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

Cluster-specific parameters ("random effects")

Parameters governing inter-cluster variability

Shared parameters ("fixed effects")

Roger Levy
UC San Diego
Department of Linguistics

25 March 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
- Briefly discuss multi-level logit models
Goal: model the effects of predictors (independent variables) $X$ on a response (dependent variable) $Y$. 
Goal: model the effects of predictors (independent variables) $X$ on a response (dependent variable) $Y$.

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

The picture:
Goal: model the effects of predictors (independent variables) $X$ on a response (dependent variable) $Y$.

The picture:
Assumptions of the generalized linear model (GLM):

1. Predictors \( \{X_i\} \) influence \( Y \) through the mediation of a linear predictor \( \eta \);
Assumptions of the generalized linear model (GLM):

1. Predictors \( \{X_i\} \) influence \( Y \) through the mediation of a linear predictor \( \eta \);

2. \( \eta \) is a linear combination of the \( \{X_i\} \):
Assumptions of the generalized linear model (GLM):

1. Predictors \( \{ X_i \} \) influence \( Y \) through the mediation of a linear predictor \( \eta \);
2. \( \eta \) is a linear combination of the \( \{ X_i \} \):
   \[
   \eta = \alpha + \beta_1 X_1 + \cdots + \beta_N X_N \quad \text{(linear predictor)}
   \]
Assumptions of the generalized linear model (GLM):

1. Predictors \( \{ X_i \} \) influence \( Y \) through the mediation of a linear predictor \( \eta \);
2. \( \eta \) is a linear combination of the \( \{ X_i \} \):
   \[
   \eta = \alpha + \beta_1 X_1 + \cdots + \beta_N X_N \quad \text{(linear predictor)}
   \]
3. \( \eta \) determines the predicted mean \( \mu \) of \( Y \)
   \[
   \eta = l(\mu) \quad \text{(link function)}
   \]
Assumptions of the generalized linear model (GLM):

1. Predictors \( \{X_i\} \) influence \( Y \) through the mediation of a linear predictor \( \eta \);

2. \( \eta \) is a linear combination of the \( \{X_i\} \):
   \[
   \eta = \alpha + \beta_1 X_1 + \cdots + \beta_N X_N \quad \text{(linear predictor)}
   \]

3. \( \eta \) determines the predicted mean \( \mu \) of \( Y \)
   \[
   \eta = l(\mu) \quad \text{(link function)}
   \]

4. There is some noise distribution of \( Y \) around the predicted mean \( \mu \) of \( Y \):
   \[
   P(Y = y; \mu)
   \]
Linear regression, which underlies ANOVA, is a kind of generalized linear model.
Linear regression, which underlies ANOVA, is a kind of generalized linear model.

- The predicted mean is just the linear predictor:

\[ \eta = I(\mu) = \mu \]
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 \(\sigma\):
  \[
  \epsilon \sim N(0, \sigma)
  \]
Linear regression, which underlies ANOVA, is a kind of generalized linear model.

- The predicted mean is just the linear predictor:
  \[ \eta = I(\mu) = \mu \]

- Noise is normally (=Gaussian) distributed around 0 with standard deviation \( \sigma \):
  \[ \epsilon \sim N(0, \sigma) \]

- This gives us the traditional linear regression equation:
  \[
  Y = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n + \epsilon
  \]
How do we fit the parameters $\beta_i$ and $\sigma$ (**choose model coefficients**)?

There are two major approaches (deeply related, yet different) in widespread use:
How do we fit the parameters $\beta_i$ and $\sigma$ \textit{(choose model coefficients)}?

There are two major approaches (deeply related, yet different) in widespread use:

- The principle of \textit{maximum likelihood}: pick parameter values that maximize the probability of your data $Y$

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

- Bayesian inference: put a probability distribution on the model parameters and update it on the basis of what parameters best explain the data.
How do we fit the parameters $\beta_i$ and $\sigma$ (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 $\sigma$ 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
How do we fit the parameters $\beta_i$ and $\sigma$ (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 $\sigma$ 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

$$P(\{\beta_i\}, \sigma|Y) = \frac{P(Y|\{\beta_i\}, \sigma)P(\{\beta_i\}, \sigma)}{P(Y)}$$
How do we fit the parameters $\beta_i$ and $\sigma$ (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 $\sigma$ 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

$P(\{\beta_i\}, \sigma|Y) = \frac{P(Y|\{\beta_i\}, \sigma)P(\{\beta_i\}, \sigma)}{P(Y)}$
You are studying non-word RTs in a lexical-decision task
You are studying non-word RTs in a lexical-decision task.

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

We need to draw inferences about $\alpha$, $\beta$, and $\sigma$.

