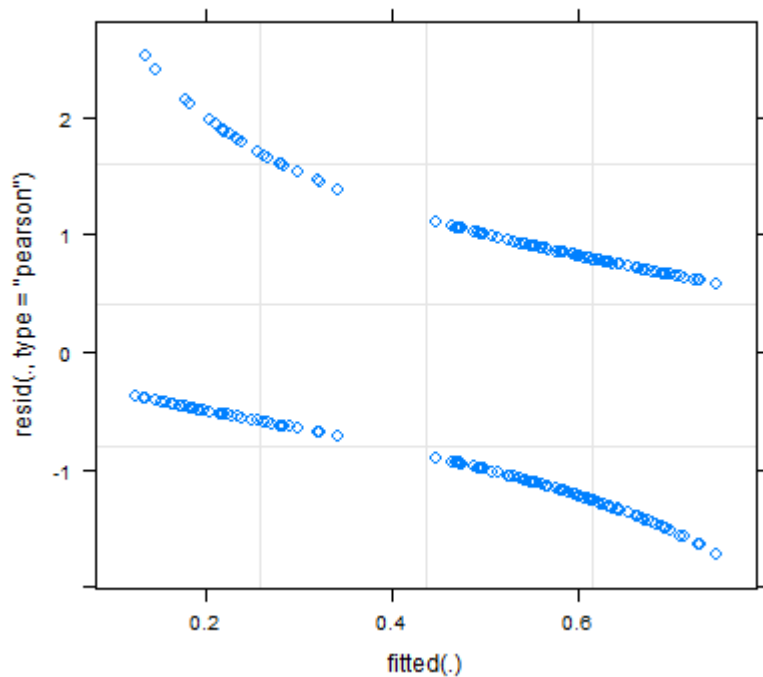
## Exercise 7.3

1. Do different voting patterns among females of different states explain some of the variation? (new random effects structure required...)

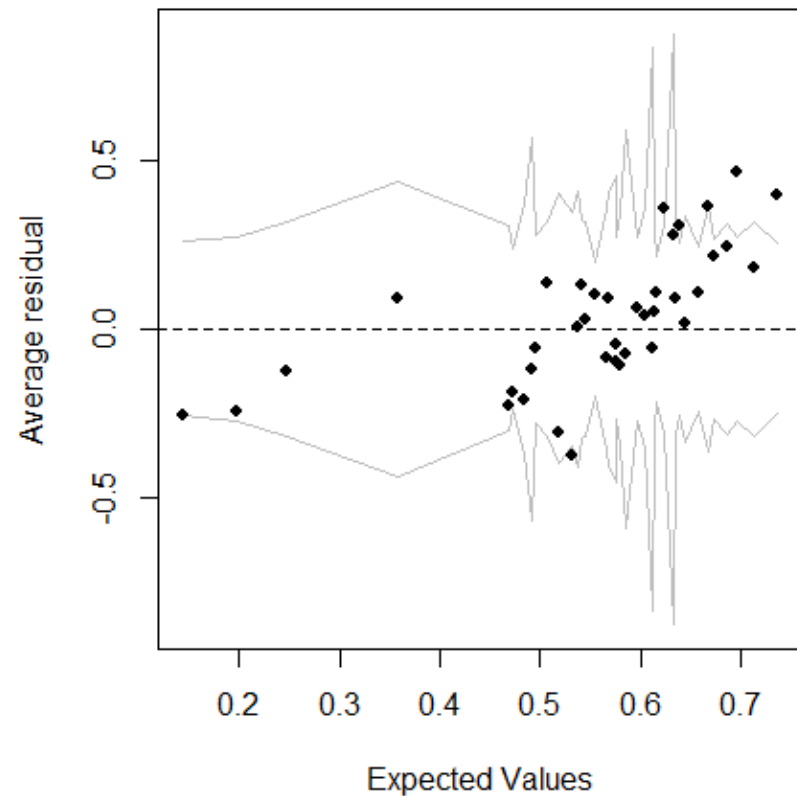
# Diagnostics checking

- Hard with GLMMs (especially binary ones) but there are options
- `binned.plot()` – for binary data.
- Check for residual over-dispersion
- Check if the random effects are normally distributed (as before)

