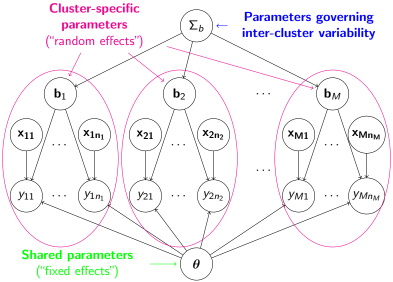
A Brief and Friendly Introduction to Mixed-Effects Models in Linguistics



slides by Roger Levy
presented (and slightly edited) by Klinton Bicknell

UC San Diego, Department of Linguistics

15 July 2009

Goals of this talk

- Briefly review generalized linear models and how to use them
- Give a precise description of multi-level models
- Show how to draw inferences using a multi-level model (*fitting* the model)
- Discuss how to interpret model parameter estimates
  - Fixed effects
  - Random effects

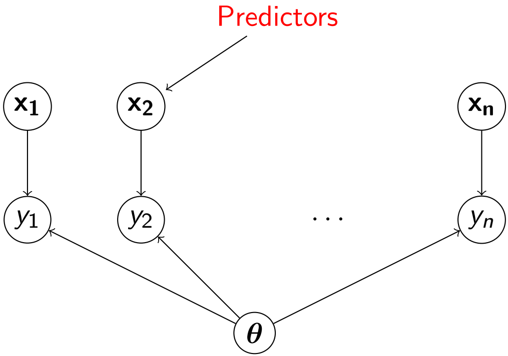
Reviewing generalized linear models I

Goal: model the effects of predictors (independent variables) $\mathbf{X}$ on a response (dependent variable) Y.

Reviewing generalized linear models I

Goal: model the effects of predictors (independent variables) $\mathbf{X}$ on a response (dependent variable) Y.

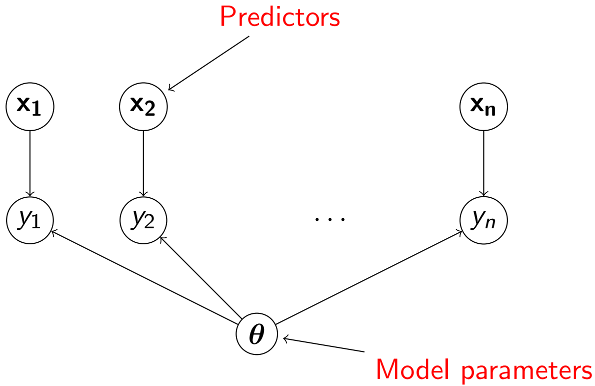
The picture:



Reviewing generalized linear models I

Goal: model the effects of predictors (independent variables) $\mathbf{X}$ on a response (dependent variable) Y.

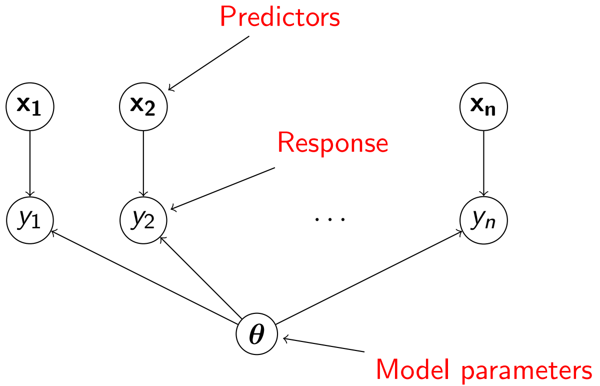
The picture:



Reviewing generalized linear models I

Goal: model the effects of predictors (independent variables) $\mathbf{X}$ on a response (dependent variable) Y.

The picture:



Reviewing GLMs II

Assumptions of the generalized linear model (GLM):

1. Predictors $\{X_i\}$ influence Y through the mediation of a linear predictor η;

Reviewing GLMs II

Assumptions of the generalized linear model (GLM):

1. Predictors $\{X_i\}$ influence Y through the mediation of a linear predictor η;
2. η is a linear combination of the $\{X_i\}$:

Reviewing GLMs II

Assumptions of the generalized linear model (GLM):

1. Predictors $\{X_i\}$ influence Y through the mediation of a linear predictor η;
2. η is a linear combination of the $\{X_i\}$:

$$\eta = \alpha + \beta_1 X_1 + \cdots + \beta_N X_N \qquad \text{(linear predictor)}$$

Reviewing GLMs II

Assumptions of the generalized linear model (GLM):

1. Predictors $\{X_i\}$ influence Y through the mediation of a linear predictor η;
2. η is a linear combination of the $\{X_i\}$:

$$\eta = \alpha + \beta_1 X_1 + \cdots + \beta_N X_N \qquad \text{(linear predictor)}$$

3. η determines the predicted mean μ of Y

$$\eta = l(\mu) \qquad \text{(link function)}$$

Reviewing GLMs II

Assumptions of the generalized linear model (GLM):

1. Predictors $\{X_i\}$ influence Y through the mediation of a linear predictor η;
2. η is a linear combination of the $\{X_i\}$:

$$\eta = \alpha + \beta_1 X_1 + \cdots + \beta_N X_N \qquad \text{(linear predictor)}$$

3. η determines the predicted mean μ of Y

$$\eta = l(\mu) \qquad \text{(link function)}$$

4. There is some noise distribution of Y around the predicted mean μ of Y:

$$P(Y = y; \mu)$$

Reviewing GLMs III

Linear regression, which underlies ANOVA, is a kind of generalized linear model.

Reviewing GLMs III

Linear regression, which underlies ANOVA, is a kind of generalized linear model.

- The predicted mean is just the linear predictor:

$$\eta = l(\mu) = \mu$$

Reviewing GLMs III

Linear regression, which underlies ANOVA, is a kind of generalized linear model.

- The predicted mean is just the linear predictor:

$$\eta = l(\mu) = \mu$$

- Noise is normally (=Gaussian) distributed around 0 with standard deviation σ:

$$\epsilon \sim N(0, \sigma)$$

Reviewing GLMs III

Linear regression, which underlies ANOVA, is a kind of generalized linear model.

- The predicted mean is just the linear predictor:

$$\eta = l(\mu) = \mu$$

- Noise is normally (=Gaussian) distributed around 0 with standard deviation σ:

$$\epsilon \sim N(0, \sigma)$$

- This gives us the traditional linear regression equation:

$$Y = \overbrace{\alpha + \beta_1 X_1 + \cdots + \beta_n X_n}^{\text{Predicted Mean } \mu = \eta} + \overbrace{\epsilon}^{\text{Noise} \sim N(0,\sigma)}$$

Reviewing GLMs IV

$$Y = \overbrace{\alpha + \beta_1 X_1 + \cdots + \beta_n X_n}^{\text{Predicted Mean}} + \overbrace{\epsilon}^{\text{Noise} \sim N(0,\sigma)}$$

- How do we fit the parameters β_i and σ (choose *model coefficients*)?
- There are two major approaches (deeply related, yet different) in widespread use:

Reviewing GLMs IV

$$Y = \overbrace{\alpha + \beta_1 X_1 + \cdots + \beta_n X_n}^{\text{Predicted Mean}} + \overbrace{\epsilon}^{\text{Noise} \sim N(0,\sigma)}$$

- How do we fit the parameters β_i and σ (choose *model coefficients*)?
- There are two major approaches (deeply related, yet different) in widespread use:
  - The principle of maximum likelihood: pick parameter values that maximize the probability of your data Y

    choose $\{\beta_i\}$ *and* σ *that make the likelihood* $P(Y|\{\beta_i\},\sigma)$ *as large as possible*

Reviewing GLMs IV

$$Y = \overbrace{\alpha + \beta_1 X_1 + \cdots + \beta_n X_n}^{\text{Predicted Mean}} + \overbrace{\epsilon}^{\text{Noise} \sim N(0,\sigma)}$$

- How do we fit the parameters β_i and σ (choose *model coefficients*)?
- There are two major approaches (deeply related, yet different) in widespread use:
  - The principle of maximum likelihood: pick parameter values that maximize the probability of your data Y

    choose $\{\beta_i\}$ *and* σ *that make the likelihood* $P(Y|\{\beta_i\}, \sigma)$ *as large as possible*