Example: “Does neighborhood density affects RT?” → is $\beta$ reliably non-zero?
You are studying non-word RTs in a lexical-decision task

- tpozt  Word or non-word?
- houze  Word or non-word?
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)*
You are studying non-word RTs in a lexical-decision task

- tpozt  \textit{Word or non-word?}
- houze  \textit{Word or non-word?}

Non-words with different \textit{neighborhood densities}\textsuperscript{*} should have different average RT \textsuperscript*\((=\text{number of neighbors of edit-distance 1})\)

A simple model: assume that neighborhood density has a \textit{linear} effect on average RT, and trial-level noise is \textit{normally distributed}\textsuperscript{*} \textsuperscript*\((\text{n.b. wrong–RTs are skewed—but not horrible.})\)
You are studying non-word RTs in a lexical-decision task

- **tpozt**  \textit{Word or non-word?}
- **houze** \textit{Word or non-word?}

Non-words with different \textit{neighborhood densities} should have different average RT \(^* (= \text{number of neighbors of edit-distance 1})\)

A simple model: assume that neighborhood density has a \textit{linear} effect on average RT, and trial-level noise is \textit{normally distributed} \(^* (= \text{n.b. wrong–RTs are skewed—but not horrible.})\)

If \(x_i\) is neighborhood density, our simple model is

\[
RT_i = \alpha + \beta x_i + \epsilon_i \sim N(0, \sigma)
\]
Reviewing GLMs V: a simple example

- You are studying non-word RTs in a lexical-decision task
  
  tpozt  \textit{Word or non-word?}
  
  houze  \textit{Word or non-word?}

- Non-words with different \textit{neighborhood densities}\footnote{should have different average RT \footnote{\textit{(= number of neighbors of edit-distance 1)}}} should have different average RT \footnote{\textit{(n.b. wrong--RTs are skewed—but not horrible.)}}

- A simple model: assume that neighborhood density has a \textit{linear} effect on average RT, and trial-level noise is \textit{normally distributed} \footnote{\textit{*(n.b. wrong--RTs are skewed—but not horrible.)}}

- If \(x_i\) is neighborhood density, our simple model is

\[
RT_i = \alpha + \beta x_i + \epsilon_i \sim N(0, \sigma)
\]

- We need to draw inferences about \(\alpha\), \(\beta\), and \(\sigma\)
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
  \[
  RT_i = \alpha + \beta x_i + \epsilon_i \sim N(0,\sigma)
  \]

- We need to draw inferences about $\alpha$, $\beta$, and $\sigma$
- e.g., “Does neighborhood density affects RT?" → is $\beta$ reliably non-zero?
We’ll use length-4 nonword data from (Bicknell et al., 2008) (thanks!), such as:

- Few neighbors: gaty, peme, rixy
- Many neighbors: lish, pait, yine
We’ll use length-4 nonword data from (Bicknell et al., 2008) (thanks!), such as:

- Few neighbors
  - gaty
  - pem
  - rixy
- Many neighbors
  - lish
  - pait
  - yine

There’s a wide range of neighborhood density:
Here’s a translation of our simple model into R:

```r
RT \sim 1 + x
```
Reviewing GLMs VII: maximum-likelihood model fitting

\[ RT_i = \alpha + \beta X_i + \epsilon_i \sim N(0, \sigma) \]

- Here’s a translation of our simple model into R:
  \[ RT \sim 1 + x \]
- The noise is implicit in asking R to fit a *linear* model
Here’s a translation of our simple model into R:

\[
RT \sim 1 + x
\]

The noise is implicit in asking R to fit a \textit{linear} model

(We can omit the 1; R assumes it unless otherwise directed)
Here’s a translation of our simple model into R:

\[
\text{RT} \sim x
\]

The noise is implicit in asking R to fit a \textit{linear} model

(We can omit the 1; R assumes it unless otherwise directed)
Here’s a translation of our simple model into R:

\[ RT_i = \alpha + \beta X_i + \epsilon_i \]

- 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)

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

Gaussian noise, implicit intercept

[...]

(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
```
Here’s a translation of our simple model into R:

\[
RT \sim x
\]

The noise is implicit in asking R to fit a \textit{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)

```r
> 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
Here’s a translation of our simple model into R:

\[
RT_i = \alpha + \beta X_i + \epsilon_i \sim N(0,\sigma)
\]

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)

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

[...]

\[\hat{\alpha}\] Estimate Std. Error t value Pr(>|t|)
(Intercept) \textbf{382.997} \quad \textbf{26.837} \quad \textbf{14.271} \quad <2e-16 \quad ***
neighbors \quad \textbf{4.828} \quad \textbf{6.553} \quad \textbf{0.737} \quad 0.466

> sqrt(summary(m)[["dispersion"]])

[1] 107.2248
Reviewing GLMs VII: maximum-likelihood model fitting

\[ RT_i = \alpha + \beta X_i + \epsilon_i \sim N(0, \sigma) \]

- 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)

```r
> 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
```

\( \hat{\alpha} \quad \hat{\beta} \quad \hat{\sigma} \)
Here’s a translation of our simple model into R:

\[ RT \sim x \]

The noise is implicit in asking R to fit a \textit{linear} model.

