Basics of Multilevel Models

- Generalize linear regression, allowing intercepts, or slopes, or ... to vary by group ID.
- In a regression, include a categorical predictor which may interact with other predictors.

Complete pooling ignores groups,
No pooling fits a model for each group separately.
We want to explore middle ground between these 2 extremes.

Notation

- Units numbered $i = 1, \ldots, n$. (Not necessarily experimental or observational units, as those may be measured repeatedly.)
- Response is y_i
- Data matrix $\mathbf{X}$ of dimension n by k. The ith row is called $\mathbf{X}_i$, kth column is $\mathbf{X}_{(k-1)}$, as the first column is $\mathbf{X}_{(0)} = \mathbf{1}$.
- Coefficient vector $\boldsymbol{\beta}^{\mathsf{T}} = (\beta_0 \ \cdots \ \beta_k)$ such that $E(\mathbf{y}) = \mathbf{X}\boldsymbol{\beta}$.

Multilevel Extensions

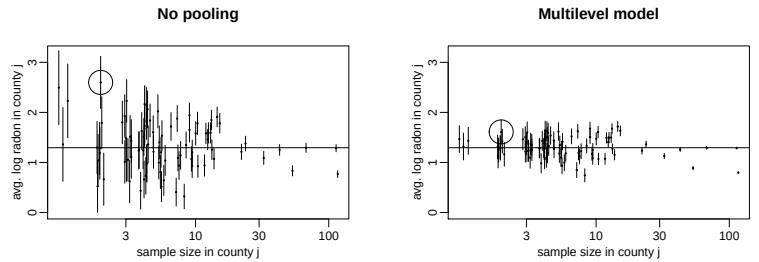
- Groups $j = 1, \ldots, J$, e.g. centers at which our clinical trials are run.
- Possible second level of grouping $k = 1, \ldots, K$ (could be confused with predictor label).
- Indexing $j[i]$ tells us which group the ith obs. belongs to.

More Notation

- Coefficient vector can be separated as in $\boldsymbol{\beta}^{\mathsf{T}} = (\alpha \ \beta)$
  Group level coefficients may be called $\boldsymbol{\gamma}$.
  In R and BUGs code, $\alpha \to$ a, $\beta \to$ b, $\gamma \to$ g
- With multiple predictors, $y_i = \mathbf{X}_i\mathbf{B} + \epsilon_i$ where $\mathbf{B}$ is a matrix of coefficients.
- In addition to $\sigma^2 = \text{Var}(\epsilon_i)$, we have other variances σ^2_α, σ^2_β.
- Some predictors are measured only at the group level.
  $\alpha_j \sim N(U_j\boldsymbol{\gamma}, \sigma^2_\alpha)$ where $\mathbf{U}$ has J rows.

Partial Pooling

Radon seeps from soil into people's homes.
A study of radon level was done in 85 Minnesota counties.
Consider the 85 county means (sample sizes vary from 1 to 116).



No pooling means we estimate each mean independently.
Partial pooling uses a weighted average of the overall mean (1.3) and the county mean. As n increases, the overall mean gets less weight.

Multilevel Pooling

For counties with few measurements, it really helps to "borrow" information from other counties. Pooling pulls in extremes and reduces variance.

$$\widehat{a}_j \approx \frac{\frac{n_j}{\sigma_y^2}\overline{y}_{\cdot j} + \frac{1}{\sigma_\alpha^2}\overline{y}_{\cdot\cdot}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}} \text{ with variance: } \frac{1}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}}$$

The weights applied to the two means are inverse variances (called precisions). $\text{Var}(\overline{y}_{\cdot j}) = \frac{\sigma_y^2}{n_j}$ and $\text{Var}(\overline{y}_{\cdot\cdot}) = \sigma_\alpha^2$