  - Bayesian inference: put a probability distribution on the model parameters and update it on the basis of what parameters best explain the data

Reviewing GLMs IV

$$Y = \overbrace{\alpha + \beta_1 X_1 + \cdots + \beta_n X_n}^{\text{Predicted Mean}} + \overbrace{\epsilon}^{\text{Noise} \sim N(0,\sigma)}$$

- How do we fit the parameters β_i and σ (choose *model coefficients*)?
- There are two major approaches (deeply related, yet different) in widespread use:
  - The principle of maximum likelihood: pick parameter values that maximize the probability of your data Y

    choose $\{\beta_i\}$ and σ that make the likelihood $P(Y|\{\beta_i\}, \sigma)$ as large as possible

  - Bayesian inference: put a probability distribution on the model parameters and update it on the basis of what parameters best explain the data

$$P(\{\beta_i\}, \sigma|Y) = \frac{P(Y|\{\beta_i\}, \sigma) \overbrace{P(\{\beta_i\}, \sigma)}^{\text{Prior}}}{P(Y)}$$

Reviewing GLMs IV

$$Y = \overbrace{\alpha + \beta_1 X_1 + \cdots + \beta_n X_n}^{\text{Predicted Mean}} + \overbrace{\epsilon}^{\text{Noise} \sim N(0,\sigma)}$$

- How do we fit the parameters β_i and σ (choose *model coefficients*)?
- There are two major approaches (deeply related, yet different) in widespread use:
  - The principle of maximum likelihood: pick parameter values that maximize the probability of your data Y

    choose $\{\beta_i\}$ and σ that make the likelihood $P(Y|\{\beta_i\},\sigma)$ as large as possible

  - Bayesian inference: put a probability distribution on the model parameters and update it on the basis of what parameters best explain the data

$$P(\{\beta_i\},\sigma|Y) = \frac{\overbrace{P(Y|\{\beta_i\},\sigma)}^{\text{Likelihood}}\overbrace{P(\{\beta_i\},\sigma)}^{\text{Prior}}}{P(Y)}$$

Reviewing GLMs V: a simple example

- You are studying non-word RTs in a lexical-decision task

Reviewing GLMs V: a simple example

- You are studying non-word RTs in a lexical-decision task

  `tpozt` *Word or non-word?*

Reviewing GLMs V: a simple example

- You are studying non-word RTs in a lexical-decision task

`tpozt`	*Word or non-word?*
`houze`	*Word or non-word?*

Reviewing GLMs V: a simple example

- You are studying non-word RTs in a lexical-decision task

  `tpozt` *Word or non-word?*

  `houze` *Word or non-word?*

- Non-words with different *neighborhood densities** should have different average RT *(= number of neighbors of edit-distance 1)

Reviewing GLMs V: a simple example

- You are studying non-word RTs in a lexical-decision task

  `tpozt` *Word or non-word?*

  `houze` *Word or non-word?*

- Non-words with different *neighborhood densities** should have different average RT *(= number of neighbors of edit-distance 1)
- A simple model: assume that neighborhood density has a *linear* effect on average RT, and trial-level noise is *normally distributed** *(n.b. wrong–RTs are skewed—but not horrible.)

Reviewing GLMs V: a simple example

- You are studying non-word RTs in a lexical-decision task

  `tpozt` *Word or non-word?*

  `houze` *Word or non-word?*

- Non-words with different *neighborhood densities** should have different average RT *(= number of neighbors of edit-distance 1)
- A simple model: assume that neighborhood density has a *linear* effect on average RT, and trial-level noise is *normally distributed** *(n.b. wrong–RTs are skewed—but not horrible.)
- If x_i is neighborhood density, our simple model is

$$RT_i = \alpha + \beta x_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

Reviewing GLMs V: a simple example

- You are studying non-word RTs in a lexical-decision task

  `tpozt` *Word or non-word?*
  `houze` *Word or non-word?*

- Non-words with different *neighborhood densities** should have different average RT *(= number of neighbors of edit-distance 1)
- A simple model: assume that neighborhood density has a *linear* effect on average RT, and trial-level noise is *normally distributed** *(n.b. wrong–RTs are skewed—but not horrible.)
- If x_i is neighborhood density, our simple model is

$$RT_i = \alpha + \beta x_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- We need to draw inferences about α, β, and σ

Reviewing GLMs V: a simple example

- You are studying non-word RTs in a lexical-decision task

  `tpozt` *Word or non-word?*

  `houze` *Word or non-word?*

- Non-words with different *neighborhood densities** should have different average RT *(= number of neighbors of edit-distance 1)
- A simple model: assume that neighborhood density has a *linear* effect on average RT, and trial-level noise is *normally distributed** *(n.b. wrong–RTs are skewed—but not horrible.)
- If x_i is neighborhood density, our simple model is

$$RT_i = \alpha + \beta x_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- We need to draw inferences about α, β, and σ
- e.g., "Does neighborhood density affects RT?"→ is β reliably non-zero?

Reviewing GLMs VI

- We'll use length-4 nonword data from (Bicknell et al., 2008), such as:

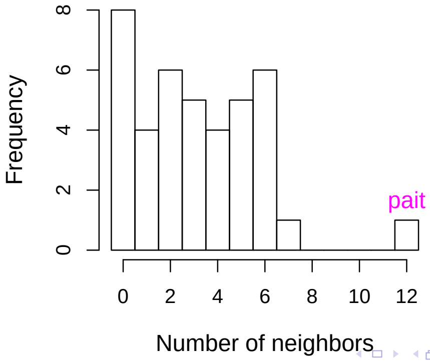
Few neighbors			*Many neighbors*		
`gaty`	`peme`	`rixy`	`lish`	`pait`	`yine`

Reviewing GLMs VI

- We'll use length-4 nonword data from (Bicknell et al., 2008), such as:

Few neighbors			*Many neighbors*		
`gaty`	`peme`	`rixy`	`lish`	`pait`	`yine`

- There's a wide range of neighborhood density:



Reviewing GLMs VII: maximum-likelihood model fitting

$$RT_i = \alpha + \beta X_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- Here's a translation of our simple model into R:

```
RT ~ 1 + x
```

Reviewing GLMs VII: maximum-likelihood model fitting

$$RT_i = \alpha + \beta X_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- Here's a translation of our simple model into `R`:

  `RT ~ 1 + x`

- The noise is implicit in asking `R` to fit a *linear* model

Reviewing GLMs VII: maximum-likelihood model fitting

$$RT_i = \alpha + \beta X_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- Here's a translation of our simple model into `R`:

  `RT ~ 1 + x`

- The noise is implicit in asking `R` to fit a *linear* model
- (We can omit the `1`; `R` assumes it unless otherwise directed)

Reviewing GLMs VII: maximum-likelihood model fitting

$$RT_i = \alpha + \beta X_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- Here's a translation of our simple model into `R`:

```
RT ~ x
```

- The noise is implicit in asking `R` to fit a *linear* model
- (We can omit the `1`; `R` assumes it unless otherwise directed)

Reviewing GLMs VII: maximum-likelihood model fitting

$$RT_i = \alpha + \beta X_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- Here's a translation of our simple model into R:

  RT $\sim$ x

- The noise is implicit in asking R to fit a *linear* model
- (We can omit the 1; R assumes it unless otherwise directed)
- Example of fitting via maximum likelihood: one subject from Bicknell et al. (2008)

```
> m <- glm(RT ~ neighbors, d, family="gaussian")
```

Gaussian noise, implicit intercept

```
> summary(m)
[...]
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   382.997     26.837  14.271   <2e-16 ***
neighbors       4.828      6.553   0.737    0.466
> sqrt(summary(m)[["dispersion"]])
[1] 107.2248
```

Reviewing GLMs VII: maximum-likelihood model fitting

$$RT_i = \alpha + \beta X_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- Here's a translation of our simple model into R:

  RT $\sim$ x

- The noise is implicit in asking R to fit a *linear* model
- (We can omit the 1; R assumes it unless otherwise directed)
- Example of fitting via maximum likelihood: one subject from Bicknell et al. (2008)

```
> m <- glm(RT ~ neighbors, d, family="gaussian")
> summary(m)
[...]
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  382.997     26.837  14.271   <2e-16 ***
neighbors      4.828      6.553   0.737    0.466
> sqrt(summary(m)[["dispersion"]])
[1] 107.2248
```

Reviewing GLMs VII: maximum-likelihood model fitting

$$RT_i = \alpha + \beta X_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- Here's a translation of our simple model into R:

  $$\texttt{RT} \sim \texttt{x}$$

- The noise is implicit in asking R to fit a *linear* model
- (We can omit the 1; R assumes it unless otherwise directed)
- Example of fitting via maximum likelihood: one subject from Bicknell et al. (2008)

```
> m <- glm(RT ~ neighbors, d, family="gaussian")
> summary(m)
[...]
         α̂ →  Estimate Std. Error t value Pr(>|t|)
(Intercept)   382.997     26.837  14.271   <2e-16 ***
neighbors       4.828      6.553   0.737    0.466
> sqrt(summary(m)[["dispersion"]])
[1] 107.2248
```

Reviewing GLMs VII: maximum-likelihood model fitting

$$RT_i = \alpha + \beta X_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- Here's a translation of our simple model into R:

  $$\texttt{RT} \sim \texttt{x}$$

- The noise is implicit in asking R to fit a *linear* model
- (We can omit the 1; R assumes it unless otherwise directed)
- Example of fitting via maximum likelihood: one subject from Bicknell et al. (2008)

```
> m <- glm(RT ~ neighbors, d, family="gaussian")
> summary(m)
[...]
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   382.997     26.837  14.271   <2e-16 ***
neighbors       4.828      6.553   0.737    0.466
> sqrt(summary(m)[["dispersion"]])
[1] 107.2248
```

$\widehat{\alpha}$ → Estimate (Intercept): 382.997

$\widehat{\beta}$ → Estimate neighbors: 4.828

Reviewing GLMs VII: maximum-likelihood model fitting

$$RT_i = \alpha + \beta X_i + \overbrace{\epsilon_i}^{\sim N(0,\sigma)}$$

- Here's a translation of our simple model into R:

  RT ~ x

- The noise is implicit in asking R to fit a *linear* model
- (We can omit the 1; R assumes it unless otherwise directed)
- Example of fitting via maximum likelihood: one subject from Bicknell et al. (2008)

```
> m <- glm(RT ~ neighbors, d, family="gaussian")
> summary(m)
[...]
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  382.997     26.837  14.271   <2e-16 ***
neighbors      4.828      6.553   0.737    0.466
> sqrt(summary(m)[["dispersion"]])
[1] 107.2248
```

$\widehat{\alpha}$ → 382.997

$\widehat{\beta}$ → 4.828

$\widehat{\sigma}$ → 107.2248

Reviewing GLMs: maximum-likelihood fitting VIII

Intercept	383.00
neighbors	4.83
$\widehat{\sigma}$	107.22

Reviewing GLMs: maximum-likelihood fitting VIII

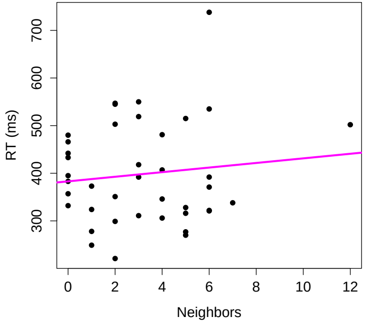
Intercept	383.00
neighbors	4.83
$\widehat{\sigma}$	107.22

- Estimated coefficients are what underlies "best linear fit" plots

Reviewing GLMs: maximum-likelihood fitting VIII

`Intercept`	`383.00`
`neighbors`	`4.83`
$\widehat{\sigma}$	`107.22`

- Estimated coefficients are what underlies "best linear fit" plots



Reviewing GLMs IX: Bayesian model fitting

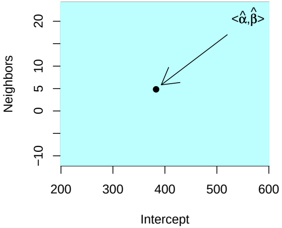
$$P(\{\beta_i\},\sigma|Y) = \frac{\overbrace{P(Y|\{\beta_i\},\sigma)}^{\text{Likelihood}}\ \overbrace{P(\{\beta_i\},\sigma)}^{\text{Prior}}}{P(Y)}$$

- Alternative to maximum-likelihood: Bayesian model fitting

Reviewing GLMs IX: Bayesian model fitting

$$P(\{\beta_i\}, \sigma | Y) = \frac{\overbrace{P(Y|\{\beta_i\}, \sigma)}^{\text{Likelihood}} \overbrace{P(\{\beta_i\}, \sigma)}^{\text{Prior}}}{P(Y)}$$

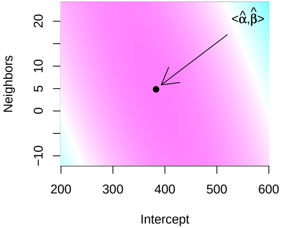
- Alternative to maximum-likelihood: Bayesian model fitting
- Simple (uniform, non-informative) prior: all combinations of (α, β, σ) equally probable



Reviewing GLMs IX: Bayesian model fitting

$$P(\{\beta_i\},\sigma|Y) = \frac{\overbrace{P(Y|\{\beta_i\},\sigma)}^{\text{Likelihood}}\ \overbrace{P(\{\beta_i\},\sigma)}^{\text{Prior}}}{P(Y)}$$

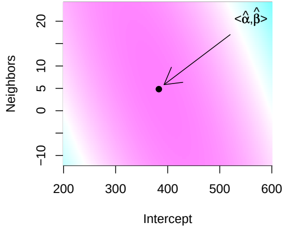
- Alternative to maximum-likelihood: Bayesian model fitting
- Simple (uniform, non-informative) prior: all combinations of (α, β, σ) equally probable
- Multiply by likelihood $\rightarrow$ posterior probability distribution over (α, β, σ)



Reviewing GLMs IX: Bayesian model fitting

$$P(\{\beta_i\}, \sigma | Y) = \frac{\overbrace{P(Y|\{\beta_i\}, \sigma)}^{\text{Likelihood}} \overbrace{P(\{\beta_i\}, \sigma)}^{\text{Prior}}}{P(Y)}$$

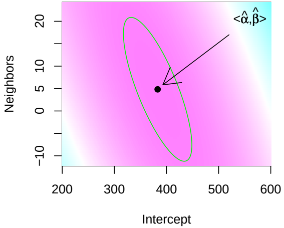
- Alternative to maximum-likelihood: Bayesian model fitting
- Simple (uniform, non-informative) prior: all combinations of (α, β, σ) equally probable
- Multiply by likelihood $\rightarrow$ posterior probability distribution over (α, β, σ)



Reviewing GLMs IX: Bayesian model fitting

$$P(\{\beta_i\},\sigma|Y) = \frac{\overbrace{P(Y|\{\beta_i\},\sigma)}^{\text{Likelihood}}\ \overbrace{P(\{\beta_i\},\sigma)}^{\text{Prior}}}{P(Y)}$$

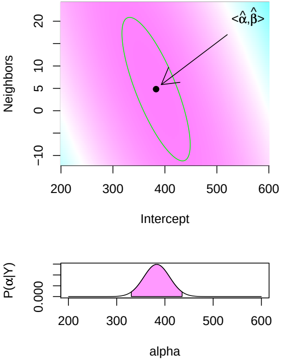
- Alternative to maximum-likelihood: Bayesian model fitting
- Simple (uniform, non-informative) prior: all combinations of (α, β, σ) equally probable
- Multiply by likelihood $\rightarrow$ posterior probability distribution over (α, β, σ)
- Bound the region of highest posterior probability containing 95% of probability density $\rightarrow$ HPD confidence region



Reviewing GLMs IX: Bayesian model fitting

$$P(\{\beta_i\},\sigma|Y) = \frac{\overbrace{P(Y|\{\beta_i\},\sigma)}^{\text{Likelihood}}\overbrace{P(\{\beta_i\},\sigma)}^{\text{Prior}}}{P(Y)}$$