(We can omit the \texttt{1}; R assumes it unless otherwise directed).

Example of fitting via maximum likelihood: one subject from Bicknell et al. (2008).

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

[...]

\[ \hat{\alpha} \quad \hat{\beta} \]

(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
```

\( \hat{\alpha} \) \quad \hat{\beta} \quad \hat{\sigma}
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>383.00</td>
</tr>
<tr>
<td>neighbors</td>
<td>4.83</td>
</tr>
<tr>
<td>$\hat{\sigma}$</td>
<td>107.22</td>
</tr>
</tbody>
</table>
Estimated coefficients are what underlies “best linear fit” plots.
Estimated coefficients are what underlies “best linear fit” plots.
Reviewing GLMs IX: Bayesian model fitting

\[ P(\{\beta_i\}, \sigma \mid Y) = \frac{\text{Likelihood} \times \text{Prior}}{P(Y)} \]

- Alternative to maximum-likelihood:
  Bayesian model fitting
Reviewing GLMs IX: Bayesian model fitting

\[
P(\{\beta_i\}, \sigma \mid Y) = \frac{\text{Likelihood}}{P(\{\beta_i\}, \sigma \mid Y)} \cdot \frac{\text{Prior}}{P(\{\beta_i\}, \sigma)}
\]

▶ Alternative to maximum-likelihood: Bayesian model fitting

▶ Simple (uniform, non-informative) prior: all combinations of \((\alpha, \beta, \sigma)\) equally probable
Reviewing GLMs IX: Bayesian model fitting

\[ P(\{\beta_i\}, \sigma | Y) = \frac{P(Y|\{\beta_i\}, \sigma) P(\{\beta_i\}, \sigma)}{P(Y)} \]

- Alternative to maximum-likelihood: Bayesian model fitting

- Simple (uniform, non-informative) prior: all combinations of \((\alpha, \beta, \sigma)\) equally probable

- Multiply by likelihood → posterior probability distribution over \((\alpha, \beta, \sigma)\)
Reviewing GLMs IX: Bayesian model fitting

\[ P(\{\beta_i\}, \sigma | Y) = \frac{\text{Likelihood} \times \text{Prior}}{P(Y)} \]

- Alternative to maximum-likelihood: Bayesian model fitting
- Simple (uniform, non-informative) prior: all combinations of \((\alpha, \beta, \sigma)\) equally probable
- Multiply by likelihood \(\rightarrow\) posterior probability distribution over \((\alpha, \beta, \sigma)\)
Reviewing GLMs IX: Bayesian model fitting

\[ P(\{\beta_i\}, \sigma | Y) = \frac{\text{Likelihood} \cdot \text{Prior}}{P(Y)} \]

- **Alternative to maximum-likelihood:** Bayesian model fitting

- **Simple** (uniform, non-informative) prior: all combinations of \((\alpha, \beta, \sigma)\) equally probable

- Multiply by likelihood → posterior probability distribution over \((\alpha, \beta, \sigma)\)

- Bound the region of highest posterior probability containing 95% of probability density → HPD confidence region
Reviewing GLMs IX: Bayesian model fitting

\[ P(\{\beta_i\}, \sigma | Y) = \frac{\text{Likelihood} \cdot \text{Prior}}{P(Y)} \]

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

\[ P(\alpha | Y) \]

\[ p_{\text{MCMC}} (\text{Baayen et al., 2008}) \text{ is } 1 \text{ minus the largest possible symmetric confidence interval wholly on one side of 0} \]
Reviewing GLMs IX: Bayesian model fitting

\[
P(\{\beta_i\}, \sigma | Y) = \frac{\text{Likelihood} P(Y|\{\beta_i\}, \sigma) P(\{\beta_i\}, \sigma)}{P(Y)}
\]

- Alternative to maximum-likelihood: Bayesian model fitting

- Simple (uniform, non-informative) prior: all combinations of \((\alpha, \beta, \sigma)\) equally probable

- Multiply by likelihood → posterior probability distribution over \((\alpha, \beta, \sigma)\)

- Bound the region of highest posterior probability containing 95% of probability density → HPD confidence region

\[p_{\text{MCMC}} = 0.46\]

\(p_{\text{MCMC}}\) (Baayen et al., 2008) is 1 minus the largest possible symmetric confidence interval wholly on one side of 0.
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!
Recap of the graphical picture of a single-level model:
Multi-level Models III: the new graphical picture

Cluster-specific parameters ("random effects")

Shared parameters ("fixed effects")

Parameters governing inter-cluster variability
Multi-level Models III: the new graphical picture

Cluster-specific parameters ("random effects")

Shared parameters ("fixed effects")

Parameters governing inter-cluster variability
Multi-level Models III: the new graphical picture

Cluster-specific parameters ("random effects")

\[ \theta \]

\[ \Sigma_b \]

\[ b_1 \]

\[ b_2 \]

\[ \ldots \]

\[ b_M \]

\[ x_{11} \]

\[ \ldots \]

\[ x_{1n_1} \]

\[ y_{11} \]

\[ \ldots \]

\[ y_{1n_1} \]

\[ x_{21} \]

\[ \ldots \]

\[ x_{2n_2} \]

\[ y_{21} \]

\[ \ldots \]

\[ y_{2n_2} \]

\[ x_{M1} \]

\[ \ldots \]

\[ x_{Mn_M} \]

\[ y_{M1} \]

\[ \ldots \]

\[ y_{Mn_M} \]
Cluster-specific parameters ("random effects")

Shared parameters ("fixed effects")
Multi-level Models III: the new graphical picture

Cluster-specific parameters ("random effects")

Parameters governing inter-cluster variability

Shared parameters ("fixed effects")
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
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
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
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*
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

$$RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij}$$

\begin{align*}
\sim N(0,\sigma_b) & \quad \text{Noise} \sim N(0,\sigma_\epsilon)
\end{align*}
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

$$RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij}$$

In R, we’d write this relationship as

$$RT \sim 1 + x + (1 \mid participant)$$
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

$$RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij}$$

Noise $\sim N(0, \sigma_e)$

In R, we’d write this relationship as

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

Once again we can leave off the 1, and the noise term $\epsilon_{ij}$ is implicit.
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

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

In R, we’d write this relationship as

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

Once again we can leave off the 1, and the noise term $\epsilon_{ij}$ is implicit.
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

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \]

\( \sim N(0,\sigma_b) \) \hspace{1cm} \text{Noise}\sim N(0,\sigma_e) \)

- 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