# Diagnostic plots



**Binned residual plot**



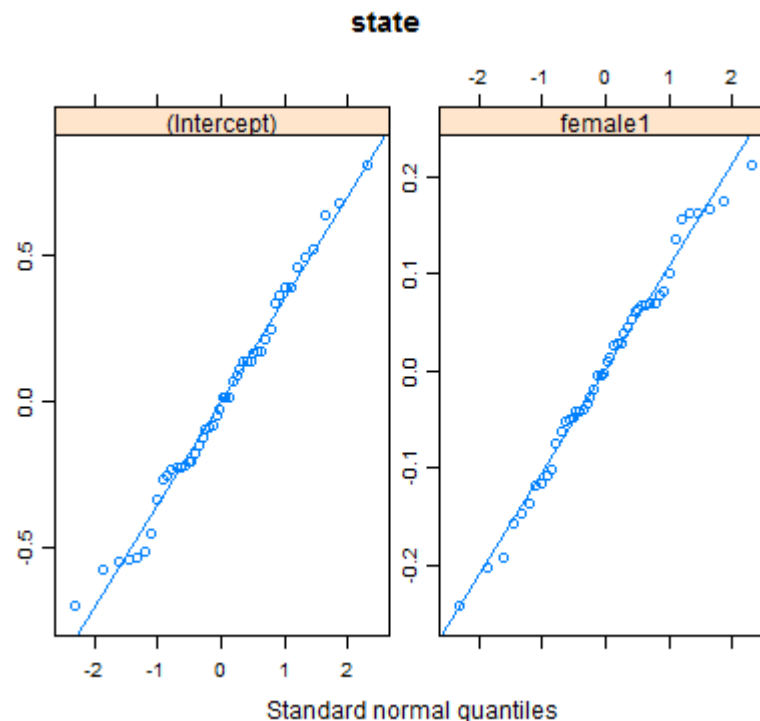
# Over-dispersion

- Binomial and Poisson models assume that the variance is a function of the mean
  - Once you know the mean, you should be able to estimate the variance!
- Residual deviance  $\approx$  Residual degrees of freedom.
- Not ideal for binary models but indicative

# Check the random effects

- Normally distributed?

```
> qqmath(ranef(mod.polls.glmm1),  
type=c('p', 'r'))
```





## Exercise 7.4

Using your last GLMM:

1. Check residual vs fitted plots
2. Check binned plot of residuals
3. Check for overdispersion of residuals
4. Check whether random effects are normal

# Poisson glmms

- Used for count data – residuals predicted to be drawn from a poisson distribution

```
> mod <- glmer(y ~ x1 + x2 +  
  (1|group), data=dat,  
  family=poisson(link=log) )
```

# Overdispersion again

- Overdispersion is a common problem in Poisson models where the actual variance exceeds the mean (recall Poisson models assume  $\mu = \sigma = \lambda$ , a single parameter).
- Overdispersion can be caused by:
  - missing covariates,
  - non-independent (aggregated) data, or
  - an excess frequency of zeroes (zero-inflation).

# Observation level random effect

- An OLRE assumes that some of the excess variation in the poisson data is actually a distribution process that needs to be accounted for
- So the OLRE is a normally distributed variance term (hence random) that gets added into a model to account for this excess variance:  $\epsilon_{\text{OLRE}} \sim N(0, \sigma_z^2)$
- It can be easily added.
- `pest$indx <- 1:nrow(pest)`
- `glmer(Y~X+ ... + (1|indx),  
data=pest, family=poisson(link=log) )`

# Observation level random effect

- OLREs are good at certain types of data problem but not at others
- Good for:
  - Variance  $>$  mean in a poisson-type distribution
  - Aggregation in count data (negative binomial distributions)
- Bad for:
  - Data with serious zero inflation (many zero values)
- OLREs can also be used for overdispersed poisson data where no other random grouping parameter is apparent; this may make them a better solution than pseudo-distributions (e.g. pseudopoisson) because they yield proper maximum likelihood estimates.