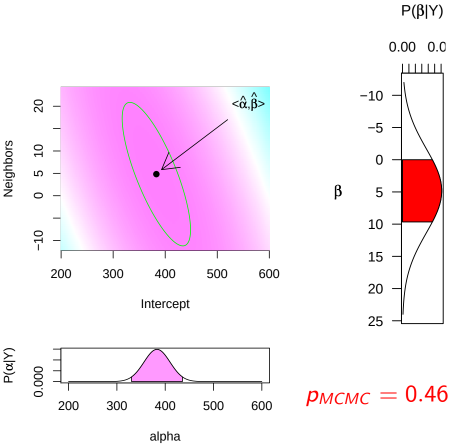
- Alternative to maximum-likelihood: Bayesian model fitting
- Simple (uniform, non-informative) prior: all combinations of (α, β, σ) equally probable
- Multiply by likelihood → posterior probability distribution over (α, β, σ)
- Bound the region of highest posterior probability containing 95% of probability density → HPD confidence region



Reviewing GLMs IX: Bayesian model fitting

$$P(\{\beta_i\},\sigma|Y) = \frac{\overbrace{P(Y|\{\beta_i\},\sigma)}^{\text{Likelihood}}\;\overbrace{P(\{\beta_i\},\sigma)}^{\text{Prior}}}{P(Y)}$$

- Alternative to maximum-likelihood: Bayesian model fitting
- Simple (uniform, non-informative) prior: all combinations of (α, β, σ) equally probable
- Multiply by likelihood $\rightarrow$ posterior probability distribution over (α, β, σ)
- Bound the region of highest posterior probability containing 95% of probability density $\rightarrow$ HPD confidence region



$p_{MCMC} = 0.46$

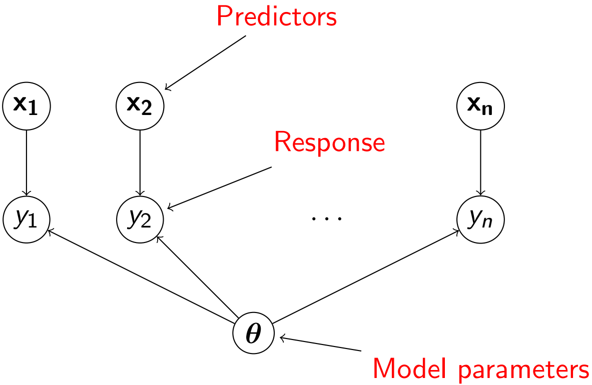
- p_{MCMC} (Baayen et al., 2008) is 1 minus the largest possible symmetric confidence interval wholly on one side of 0

Multi-level Models

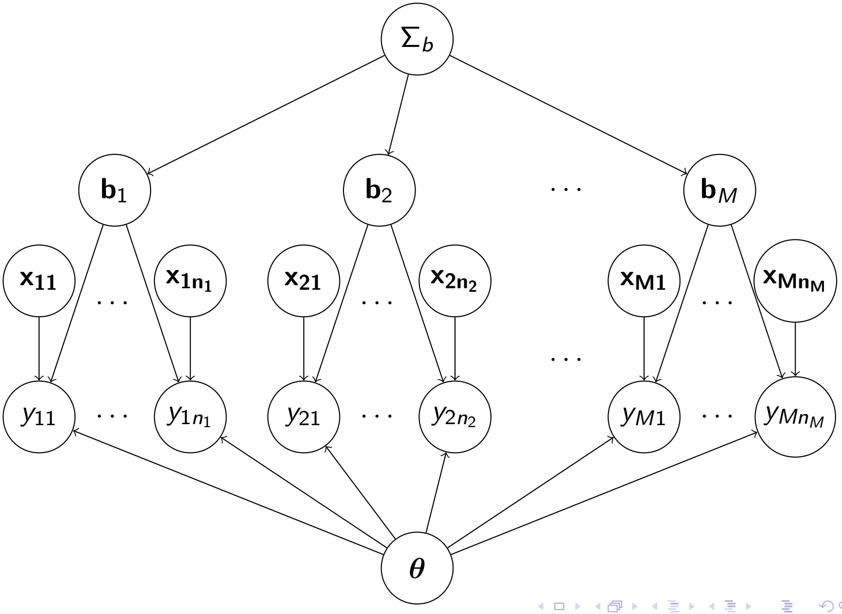
- But of course experiments don't have just one participant
- Different participants may have different idiosyncratic behavior
- And items may have idiosyncratic properties too
- We'd like to take these into account, and perhaps investigate them directly too.
- This is what multi-level (hierarchical, mixed-effects) models are for!

Multi-level Models II

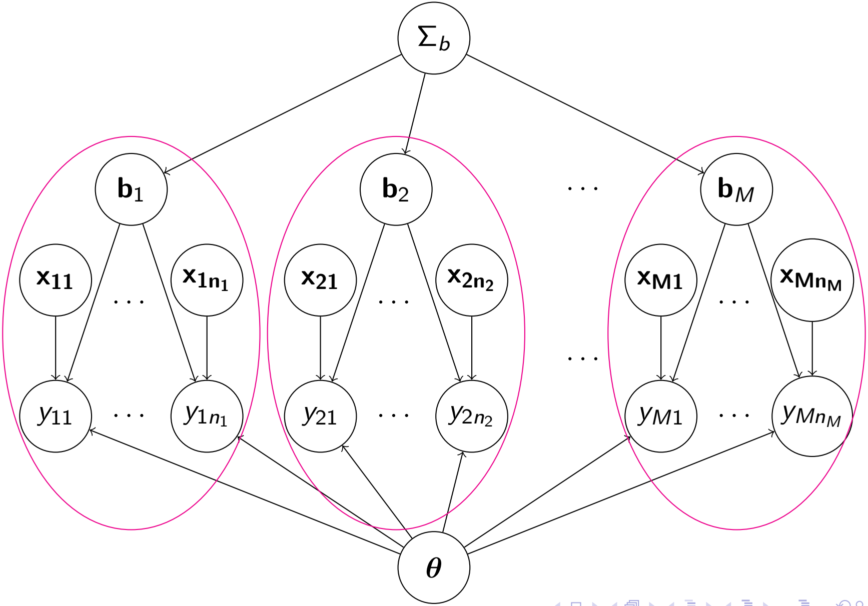
- Recap of the graphical picture of a single-level model:



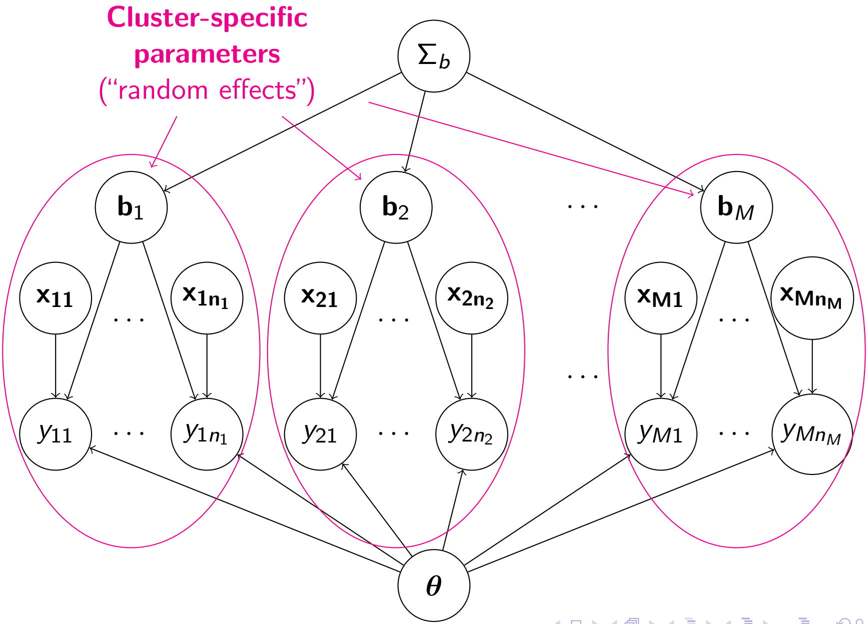
Multi-level Models III: the new graphical picture