```r
## 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)  #
```
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.
Participant-level clustering is easily visible
Participant-level clustering is easily visible

This reflects the fact that inter-participant variation (125ms) is larger than inter-trial variation (40ms)
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

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \]

\sim \mathcal{N}(0, \sigma_b) \quad \text{Noise} \sim \mathcal{N}(0, \sigma_e)

Thus far, we’ve just defined a model and used it to generate data
Statistical inference with multi-level models

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \sim \text{N}(0, \sigma_b) \text{ Noise} \sim \text{N}(0, \sigma_c) \]

- Thus far, we’ve just defined a model and used it to generate data
- We psycholinguists are usually in the opposite situation...
Statistical inference with multi-level models

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \sim N(0,\sigma_b) \quad \text{Noise} \sim N(0,\sigma_\epsilon) \]

- Thus far, we’ve just defined a model and used it to generate data.
- We psycholinguists are usually in the opposite situation.
- We have data and we need to infer a model.
  - Specifically, the “fixed-effect” parameters \( \alpha, \beta, \) and \( \sigma_\epsilon \), plus the parameter governing inter-subject variation, \( \sigma_b \).
  - e.g., hypothesis tests about effects of neighborhood density: can we reliably infer that \( \beta \) is \{non-zero, positive, \ldots\}?
Statistical inference with multi-level models

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \]

\[ \sim N(0, \sigma_b) \quad \text{Noise} \sim N(0, \sigma_e) \]

▶ Thus far, we've just defined a model and used it to generate data
▶ We psycholinguists are usually in the opposite situation...
▶ We have data and we need to infer a model
  ▶ Specifically, the "fixed-effect" parameters \(\alpha\), \(\beta\), and \(\sigma_e\), plus the parameter governing inter-subject variation, \(\sigma_b\)
  ▶ e.g., hypothesis tests about effects of neighborhood density: can we reliably infer that \(\beta\) is \{non-zero, positive, \ldots\}? 
▶ 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

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \sim N(0,\sigma) \]

\[ \text{Noise} \sim N(0,\sigma_e) \]

\[
> m \leftarrow \text{lmer(time} \sim \text{neighbors.centered} + \\
> \quad (1 \mid \text{participant}), \text{dat,REML=F})
\]

\[
> \text{print(m,corr=F)}
\]

[...]

Random effects:

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name</th>
<th>Variance</th>
<th>Std.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>participant</td>
<td>(Intercept)</td>
<td>4924.9</td>
<td>70.177</td>
</tr>
<tr>
<td>Residual</td>
<td></td>
<td>19240.5</td>
<td>138.710</td>
</tr>
</tbody>
</table>

Number of obs: 1760, groups: participant, 44

Fixed effects:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>583.787</td>
<td>11.082</td>
<td>52.68</td>
<td></td>
</tr>
<tr>
<td>neighbors.centered</td>
<td>8.986</td>
<td>1.278</td>
<td>7.03</td>
<td></td>
</tr>
</tbody>
</table>
### Fitting a multi-level model using maximum likelihood

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \sim N(0,\sigma) \]

\[ \text{Noise} \sim N(0,\sigma) \]

```r
> 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

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \sim N(0,\sigma_b) \quad \text{Noise}\sim N(0,\sigma_\epsilon) \]

\[
> m <- \text{lmer}(time \sim \text{neighbors.centered} + \\
> \quad (1 \mid \text{participant}), \text{dat,REML=F}) \\
> \text{print}(m, \text{corr=F})
\]

[...]