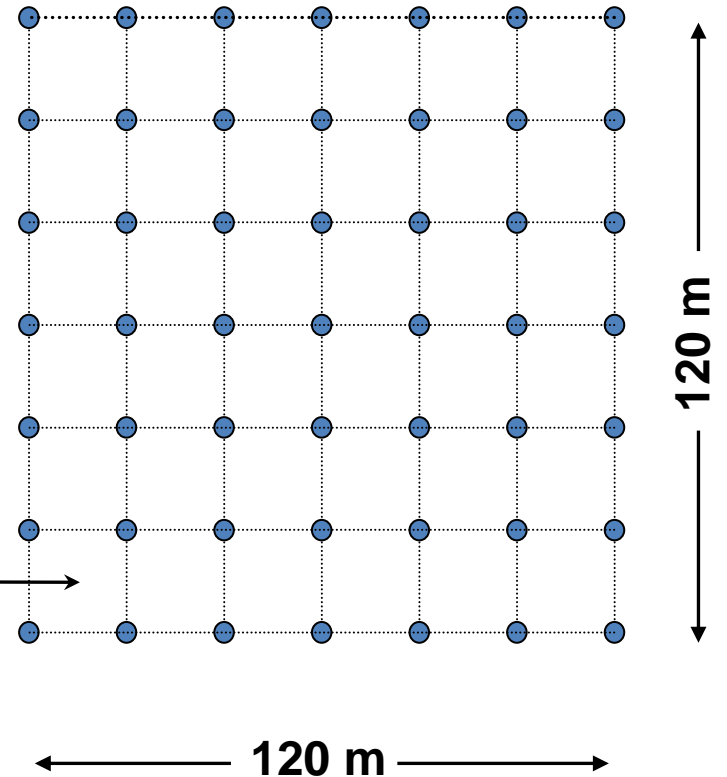
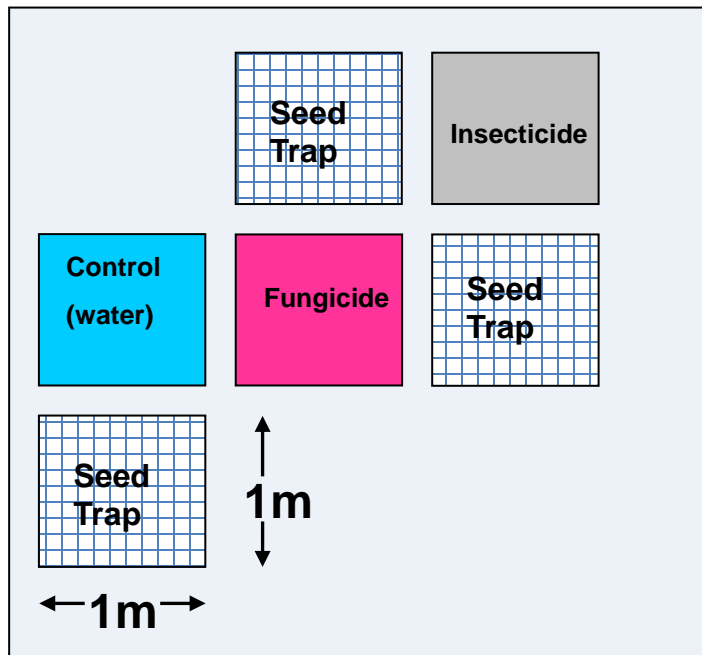
# The pest dataset



- Forest composition depends substantially on the fate of seedling trees. Recruitment of seedling trees depends on availability of seeds and also survival through disturbance hazards and competition. Seedling survival may be heavily impacted by insect herbivory and fungal pathogen attack. But are both processes important in forest ecosystems? Or is one of these hazards overriding in its influence?



# The pest dataset



## Exercise 7.5

- Use ggplot to look at how number of seedlings responds to treatments across stations
- Fit a GLMM to the pest data set to answer the following questions
  1. What is effect of insecticide and fungicide on number of seedlings?
  2. Does the model fit the data well?
  3. What might you do to improve the model fit? (think model covariates, OLREs)



# Inference: Bootstrapping with GLMMs

- Bootstrap approach is similar to that with a Gaussian response.
- Parametric bootstrap still makes assumptions about relationships between means and variances (i.e. overdispersion is STILL a problem)

## Exercise 7.6

- Evaluate whether treatments are affecting survival using LRTs
- Evaluate the same using bootstrapped confidence intervals. Compare the models without and with the individual-level random effect

# Making predictions with GLMs

- Use the same function `predict` as for `lm`
- Some differences!
- One extra argument is `'type'`.  
This specifies what scale you want the prediction on
  - The data scale ('response')
  - The linear predictor scale ('link')
- For example:  

```
predict(mod5, newdata=newdata,  
type='response')
```

# Plotting with standard errors

- There is no `interval` argument for `predict.glm()`
- Can get standard errors, and calculate the confidence intervals manually.
- Standard errors are on link scale – and are asymmetrical on the response scale.
- **Therefore need to do the prediction on the link scale (log or logit usually) and back-transform.**
- Remember to add standard errors to **UNTRANSFORMED** predictions.

# Calculating the expected values

1. Extract the model matrix ( $X$ ), using  
`model.matrix()`
2. Calculate the expected values on the linear predictor scale as  
`X %*% fixef(model)`
3. Add on the random effect BLUPs if you want.
4. You can also use the built-in predict function for the expected values (not confidence intervals)

# Calculating the confidence intervals

1. Extract the variance covariance matrix with  
`vcov(mod)`
2. Calculate the standard errors as  
`sqrt(diag(X %*% VCV %*% t(X)))`
3. Add or subtract the standard errors
4. Back transform to the data scale.
5. NB: for unknown groups you need to add on the variance components BEFORE you take the root in step 2 (recall previous lesson)

## Exercise 7.7

- Fit a model to the plantdamage2.csv data on **seedling survival**. Please note the structure of the data: four seedlings were planted per treatment in each greenhouse and their survival was observed after some fixed period of time.
- Calculate the predictions and confidence intervals for all treatments in the first four greenhouses
  - for a known shadehouse
  - For an unknown shadehouse
- Plot the results in fitted and back-transformed ranges.