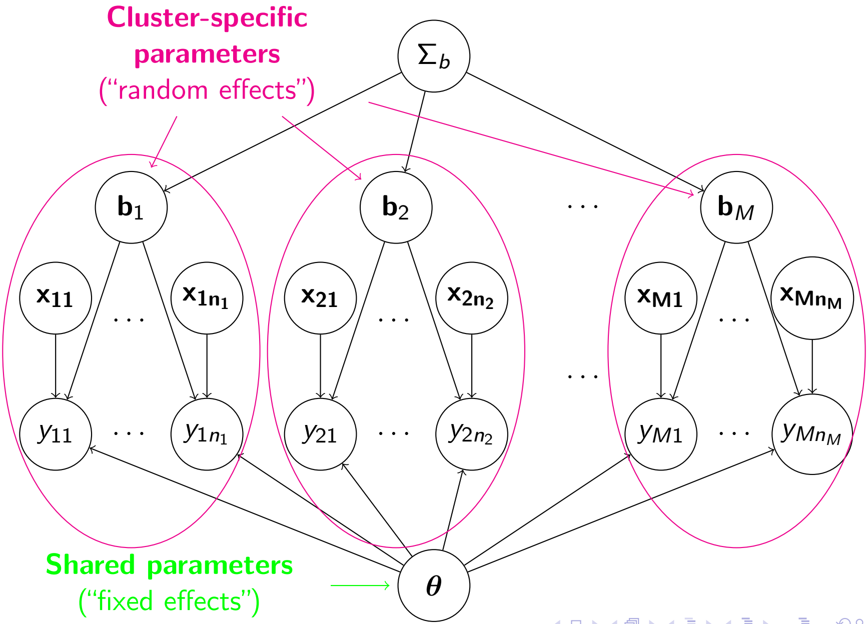
Multi-level Models III: the new graphical picture



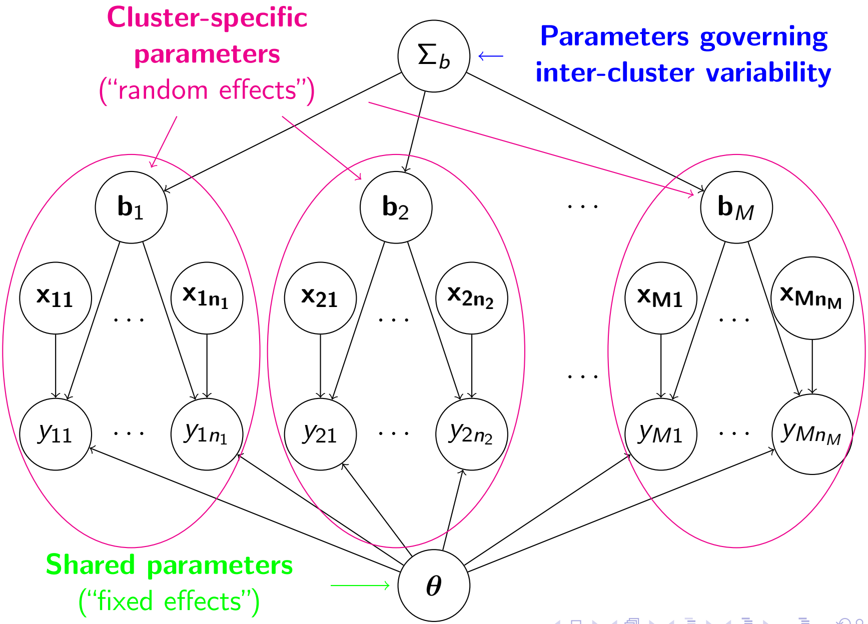
Multi-level Models III: the new graphical picture



Multi-level Models III: the new graphical picture



Multi-level Models III: the new graphical picture



Multi-level Models IV

An example of a multi-level model:

- Back to your lexical-decision experiment

  `tpozt` *Word or non-word?*

  `houze` *Word or non-word?*

- Non-words with different *neighborhood densities* should have different average decision time

Multi-level Models IV

An example of a multi-level model:

- Back to your lexical-decision experiment

`tpozt`	*Word or non-word?*
`houze`	*Word or non-word?*

- Non-words with different *neighborhood densities* should have different average decision time
- Additionally, different participants in your study may also have:
  - different overall decision speeds
  - differing sensitivity to neighborhood density

Multi-level Models IV

An example of a multi-level model:

- Back to your lexical-decision experiment

  `tpozt` *Word or non-word?*

  `houze` *Word or non-word?*

- Non-words with different *neighborhood densities* should have different average decision time
- Additionally, different participants in your study may also have:
  - different overall decision speeds
  - differing sensitivity to neighborhood density
- You want to draw inferences about all these things at the same time

Multi-level Models V: Model construction

- Once again we'll assume for simplicity that the number of word neighbors x has a linear effect on mean reading time, and that trial-level noise is normally distributed*

Multi-level Models V: Model construction

- Once again we'll assume for simplicity that the number of word neighbors x has a linear effect on mean reading time, and that trial-level noise is normally distributed*
- Random effects, starting simple: let each participant i have idiosyncratic differences in reading speed

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

Multi-level Models V: Model construction

- Once again we'll assume for simplicity that the number of word neighbors x has a linear effect on mean reading time, and that trial-level noise is normally distributed*
- Random effects, starting simple: let each participant i have idiosyncratic differences in reading speed

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- In R, we'd write this relationship as

```
RT ~ 1 + x + (1 | participant)
```

Multi-level Models V: Model construction

- Once again we'll assume for simplicity that the number of word neighbors x has a linear effect on mean reading time, and that trial-level noise is normally distributed*
- Random effects, starting simple: let each participant i have idiosyncratic differences in reading speed

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- In R, we'd write this relationship as

```
RT ∼ 1 + x + (1 | participant)
```

- Once again we can leave off the 1, and the noise term ϵ_{ij} is implicit

Multi-level Models V: Model construction

- Once again we'll assume for simplicity that the number of word neighbors x has a linear effect on mean reading time, and that trial-level noise is normally distributed*
- Random effects, starting simple: let each participant i have idiosyncratic differences in reading speed

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- In R, we'd write this relationship as

```
RT ~      x + (1 | participant)
```

- Once again we can leave off the 1, and the noise term ϵ_{ij} is implicit

Multi-level Models VI: simulating data

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- One beauty of multi-level models is that you can simulate trial-level data
- This is invaluable for achieving deeper understanding of both your analysis and your data

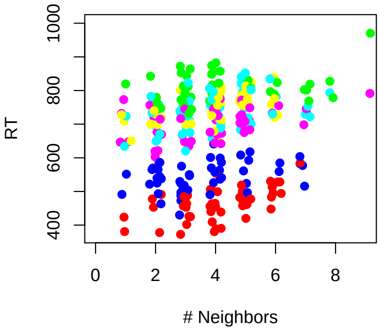
Multi-level Models VI: simulating data

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- One beauty of multi-level models is that you can simulate trial-level data
- This is invaluable for achieving deeper understanding of both your analysis and your data

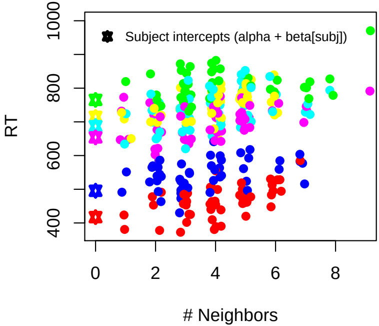
```
## simulate some data
> sigma.b <- 125        # inter-subject variation larger than
> sigma.e <- 40         # intra-subject, inter-trial variation
> alpha <- 500
> beta <- 12
> M <- 6                               # number of participants
> n <- 50                              # trials per participant
> b <- rnorm(M, 0, sigma.b)            # individual differences
> nneighbors <- rpois(M*n,3) + 1       # generate num. neighbors
> subj <- rep(1:M,n)
> RT <- alpha + beta * nneighbors + # simulate RTs!
    b[subj] + rnorm(M*n,0,sigma.e)  #
```