Random effects:

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name (Intercept)</th>
<th>Variance</th>
<th>Std.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>participant</td>
<td></td>
<td>4924.9</td>
<td>70.177</td>
</tr>
<tr>
<td>Residual</td>
<td></td>
<td>19240.5</td>
<td>138.710</td>
</tr>
</tbody>
</table>

Number of obs: 1760, groups: participant, 44

Fixed effects:

<table>
<thead>
<tr>
<th>(\hat{\alpha}) (Intercept)</th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>583.787</td>
<td>11.082</td>
<td>52.68</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\hat{\beta}) neighbors.centered</th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.986</td>
<td>1.278</td>
<td>7.03</td>
<td></td>
</tr>
</tbody>
</table>
Fitting a multi-level model using maximum likelihood

\[
RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij}
\]

\[\sim N(0, \sigma_b)\]

\[\text{Noise} \sim N(0, \sigma_e)\]

\[
> m \leftarrow \text{lmer(time} \sim \text{neighbors.centered} + \\
> \quad (1 \mid \text{participant}), \text{dat, REML=F})
\]

\[
> \text{print}(m, \text{corr=F})
\]

[...]

Random effects:

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name</th>
<th>Variance</th>
<th>Std.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>participant</td>
<td>(Intercept)</td>
<td>4924.9</td>
<td>70.177</td>
</tr>
<tr>
<td></td>
<td>Residual</td>
<td>19240.5</td>
<td>138.710</td>
</tr>
</tbody>
</table>

Number of obs: 1760, groups: participant, 44

Fixed effects:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>583.787</td>
<td>11.082</td>
<td>52.68</td>
</tr>
<tr>
<td>neighbors.centered</td>
<td>8.986</td>
<td>1.278</td>
<td>7.03</td>
</tr>
</tbody>
</table>
Fitting a multi-level model using maximum likelihood

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \sim N(0, \sigma_b) \]

\[ \text{Noise} \sim N(0, \sigma) \]

\[
> m <- \text{lmer(time} \sim \text{neighbors.centered} + \\
> (1 \mid \text{participant}), \text{dat, REML=F})
> \text{print(m, corr=F)}
\]

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

Fixed effects:  
- (Intercept) 583.787 11.082 52.68  
- neighbors.centered 8.986 1.278 7.03
Interpreting parameter estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>583.79</td>
</tr>
<tr>
<td>neighbors.centered</td>
<td>8.99</td>
</tr>
<tr>
<td>$\hat{\sigma}_b$</td>
<td>70.18</td>
</tr>
<tr>
<td>$\hat{\sigma}_\epsilon$</td>
<td>138.7</td>
</tr>
</tbody>
</table>

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

Inter-trial variability $\sigma_\epsilon$ 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 $\sigma_b$ is what's new:
- Variability in average RT in the population from which the participants were drawn has standard deviation 70.18ms.
Interpreting parameter estimates

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>583.79</td>
</tr>
<tr>
<td>neighbors.centered</td>
<td>8.99</td>
</tr>
<tr>
<td>$\hat{\sigma}_b$</td>
<td>70.18</td>
</tr>
<tr>
<td>$\hat{\sigma}_\epsilon$</td>
<td>138.7</td>
</tr>
</tbody>
</table>

The fixed effects are interpreted just as in a traditional single-level model:
Interpreting parameter estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>583.79</td>
</tr>
<tr>
<td>neighbors.centered</td>
<td>8.99</td>
</tr>
<tr>
<td>( \hat{\sigma}_b )</td>
<td>70.18</td>
</tr>
<tr>
<td>( \hat{\sigma}_\epsilon )</td>
<td>138.7</td>
</tr>
</tbody>
</table>

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

- The “average” RT for a non-word in this study is 583.79ms
- Every extra neighbor increases “average” RT by 8.99ms
Interpreting parameter estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>583.79</td>
</tr>
<tr>
<td>neighbors.centered</td>
<td>8.99</td>
</tr>
<tr>
<td>$\hat{\sigma}_b$</td>
<td>70.18</td>
</tr>
<tr>
<td>$\hat{\sigma}_\epsilon$</td>
<td>138.7</td>
</tr>
</tbody>
</table>

- The *fixed effects* are interpreted just as in a traditional single-level model:
  - The “average” RT for a non-word in this study is 583.79ms
  - Every extra neighbor increases “average” RT by 8.99ms
- Inter-trial variability $\sigma_\epsilon$ also has the same interpretation
Interpreting parameter estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>583.79</td>
</tr>
<tr>
<td>neighbors.centered</td>
<td>8.99</td>
</tr>
<tr>
<td>$\hat{\sigma}_b$</td>
<td>70.18</td>
</tr>
<tr>
<td>$\hat{\sigma}_\epsilon$</td>
<td>138.7</td>
</tr>
</tbody>
</table>