Complete Pooling With Predictors

A predictor of interest is "floor" where 0 means basement, 1 = ground floor. We expect higher radon readings in basements. First use complete pooling:

```
lm.pooled <- lm(log.radon ~ floor)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.327	0.030	44.640	0.000
floor	-0.613	0.073	-8.421	0.000

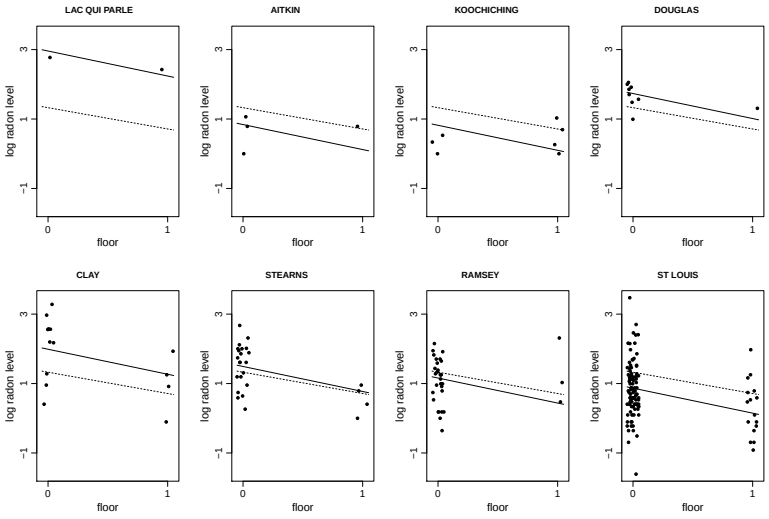
```
lm.unpooled <- lm(log.radon ~ floor + factor(county) - 1)
```

	Estimate	Std. Error	t value	Pr(>|t|)
floor	-0.721	0.074	-9.800	0.000
factor(county)1	0.841	0.379	2.220	0.027
factor(county)2	0.875	0.105	8.333	0.000
factor(county)85	1.187	0.535	2.218	0.027

Note that slope changes when we add county-level intercepts.

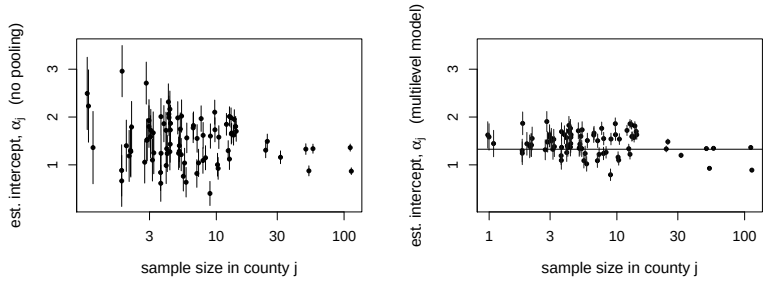
Complete versus No Pooling

Solid line: No pooling, dashed: complete pooling



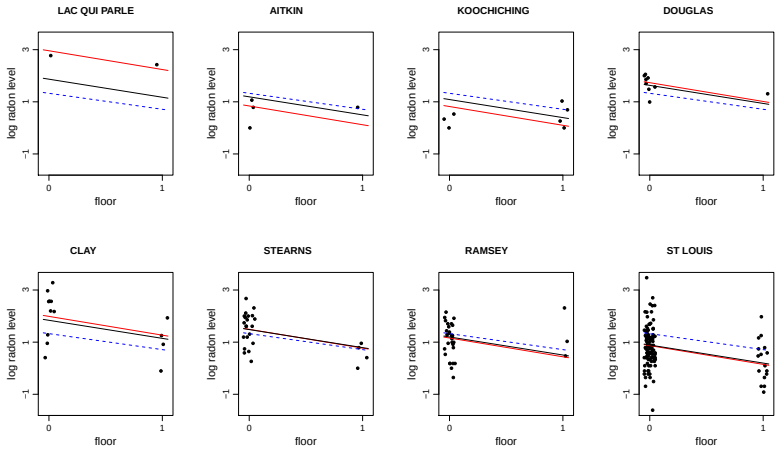
Intercept comparison

Slope is all the same, but intercepts are allowed to vary by county.



Similar to figure 12.1, but we did add one slope.

Pooled Effects



Red = unpooled, blue = completely pooled, black = partially pooled. Slopes are still all the same.

Parameter Estimates

Individual county intercepts are not of interest. Variance is.

Parameter	Estimate
σ_α	0.33
μ_α	1.46
β	-0.69
σ_y	0.76

Total variance, $\text{Var}(y_i) = \text{Var}(\alpha_{j[i]}) + \text{Var}(\epsilon_i) = \sigma_\alpha^2 + \sigma_y^2$ estimated as $0.33^2 + 0.76^2 = 0.687$. Of that, what proportion is due to county-to-county variation? $0.33^2/0.687 = 0.16$

Two houses from different counties have covariance: _____

Two houses from the same county share $\alpha_{j[i]}$, and have covariance _____.

Intraclass correlation between two houses in the same county is: _____

Variance Interpretation

$0.33^2/0.76^2 = 0.19$ so counties are about one-fifth as variable as individuals.

When we average individuals together into a sample mean, we also reduce variation. County-to-county variation is comparable to the variance of a mean of 5 observations.

If a county has less than 5 obs, then it's sample mean is more variable than that of α_j, so it's pooled intercept should be closer to the complete-pooled intercept than to its individual unpooled estimate. At n=5, they should get averaged together. With larger sample sizes, partial pooled estimate should be closer to the unpooled values.

"Shrinkage" describes the movement of estimators toward a common value.

§12.4 Fitting Multilevel Models in R

- `lme` in package `nlme`