Multi-level Models VII: simulating data



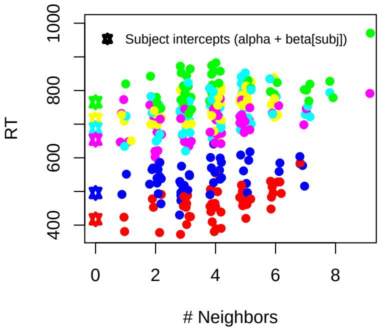
- Participant-level clustering is easily visible

Multi-level Models VII: simulating data



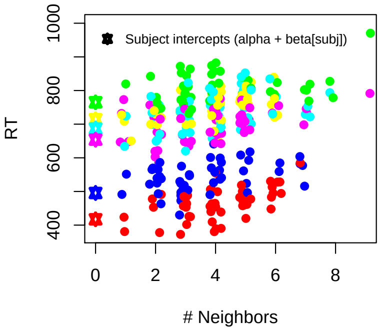
- Participant-level clustering is easily visible

Multi-level Models VII: simulating data



- Participant-level clustering is easily visible
- This reflects the fact that inter-participant variation (125ms) is larger than inter-trial variation (40ms)

Multi-level Models VII: simulating data



- Participant-level clustering is easily visible
- This reflects the fact that inter-participant variation (125ms) is larger than inter-trial variation (40ms)
- And the effects of neighborhood density are also visible

Statistical inference with multi-level models

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- Thus far, we've just defined a model and used it to generate data

Statistical inference with multi-level models

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- Thus far, we've just defined a model and used it to generate data
- We linguists are usually in the opposite situation…

Statistical inference with multi-level models

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- Thus far, we've just defined a model and used it to generate data
- We linguists are usually in the opposite situation. . .
- We *have* data and we need to infer a model
  - Specifically, the "fixed-effect" parameters α, β, and σ_ϵ, plus the parameter governing inter-subject variation, σ_b
  - e.g., hypothesis tests about effects of neighborhood density: can we reliably infer that β is {non-zero, positive, . . . }?

Statistical inference with multi-level models

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise} \sim N(0,\sigma_\epsilon)}$$

- Thus far, we've just defined a model and used it to generate data
- We linguists are usually in the opposite situation...
- We *have* data and we need to infer a model
  - Specifically, the "fixed-effect" parameters α, β, and σ_ϵ, plus the parameter governing inter-subject variation, σ_b
  - e.g., hypothesis tests about effects of neighborhood density: can we reliably infer that β is {non-zero, positive, ...}?
- Fortunately, we can use the same principles as before to do this:
  - The principle of maximum likelihood
  - Or Bayesian inference

Fitting a multi-level model using maximum likelihood

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

```
> m <- lmer(time ~ neighbors.centered +
    (1 | participant),dat,REML=F)
> print(m,corr=F)

[...]
Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept)  4924.9   70.177
 Residual                19240.5  138.710
Number of obs: 1760, groups: participant, 44

Fixed effects:
                   Estimate Std. Error t value
(Intercept)         583.787     11.082   52.68
neighbors.centered    8.986      1.278    7.03
```

Fitting a multi-level model using maximum likelihood

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

```
> m <- lmer(time ~ neighbors.centered +
    (1 | participant),dat,REML=F)
> print(m,corr=F)

[...]
Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept)  4924.9   70.177
 Residual                19240.5  138.710
Number of obs: 1760, groups: participant, 44

Fixed effects:
                   Estimate Std. Error t value
(Intercept)         583.787     11.082   52.68
neighbors.centered    8.986      1.278    7.03
```

($\hat{\alpha}$ → 583.787)

Fitting a multi-level model using maximum likelihood

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

```
> m <- lmer(time ~ neighbors.centered +
    (1 | participant),dat,REML=F)
> print(m,corr=F)

[...]
Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept)  4924.9   70.177
 Residual                19240.5  138.710
Number of obs: 1760, groups: participant, 44

Fixed effects:
            α̂ →        Estimate Std. Error t value
(Intercept)            583.787     11.082   52.68
neighbors.centered       8.986      1.278    7.03
                                  ← β̂
```

Fitting a multi-level model using maximum likelihood

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

```
> m <- lmer(time ~ neighbors.centered +
    (1 | participant),dat,REML=F)
> print(m,corr=F)

[...]
Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept)  4924.9   70.177
 Residual                19240.5  138.710
Number of obs: 1760, groups: participant, 44

Fixed effects:
                   Estimate Std. Error t value
(Intercept)         583.787     11.082   52.68
neighbors.centered    8.986      1.278    7.03
```

$\widehat{\sigma}_b$ → 70.177

$\widehat{\alpha}$ → 583.787

$\widehat{\beta}$ → 8.986

Fitting a multi-level model using maximum likelihood

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

```
> m <- lmer(time ~ neighbors.centered +
    (1 | participant),dat,REML=F)
> print(m,corr=F)

[...]
Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept)  4924.9   70.177
 Residual                19240.5  138.710
Number of obs: 1760, groups: participant, 44

Fixed effects:
                   Estimate Std. Error t value
(Intercept)         583.787     11.082   52.68
neighbors.centered    8.986      1.278    7.03
```

$\widehat{\sigma}_b$ → 70.177

$\widehat{\sigma}_\epsilon$ → 138.710

$\widehat{\alpha}$ → 583.787

$\widehat{\beta}$ → 8.986

Interpreting parameter estimates

Intercept	583.79
neighbors.centered	8.99
$\widehat{\sigma}_b$	70.18
$\widehat{\sigma}_\epsilon$	138.7

Interpreting parameter estimates

`Intercept`	583.79
`neighbors.centered`	8.99
$\widehat{\sigma}_b$	70.18
$\widehat{\sigma}_\epsilon$	138.7

- The *fixed effects* are interpreted just as in a traditional single-level model:

Interpreting parameter estimates

`Intercept`	583.79
`neighbors.centered`	8.99
$\widehat{\sigma}_b$	70.18
$\widehat{\sigma}_\epsilon$	138.7

- The *fixed effects* are interpreted just as in a traditional single-level model:
  - The ‘base’ RT for a non-word in this study is 583.79ms

Interpreting parameter estimates

`Intercept`	583.79
`neighbors.centered`	8.99
$\widehat{\sigma}_b$	70.18
$\widehat{\sigma}_\epsilon$	138.7

- The *fixed effects* are interpreted just as in a traditional single-level model:
  - The 'base' RT for a non-word in this study is 583.79ms
  - Every extra neighbor increases 'base' RT by 8.99ms

Interpreting parameter estimates

`Intercept`	583.79
`neighbors.centered`	8.99
$\widehat{\sigma}_b$	70.18
$\widehat{\sigma}_\epsilon$	138.7

- The *fixed effects* are interpreted just as in a traditional single-level model:
  - The 'base' RT for a non-word in this study is 583.79ms
  - Every extra neighbor increases 'base' RT by 8.99ms
- Inter-trial variability σ_ϵ also has the same interpretation

Interpreting parameter estimates

Intercept	583.79
neighbors.centered	8.99
$\widehat{\sigma}_b$	70.18
$\widehat{\sigma}_\epsilon$	138.7

- The *fixed effects* are interpreted just as in a traditional single-level model:
  - The 'base' RT for a non-word in this study is 583.79ms
  - Every extra neighbor increases 'base' RT by 8.99ms
- Inter-trial variability σ_ϵ also has the same interpretation
  - Inter-trial variability for a given participant is Gaussian, centered around the participant+word-specific mean with standard deviation 138.7ms

Interpreting parameter estimates

`Intercept`	583.79
`neighbors.centered`	8.99
$\widehat{\sigma}_b$	70.18
$\widehat{\sigma}_\epsilon$	138.7

- The *fixed effects* are interpreted just as in a traditional single-level model:
  - The 'base' RT for a non-word in this study is 583.79ms
  - Every extra neighbor increases 'base' RT by 8.99ms
- Inter-trial variability σ_ϵ also has the same interpretation
  - Inter-trial variability for a given participant is Gaussian, centered around the participant+word-specific mean with standard deviation 138.7ms
- Inter-participant variability σ_b is what's new:

Interpreting parameter estimates

`Intercept`	583.79
`neighbors.centered`	8.99
$\widehat{\sigma}_b$	70.18
$\widehat{\sigma}_\epsilon$	138.7

- The *fixed effects* are interpreted just as in a traditional single-level model:
  - The 'base' RT for a non-word in this study is 583.79ms
  - Every extra neighbor increases 'base' RT by 8.99ms
- Inter-trial variability σ_ϵ also has the same interpretation
  - Inter-trial variability for a given participant is Gaussian, centered around the participant+word-specific mean with standard deviation 138.7ms
- Inter-participant variability σ_b is what's new:
  - Variability in average RT in the population from which the participants were drawn has standard deviation 70.18ms

Inferences about cluster-level parameters

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- What about the participants' idiosyncracies themselves—the b_i?