- The **fixed effects** are interpreted just as in a traditional single-level model:
  - The “average” RT for a non-word in this study is 583.79ms
  - Every extra neighbor increases “average” RT by 8.99ms
- Inter-trial variability $\sigma_\epsilon$ 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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>583.79</td>
</tr>
<tr>
<td>neighbors.centered</td>
<td>8.99</td>
</tr>
<tr>
<td>$\hat{\sigma}_b$</td>
<td>70.18</td>
</tr>
<tr>
<td>$\hat{\sigma}_\epsilon$</td>
<td>138.7</td>
</tr>
</tbody>
</table>

- The *fixed effects* are interpreted just as in a traditional single-level model:
  - The “average” RT for a non-word in this study is 583.79ms
  - Every extra neighbor increases “average” RT by 8.99ms
- Inter-trial variability $\sigma_\epsilon$ 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 $\sigma_b$ is what’s new:
Interpreting parameter estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>583.79</td>
</tr>
<tr>
<td>neighbors.centered</td>
<td>8.99</td>
</tr>
<tr>
<td>$\hat{\sigma}_b$</td>
<td>70.18</td>
</tr>
<tr>
<td>$\hat{\sigma}_\epsilon$</td>
<td>138.7</td>
</tr>
</tbody>
</table>

- The **fixed effects** are interpreted just as in a traditional single-level model:
  - The “average” RT for a non-word in this study is 583.79ms
  - Every extra neighbor increases “average” RT by 8.99ms
- Inter-trial variability $\sigma_\epsilon$ 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 $\sigma_b$ 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

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \]

\( b_i \sim N(0, \sigma_b) \)  
Noise \( \sim N(0, \sigma_\epsilon) \)

What about the participants’ idiosyncracies themselves—the \( b_i \)?
Inferences about cluster-level parameters

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \]

\( \sim N(0,\sigma_b) \)  
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

\[ RT_{ij} = \alpha + \beta x_{ij} + \hat{b}_i + \hat{\epsilon}_{ij} \]

\[ \sim \mathcal{N}(0, \sigma_b) \quad \text{Noise} \sim \mathcal{N}(0, \sigma_\epsilon) \]

- 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:

\[ P(b_i | \hat{\alpha}, \hat{\beta}, \hat{\sigma}_b, \hat{\sigma}_\epsilon) \]
Inferences about cluster-level parameters

\[ RT_{ij} = \alpha + \beta x_{ij} + b_i + \epsilon_{ij} \]

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:

\[ P(b_i|\hat{\alpha}, \hat{\beta}, \hat{\sigma}_b, \hat{\sigma}_e) \]

The BLUPS are the \textit{conditional modes} of \( b_i \)—the choices that maximize the above probability.
The BLUP participant-specific “average” 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.
The BLUP participant-specific “average” 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.
The BLUP participant-specific “average” 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.
Participants may also have idiosyncratic sensitivities to neighborhood density.
Participants may also have idiosyncratic sensitivities to neighborhood density.

Incorporate by adding cluster-level slopes into the model:

\[
RT_{ij} = \alpha + \beta x_{ij} + b_1 + b_2 x_{ij} + \epsilon_{ij}
\]

\(\sim N(0, \Sigma_b)\)  \(\sim N(0, \sigma_\epsilon)\)
Participants may also have idiosyncratic sensitivities to *neighborhood density*

Incorporate by adding cluster-level slopes into the model:

\[
RT_{ij} = \alpha + \beta x_{ij} + b_{1i} + b_{2i} x_{ij} + \epsilon_{ij}
\]

\[
\sim N(0, \Sigma_b) \quad \text{Noise} \sim N(0, \sigma_\epsilon)
\]

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

\[
RT \sim 1 + x + (1 + x \mid \text{participant})
\]
Inference about cluster-level parameters III

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

\[
RT_{ij} = \alpha + \beta x_{ij} + b_1i + b_2i x_{ij} + \epsilon_{ij}
\]

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

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

[...]