```
require(nlme)
lme.fit <- lme(log.radon ~ floor, data = minn, random = ~1 |
    county)
```

  Creates S3 objects (subset with $). Has intervals, update functions. Handles nested hierarchical models.

- `lmer` in package `lme4`

```
require(lme4)
lmer.fit <- lmer(log.radon ~ floor + (1 | county), data = minn)
```

  Creates S4 objects (subset with `@`). No intervals, predict, or update. Does have `refit` to use the same model with a different response. Preferred for crossed levels.

More R fits

`nlme` written by Douglas Bates (U Minn) and Josè Piñheiro. Includes `gls` which allows us to specify non-constant variance and correlation structures. Feels more finished.

`lmer` by Doug Bates is a work in progress. Expect it to change. Will fit glm's like binomial & Poisson regression. Bates intentionally does not make it really easy to get predictions, etc., because he wants people to use it if they know what they are doing.
SAS Alternative:

```
PROC mixed method=reml nobound;
    class county;
    model logRadon = floor /ddfm=satterth s;
    random int/subject=county;
run;
```

Different optimization choices, don't expect agreement.
SAS specifies correlations and variance structures differently.

Extracting from lmer fit

```
coef(lmer.fit)$county[c(1:3, NA, 85), ]

##                      (Intercept)  floor
## AITKIN                     1.192 -0.693
## ANOKA                      0.928 -0.693
## BECKER                     1.479 -0.693
## NA                            NA     NA
## YELLOW MEDICINE            1.386 -0.693

fixef(lmer.fit)

## (Intercept)       floor
##       1.462      -0.693

ranef(lmer.fit)$county[c(1:3, 85), 1]

## [1] -0.2701 -0.5339  0.0176 -0.0754
```


Extracting from lmer fit 2

```
se.fixef(lmer.fit)  ## within arm package

## (Intercept)       floor
##      0.0516      0.0704

se.ranef(lmer.fit)$county[c(1:3, 85), 1]

## AITKIN                ANOKA
##                0.1870                0.0304
## BECKER                YELLOW MEDICINE
##                0.2096                0.2383
```

95% CI for slope (there's only one)

```
fixef(lmer.fit)["floor"] + c(-2, 2) * se.fixef(lmer.fit)["floor"]

## [1] -0.834 -0.552
```


Extracting from lmer fit 3

County intercepts for Mahnomen ($n_j = 1$) and St Louis ($n_j = 116$) counties

```
coef(lmer.fit)$county[42, 1] + c(-2, 2) * se.ranef(lmer.fit)$county[42]

## [1] 0.893 1.998

coef(lmer.fit)$county[73, 1] + c(-2, 2) * se.ranef(lmer.fit)$county[73]