Inferences about cluster-level parameters

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- What about the participants' idiosyncracies themselves—the b_i?
- We can also draw inferences about these—you may have heard about BLUPs

Inferences about cluster-level parameters

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- What about the participants' idiosyncracies themselves—the b_i?
- We can also draw inferences about these—you may have heard about BLUPs
- To understand these: committing to fixed-effect and random-effect parameter estimates determines a conditional probability distribution on participant-specific effects:

$$P(b_i|\widehat{\alpha}, \widehat{\beta}, \widehat{\sigma}_b, \widehat{\sigma}_\epsilon)$$

Inferences about cluster-level parameters

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_i}^{\sim N(0,\sigma_b)} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

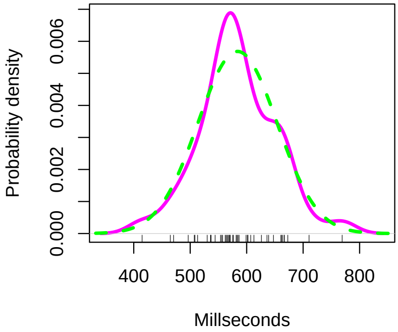
- What about the participants' idiosyncracies themselves—the b_i?
- We can also draw inferences about these—you may have heard about BLUPs
- To understand these: committing to fixed-effect and random-effect parameter estimates determines a conditional probability distribution on participant-specific effects:

$$P(b_i|\widehat{\alpha}, \widehat{\beta}, \widehat{\sigma}_b, \widehat{\sigma}_\epsilon)$$

- The BLUPS are the conditional modes of b_i—the choices that maximize the above probability

Inferences about cluster-level parameters II

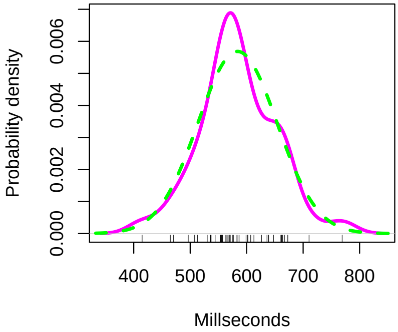
- The BLUP participant-specific 'base' RTs for this dataset are black lines on the base of this graph



  - The solid line is a guess at their distribution

Inferences about cluster-level parameters II

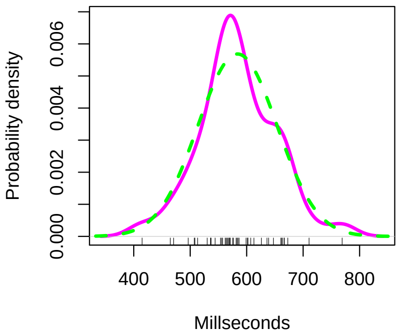
- The BLUP participant-specific 'base' RTs for this dataset are black lines on the base of this graph



  - The solid line is a guess at their distribution
  - The dotted line is the distribution predicted by the model for the population from which the participants are drawn

Inferences about cluster-level parameters II

- The BLUP participant-specific ‘base’ RTs for this dataset are black lines on the base of this graph



  - The solid line is a guess at their distribution
  - The dotted line is the distribution predicted by the model for the population from which the participants are drawn
  - Reasonably close correspondence

Inference about cluster-level parameters III

- Participants may also have idiosyncratic sensitivities to *neighborhood density*

Inference about cluster-level parameters III

- Participants may also have idiosyncratic sensitivities to *neighborhood density*
- Incorporate by adding cluster-level slopes into the model:

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_{1i} + b_{2i}}^{\sim N(0,\Sigma_b)} x_{ij} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

Inference about cluster-level parameters III

- Participants may also have idiosyncratic sensitivities to *neighborhood density*
- Incorporate by adding cluster-level slopes into the model:

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_{1i} + b_{2i}}^{\sim N(0,\Sigma_b)} x_{ij} + \overbrace{\epsilon_{ij}}^{\text{Noise} \sim N(0,\sigma_\epsilon)}$$

- In `R` (once again we can omit the `1`'s):

```
RT ∼ 1 + x + (1 + x | participant)
```

Inference about cluster-level parameters III

- Participants may also have idiosyncratic sensitivities to *neighborhood density*
- Incorporate by adding cluster-level slopes into the model:

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_{1i} + b_{2i}}^{\sim N(0,\Sigma_b)} x_{ij} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

- In R (once again we can omit the 1's):

$$\text{RT} \sim 1 + x + (1 + x \mid \text{participant})$$

```
> lmer(RT ~ neighbors.centered +
    (neighbors.centered | participant), dat,REML=F)

[...]
Random effects:
 Groups      Name                Variance  Std.Dev. Corr
 participant (Intercept)          4928.625  70.2042
             neighbors.centered     19.421   4.4069 -0.307
 Residual                        19107.143 138.2286
```

Inference about cluster-level parameters III

- Participants may also have idiosyncratic sensitivities to *neighborhood density*
- Incorporate by adding cluster-level slopes into the model:

$$RT_{ij} = \alpha + \beta x_{ij} + \overbrace{b_{1i} + b_{2i}}^{\sim N(0,\Sigma_b)} x_{ij} + \overbrace{\epsilon_{ij}}^{\text{Noise}\sim N(0,\sigma_\epsilon)}$$

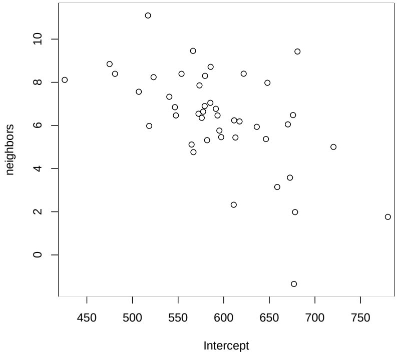
- In R (once again we can omit the 1's):

  RT ∼ 1 + x + (1 + x | participant)

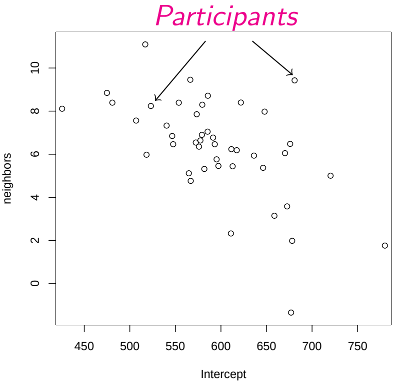
```
> lmer(RT ~ neighbors.centered +
    (neighbors.centered | participant), dat,REML=F)

[...]
Random effects:
 Groups      Name               Variance  Std.Dev. Corr
 participant (Intercept)         4928.625  70.2042
             neighbors.centered    19.421   4.4069 -0.307
 Residual                       19107.143 138.2286
```

These three numbers jointly characterize $\widehat{\Sigma}_b$

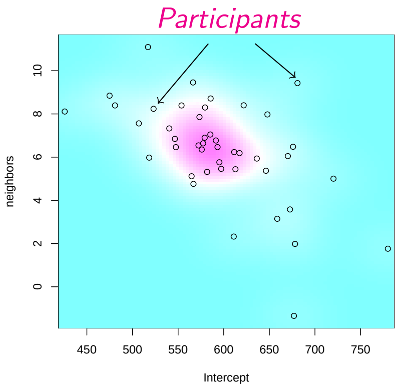
Inference about cluster-level parameters IV