Random effects:

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name</th>
<th>Variance</th>
<th>Std.Dev.</th>
<th>Corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>participant</td>
<td>(Intercept)</td>
<td>4928.625</td>
<td>70.2042</td>
<td></td>
</tr>
<tr>
<td></td>
<td>neighbors.centered</td>
<td>19.421</td>
<td>4.4069</td>
<td>-0.307</td>
</tr>
<tr>
<td>Residual</td>
<td></td>
<td>19107.143</td>
<td>138.2286</td>
<td></td>
</tr>
</tbody>
</table>
Participants may also have idiosyncratic sensitivities to *neighborhood density*

- Incorporate by adding cluster-level slopes into the model:

\[
RT_{ij} = \alpha + \beta x_{ij} + b_1 + b_2 x_{ij} + \epsilon_{ij}
\]

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

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

[...]
```

These three numbers jointly characterize \(\hat{\Sigma}_b\).
Inference about cluster-level parameters IV

Correlation visible in participant-specific BLUPs

\[ \hat{\Sigma} = \begin{pmatrix} 70.20 & -0.3097 & -0.3097 & 4.41 \\ -0.3097 & 70.20 & -0.3097 & -0.3097 \\ -0.3097 & -0.3097 & 70.20 & -0.3097 \\ -0.3097 & -0.3097 & -0.3097 & 70.20 \end{pmatrix} \]
Inference about cluster-level parameters IV

Participants

Correlation visible in participant-specific BLUPs

Participants who were faster overall also tend to be more affected by neighborhood density

\[ \hat{\Sigma} = \left( \begin{array}{ccc} 70 & -0.20 & -0.3097 \\ -0.20 & 4 & 0.41 \end{array} \right) \]
Correlation visible in participant-specific BLUPs

Participants who were faster overall also tend to be more affected by neighborhood density

\[ \hat{\Sigma} = \left( \begin{array}{ccc} 70 & -0.20 & -0.3097 \\ -0.20 & 0.41 & 4 \\ -0.3097 & 0.41 & 0.3097 \end{array} \right) \]
Inference about cluster-level parameters IV

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

\[
\hat{\Sigma} = \begin{pmatrix}
70.20 & -0.3097 \\
-0.3097 & 4.41
\end{pmatrix}
\]
Bayesian inference for multilevel models

\[ P(\{\beta_i\}, \sigma_b, \sigma_\epsilon | Y) = \frac{P(Y|\{\beta_i\}, \sigma_b, \sigma_\epsilon) P(\{\beta_i\}, \sigma_b, \sigma_\epsilon)}{P(Y)} \]

- 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

\[ P(\{\beta_i\}, \sigma_b, \sigma_c | Y) = \frac{P(Y|\{\beta_i\}, \sigma_b, \sigma_c) P(\{\beta_i\}, \sigma_b, \sigma_c)}{P(Y)} \]

- 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

\[
P({\beta_i}, \sigma_b, \sigma_\epsilon | Y) = \frac{P(Y|{\beta_i}, \sigma_b, \sigma_\epsilon) P({\beta_i}, \sigma_b, \sigma_\epsilon)}{P(Y)}
\]

▶ 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
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 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 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
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

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

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 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
  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 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

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 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

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 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
  
  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)
The logit link function for categorical data

- Much psycholinguistic data is *categorical* rather than *continuous*:
  - Yes/no answers to alternations questions
  - Speaker choice: (*realized* *(that) her goals were unattainable*)
  - Cloze continuations, and so forth...
The logit link function for categorical data

- Much psycholinguistic data is *categorical* rather than *continuous*:
  - Yes/no answers to alternations questions
  - Speaker choice: *(realized *(that)* her goals were unattainable)*
  - Cloze continuations, and so forth…
- Linear models inappropriate; they predict continuous values
The logit link function for categorical data

- Much psycholinguistic data is *categorical* rather than *continuous*:
  - Yes/no answers to alternations questions
  - Speaker choice: \((\text{realized (that) her goals were unattainable})\)
  - Cloze continuations, and so forth...

- Linear models inappropriate; they predict continuous values

- We can stay within the multi-level generalized linear models framework but use different *link functions* and *noise distributions* to analyze categorical data
The logit link function for categorical data

- Much psycholinguistic data is *categorical* rather than *continuous*:
  - Yes/no answers to alternations questions
  - Speaker choice: *(realized *(that) her goals were unattainable)*
  - Cloze continuations, and so forth...

- Linear models inappropriate; they predict continuous values

- We can stay within the multi-level generalized linear models framework but use different link functions and noise distributions to analyze categorical data

- e.g., the logit model *(Agresti, 2002; Jaeger, 2008)*
The logit link function for categorical data

- Much psycholinguistic data is categorical rather than continuous:
  - Yes/no answers to alternations questions
  - Speaker choice: *(realized *(that)* her goals were unattainable)*
  - Cloze continuations, and so forth...

- Linear models inappropriate; they predict continuous values

- We can stay within the multi-level generalized linear models framework but use different link functions and noise distributions to analyze categorical data

- e.g., the logit model *(Agresti, 2002; Jaeger, 2008)*

\[ \eta_{ij} = \alpha + \beta X_{ij} + b_i \]  
*(linear predictor)*
The logit link function for categorical data

- Much psycholinguistic data is *categorical* rather than *continuous*:
  - Yes/no answers to alternations questions
  - Speaker choice: *(realized (that) her goals were unattainable)*
  - Cloze continuations, and so forth . . .

- Linear models inappropriate; they predict continuous values
- We can stay within the multi-level generalized linear models framework but use different **link functions** and **noise distributions** to analyze categorical data
- e.g., the logit model (Agresti, 2002; Jaeger, 2008)

\[
\eta_{ij} = \alpha + \beta X_{ij} + b_i \quad \text{(linear predictor)}
\]

\[
\eta_{ij} = \log \frac{\mu_{ij}}{1 - \mu_{ij}} \quad \text{(link function)}
\]
The logit link function for categorical data

- Much psycholinguistic data is *categorical* rather than *continuous*:
  - Yes/no answers to alternations questions
  - Speaker choice: *(realized* *(that)* *her goals were unattainable)*
  - Cloze continuations, and so forth...
- Linear models inappropriate; they predict continuous values
- We can stay within the multi-level generalized linear models framework but use different link functions and noise distributions to analyze categorical data
- e.g., the logit model (Agresti, 2002; Jaeger, 2008)

\[
\begin{align*}
\eta_{ij} &= \alpha + \beta X_{ij} + b_i \quad \text{(linear predictor)} \\
\eta_{ij} &= \log \frac{\mu_{ij}}{1 - \mu_{ij}} \quad \text{(link function)} \\
\Pr(Y = y; \mu_{ij}) &= \mu_{ij} \quad \text{(binomial noise distribution)}
\end{align*}
\]
We’ll look at the effect of neighborhood density on correct identification as non-word in Bicknell et al. (2008)
We’ll look at the effect of neighborhood density on correct identification as non-word in Bicknell et al. (2008)

Assuming that any effect is linear in log-odds space (see Jaeger (2008) for discussion)
We’ll look at the effect of neighborhood density on correct identification as non-word in Bicknell et al. (2008)

Assuming that any effect is linear in log-odds space (see Jaeger (2008) for discussion)