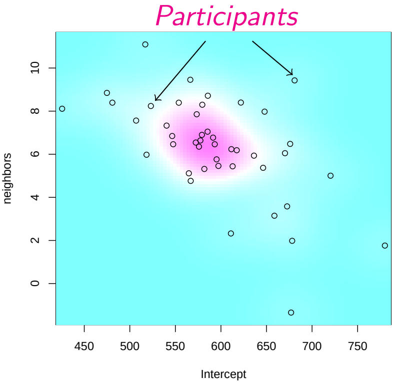
Inference about cluster-level parameters IV



Inference about cluster-level parameters IV

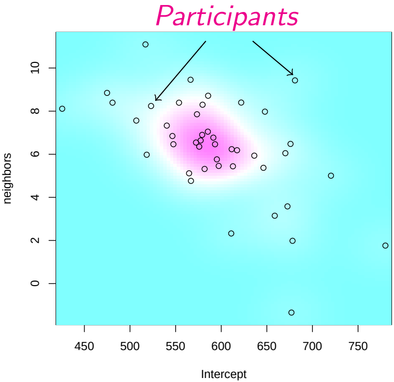


Inference about cluster-level parameters IV



- Correlation visible in participant-specific BLUPs

Inference about cluster-level parameters IV



- Correlation visible in participant-specific BLUPs
- Participants who were faster overall also tend to be more affected by neighborhood density

$$\widehat{\Sigma} = \begin{pmatrix} 70.20 & -0.3097 \\ -0.3097 & 4.41 \end{pmatrix}$$

Bayesian inference for multilevel models

$$P(\{\beta_i\}, \sigma_b, \sigma_\epsilon | Y) = \frac{\overbrace{P(Y|\{\beta_i\}, \sigma_b, \sigma_\epsilon)}^{\text{Likelihood}} \overbrace{P(\{\beta_i\}, \sigma_b, \sigma_\epsilon)}^{\text{Prior}}}{P(Y)}$$

- We can also use Bayes' rule to draw inferences about fixed effects

Bayesian inference for multilevel models

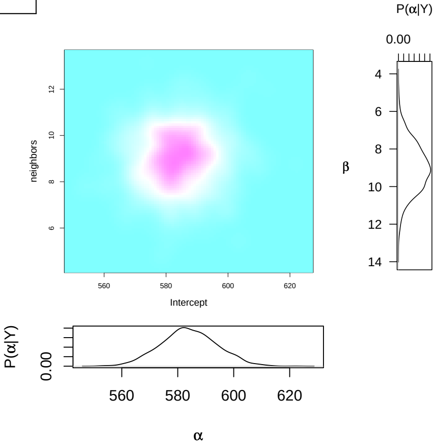
$$P(\{\beta_i\}, \sigma_b, \sigma_\epsilon | Y) = \frac{\overbrace{P(Y|\{\beta_i\}, \sigma_b, \sigma_\epsilon)}^{\text{Likelihood}} \overbrace{P(\{\beta_i\}, \sigma_b, \sigma_\epsilon)}^{\text{Prior}}}{P(Y)}$$

- We can also use Bayes' rule to draw inferences about fixed effects
- Computationally more challenging than with single-level regression; Markov-chain Monte Carlo (MCMC) sampling techniques allow us to approximate it

Bayesian inference for multilevel models

$$P(\{\beta_i\}, \sigma_b, \sigma_\epsilon | Y) = \frac{\overbrace{P(Y|\{\beta_i\}, \sigma_b, \sigma_\epsilon)}^{\text{Likelihood}}\,\overbrace{P(\{\beta_i\}, \sigma_b, \sigma_\epsilon)}^{\text{Prior}}}{P(Y)}$$

- We can also use Bayes' rule to draw inferences about fixed effects
- Computationally more challenging than with single-level regression; Markov-chain Monte Carlo (MCMC) sampling techniques allow us to approximate it



Why do you care???

- You may be asking yourself:

  Why did I come to this workshop? I could do everything you just did with an ANCOVA, treating participant as a random factor, or by looking at participant means.

Why do you care??? II

Why did I come to this workshop? I could do everything you just did with an ANCOVA, treating participant as a random factor, or by looking at participant means.

Why do you care??? II

Why did I come to this workshop? I could do everything you just did with an ANCOVA, treating participant as a random factor, or by looking at participant means.

- Yes, but there are several respects in which multi-level models go beyond AN(C)OVA:

Why do you care??? II

Why did I come to this workshop? I could do everything you just did with an ANCOVA, treating participant as a random factor, or by looking at participant means.

- Yes, but there are several respects in which multi-level models go beyond AN(C)OVA:
    1. They handle *imbalanced datasets* just as well as balanced datasets

Why do you care??? II

> Why did I come to this workshop? I could do everything you just did with an ANCOVA, treating participant as a random factor, or by looking at participant means.

- Yes, but there are several respects in which multi-level models go beyond AN(C)OVA:
    1. They handle *imbalanced datasets* just as well as balanced datasets
    2. You can use non-linear linking functions (e.g., logit models for binary-choice data)

Why do you care??? II

Why did I come to this workshop? I could do everything you just did with an ANCOVA, treating participant as a random factor, or by looking at participant means.

- Yes, but there are several respects in which multi-level models go beyond AN(C)OVA:
  1. They handle *imbalanced datasets* just as well as balanced datasets
  2. You can use non-linear linking functions (e.g., logit models for binary-choice data)
  3. You can cross cluster-level effects
     - Every trial belongs to both a participant cluster and an item cluster

Why do you care??? II

> Why did I come to this workshop? I could do everything you just did with an ANCOVA, treating participant as a random factor, or by looking at participant means.

- Yes, but there are several respects in which multi-level models go beyond AN(C)OVA:
  1. They handle *imbalanced datasets* just as well as balanced datasets
  2. You can use non-linear linking functions (e.g., logit models for binary-choice data)
  3. You can cross cluster-level effects
     - Every trial belongs to both a participant cluster and an item cluster
     - You can build a single unified model for inferences from your data

Why do you care??? II

Why did I come to this workshop? I could do everything you just did with an ANCOVA, treating participant as a random factor, or by looking at participant means.

- Yes, but there are several respects in which multi-level models go beyond AN(C)OVA:
  1. They handle *imbalanced datasets* just as well as balanced datasets
  2. You can use non-linear linking functions (e.g., logit models for binary-choice data)
  3. You can cross cluster-level effects
     - Every trial belongs to both a participant cluster and an item cluster
     - You can build a single unified model for inferences from your data
     - ANOVA requires separate by-participants and by-items analyses (quasi-F' is quite conservative)

Why do you care??? II

> Why did I come to this workshop? I could do everything you just did with an ANCOVA, treating participant as a random factor, or by looking at participant means.

- Yes, but there are several respects in which multi-level models go beyond AN(C)OVA:
  1. They handle *imbalanced datasets* just as well as balanced datasets
  2. You can use non-linear linking functions (e.g., logit models for binary-choice data)
  3. You can cross cluster-level effects
     - Every trial belongs to both a participant cluster and an item cluster
     - You can build a single unified model for inferences from your data
     - ANOVA requires separate by-participants and by-items analyses (quasi-F' is quite conservative)

Summary

- Multi-level models may seem strange and foreign
- But all you really need to understand them is three basic things
    - Generalized linear models
    - The principle of maximum likelihood
    - Bayesian inference
- As you will see in the rest of the workshop, these models open up many new interesting doors!

References I

Agresti, A. (2002). *Categorical Data Analysis*. Wiley, second edition.

Baayen, R. H., Davidson, D. J., and Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*. In press.

Bicknell, K., Elman, J. L., Hare, M., McRae, K., and Kutas, M. (2008). Online expectations for verbal arguments conditional on event knowledge. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pages 2220–2225.

Jaeger, T. F. (2008). Categorical data analysis: Away from ANOVAs (transformation or not) and towards logit mixed models. *Journal of Memory and Language*. In press.