```r
> lmer(correct ~ neighbors.centered + (1 | participant),
       dat, family="binomial")
```

Random effects:

```
Groups Name Variance Std.Dev.
participant (Intercept) 0.9243 0.9614
```

Number of obs: 1760, groups: participant, 44

Fixed effects:

```
Estimate Std. Error  z value Pr(>|z|)  
(Intercept) 3.16310 0.18998 16.649 < 2e-16 ***  
neighbors.centered -0.18207 0.03483 -5.228 1.72e-07 ***
```
We’ll look at the effect of neighborhood density on correct identification as non-word in Bicknell et al. (2008)

Assuming that any effect is linear in log-odds space (see Jaeger (2008) for discussion)

```r
> lmer(correct ~ neighbors.centered + (1 | participant),
     dat, family="binomial")
```

Random effects:

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name</th>
<th>Variance</th>
<th>Std.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>participant</td>
<td>(Intercept)</td>
<td>0.9243</td>
<td>0.9614</td>
</tr>
</tbody>
</table>

Number of obs: 1760, groups: participant, 44

Fixed effects:

\[ \hat{\alpha} \]—participants usually right

| Estimate     | Std. Error | z value | Pr(>|z|) |
|--------------|------------|---------|----------|
| (Intercept)  | 3.16310    | 0.18998 | 16.649   | < 2e-16 *** |
| neighbors.centered | -0.18207 | 0.03483 | -5.228   | 1.72e-07 *** |
We’ll look at the effect of neighborhood density on correct identification as non-word in Bicknell et al. (2008).

Assuming that any effect is linear in log-odds space (see Jaeger (2008) for discussion)

```r
> lmer(correct ~ neighbors.centered + (1 | participant), dat, family="binomial")

[...]
Random effects:
  Groups Name Variance Std.Dev. 
    participant (Intercept) 0.9243 0.9614

Number of obs: 1760, groups: participant, 44

Fixed effects:
  Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.16310   0.18998 16.649  < 2e-16 ***
neighbors.centered -0.18207   0.03483 -5.228   1.72e-07 ***
```

\[\hat{\alpha}\]—participants usually right
\[\hat{\sigma}_b\] (note there is no \[\hat{\sigma}_\epsilon\] for logit models)
We’ll look at the effect of neighborhood density on correct identification as non-word in Bicknell et al. (2008).

Assuming that any effect is linear in log-odds space (see Jaeger (2008) for discussion)

```r
> lmer(correct ~ neighbors.centered + (1 | participant),
    dat, family="binomial")
```

Random effects:

```
Groups Name       Variance Std.Dev.  
participant (Intercept) 0.9243 0.9614
```

Number of obs: 1760, groups: participant, 44

Fixed effects:

```
(Intercept) 3.16310 0.18998 16.649 < 2e-16 ***
neighbors.centered -0.18207 0.03483 -5.228 1.72e-07 ***
```

\[ \hat{\alpha} \]—participants usually right

\[ \hat{\beta} \]—effect small compared with inter-subject variation

\[ \hat{\sigma}_b \] (note there is no \[ \hat{\sigma}_\epsilon \] for logit models)
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 (and the conference...?), these models open up many new interesting doors!