## [1] 0.861 0.919
```

As plotted in figure 12.4 these include the overall mean. For deviations from mean:

```
ranef(lmer.fit)$county[73, 1] + c(-2, 2) * se.ranef(lmer.fit)$county[73

## [1] -0.600 -0.543
```

§12.5 Ways to Write Nested Models 1–2

Use radon example with $x_1 =$ floor, and $x_2 = \log$(soil uranium) concentration in soils (one per county). Assume just intercept varies with county.

- Coefficients vary across groups.
  $y_i = \alpha_{j[i]} + \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i,\ \alpha_j \sim (\mu_\alpha, \sigma^2_\alpha)$
  In matrix notation:
  $y_i = \alpha_{j[i]} + \mathbf{X}_i\boldsymbol{\beta} + \epsilon_i,\ \alpha_j \sim (\mu_\alpha, \sigma^2_\alpha)$
  Group-level errors
  $y_i = \mu_\alpha + \eta_{j[i]} + \mathbf{X}_i\boldsymbol{\beta} + \epsilon_i,\ \eta_j \sim (0, \sigma^2_\alpha)$
- Separate local regressions
  Within County: $y_i \sim N(\alpha_j + \beta x_i, \sigma^2_y)$
  Across counties: $\alpha_j \sim N(\gamma_0 + \gamma_1 u_j, \sigma^2_\alpha)$
  or: $\alpha_j = \gamma_0 + \gamma_1 u_j + \eta_j,\ \eta_j \sim N(0, \sigma^2_\alpha)$

Ways to Write Nested Models 3–5

- Large Regression $y_i \sim N(\mathbf{X}_i\boldsymbol{\beta}, \sigma^2_y)$ $\mathbf{X}_i = [1\ x_i\ u_i\ I_{i2}\ I_{i3} \cdots\ I_{iJ}]$
  (floor, uranium, Indicators for counties 2 to J)
  $\beta_j \sim N(0, \sigma^2_\alpha);\ j = 3, \ldots, J+2$
  Or takeout the column of ones (and β_0) and let
  $\beta_j \sim N(\mu_\alpha, \sigma^2_\alpha);\ j = 3, \ldots, J+2$
- Multiple Errors $y_i \sim N(\mathbf{X}_i\boldsymbol{\beta} + \eta_{j[i]}, \sigma^2_y),;\ i = 1, \ldots, n,$
  $\eta_j \sim N(0, \sigma^2_\alpha)$ $\mathbf{X}_i = [1\ x_i\ u_i]$
- Large Regression with Correlations – a vector for each county.

$$\mathbf{y}_j = \mathbf{X}_j\boldsymbol{\beta} + \boldsymbol{\epsilon}_j;\ \boldsymbol{\epsilon}_j \sim \text{iid}N(\mathbf{0}, \boldsymbol{\Sigma}),\ j = 1, \ldots, J$$

$$\boldsymbol{\Sigma}_{ii} = \sigma^2_y + \sigma^2_\alpha, \quad \boldsymbol{\Sigma}_{ik} = \begin{cases} \sigma^2_\alpha & \text{if } j[i] = j[k], i \neq k \\ 0 & \text{otherwise} \end{cases}$$

§12.6 Group-level Predictors

Uranium is a one-per-county measurement.
Similar to measure funding on each school district and test students within the district.

```
summary(lmer.fit2 <- lmer(log.radon ~ floor + u + (1 | county),
    minn))

## Linear mixed model fit by REML ['lmerMod']
## Formula: log.radon ~ floor + u + (1 | county)
##    Data: minn
##
## REML criterion at convergence: 2134
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  county   (Intercept) 0.0245   0.156
##  Residual             0.5752   0.758
## Number of obs: 919, groups: county, 85
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   1.4658     0.0379    38.6
## floor        -0.6682     0.0688    -9.7
## u             0.7203     0.0918     7.8
```


Group-level Predictors

```
coef(lmer.fit2)$county[c(1:3, NA, 85), ]

##                      (Intercept)  floor    u
## AITKIN                      1.45 -0.668 0.72
## ANOKA                       1.48 -0.668 0.72
## BECKER                      1.48 -0.668 0.72
## NA                            NA     NA   NA
## YELLOW MEDICINE             1.42 -0.668 0.72

fixef(lmer.fit2)

## (Intercept)       floor           u
##       1.466      -0.668       0.720

ranef(lmer.fit2)$county[c(1:3, 85), 1]

## [1] -0.0206  0.0112  0.0124 -0.0419
```

Interpretation

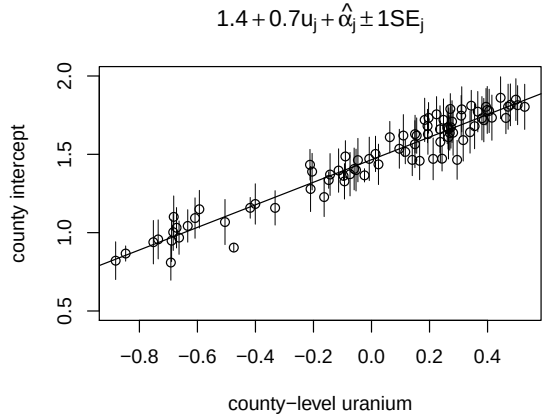
Population Average model from the fixed effects:
$\widehat{y}_i = 1.47 - 0.67x_i + 0.72u_i$ Add county "error" for prediction on 1 county, say Yellow Medicine where log radon is 0.355.

- From coef output:
  $\hat{y}_i = 1.42 - 0.668x_i + 0.72(0.355) = 1.68 - 0.67x_i$
- From fixef and ranef output:
  $\hat{y}_i = 1.47 - 0.668x_i + 0.72(0.355) - 0.042$

For all counties:

```
u <- with(minn, tapply(u, county, median))
a.hat <- cbind(1, 0, u) %*% fixef(lmer.fit2) + unlist(ranef(lmer.fit2)$county)
a.SEs <- unlist(se.coef(lmer.fit2)$county)
plot(u, t(a.hat), ylim = c(0.5, 2), xlab = "county-level uranium",
    ylab = "county intercept")
segments(u, a.hat - a.SEs, u, a.hat + a.SEs, lwd = 0.5)
title(expression(1.4 + 0.7 * u[j] + hat(alpha[j]) %+-% 1 * SE[j]))
abline(fixef(lmer.fit2)[c(1, 3)], lwd = 1, col = "black")
```


Figure 12.6



The line is the "population average" model. All points use the same slope for log(uranium), so vertical shifts are due to "random" intercept for each county.

Shrinkage

Adding uranium explains part of the county-to-county variation, none of the within county variation.

$$\alpha_j \approx \frac{n_j/\sigma_y^2}{n_j/\sigma_y^2+1/\sigma_\alpha^2} \cdot (\text{group } J \text{ est}) + \frac{1/\sigma_\alpha^2}{n_j/\sigma_y^2+1/\sigma_\alpha^2} \cdot (\text{regression est})$$

or

$$\eta_j \approx \frac{n_j/\sigma_y^2}{n_j/\sigma_y^2+1/\sigma_\alpha^2}(\overline{y}_j - \overline{X}_j\beta - U_j\gamma) + \frac{1/\sigma_\alpha^2}{n_j/\sigma_y^2+1/\sigma_\alpha^2} \cdot 0$$

§12.7 Model Building and Statistical Significance

Multilevel modeling is most necessary when at least some groups need to borrow information from the others. When σ_α is small and groups are similar.

If σ_α is large, groups vary a lot and multilevel is like no-pooling.

We don't want to force all groups to be alike. Partial pooling works well with unequal sample sizes (and precisions)

Where to begin modeling?
Start Simple and Add Complexity

- Complete Pooling – one model ignoring group effects. Can include group-level predictors.
- No pooling – a model with group effects, but no group-level predictors.
- Separate models – one for each level $j = 1, \ldots, J$ using no group level predictors (may be limited by sample sizes, n_j).
- Two-stage analysis using no-pooling or separate models to get group level estimates. Then treat these as data in a group-level model.

Group-level predictors reduce the variance between groups, σ_α^2 giving more precise estimates of α_j.

Stat Significance is Overrated

Stat literature on random effects terms $\widehat{\alpha}_j$ a prediction, not an estimate. Do not ask if we need α_j in the model.
Radon example (ignoring floor and uranium)

County	Estimate	SE(estimate)	Est/SE
AITKIN	1.31 -0.2451	0.245	-1
ANOKA	1.31 -0.425	0.104	-4
BECKER	1.31 -0.082	0.257	-1/3

Anoka county effect is not "highly significant" (even relative to Aitken). We would not drop one and keep another. We could remove all of them if we thought county variance was ignorable.
Don't use small individual p-values as a reason to exclude variables.
Biggest constraints: fitting and understanding complex models.

§12.8 Predictions

Reminders from classical regression:

- Use the `predict` function in R with `newdata=list(x=x.new), se.fit=T` to get estimated $\widehat{y}_{\text{new}}$ and its SE. To predict for a new observation $SE_{\text{prediction}} = \sqrt{s^2 + SE(\widehat{y}_{\text{new}})^2}$
- Use the `sim` function in `arm` package to generate random $\tilde{s}^2 \sim$ Scaled-Inv-χ^2 and $\tilde{\beta} \sim N(\widehat{\beta}, \tilde{s}^2(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1})$. Then new predicted response is $\tilde{y}_{\text{new}} \sim N(\mathbf{X}_{\text{new}}\tilde{\beta}, \tilde{s}^2)$
- Logistic regression: generate random $\mathbf{X}_{\text{new}}\tilde{\beta}$ as above, convert to probability with $\tilde{p}_i = \text{logit}^{-1}(\mathbf{X}_{\text{new}}\tilde{\beta})$ and generate random Bin$(n_i, \tilde{p}_i)$.
- Poisson: $\tilde{y}_{\text{new}} \sim Poisson(u_{\text{new}} e^{\mathbf{X}_{\text{new}}\tilde{\beta}})$ where u is exposure.

Elections example

R functions do not work with derived quantities, but sim does.

```
elect90.fit <- lm(DemRatio ~ DemRatio88 * party + incumbent,
    elect2)
sim.1 <- sim(elect90.fit, 1000)
sigma.sim <- sim.1@sigma
X.tilde <- model.matrix(elect90.fit)
n.tilde <- nrow(X.tilde)
y.tilde <- X.tilde %*% t(sim.1@coef) + rnorm(n.tilde * 1000,
    0, rep(sigma.sim, each = n.tilde))
quantile(y.tilde[3, ], c(0.025, 0.25, 0.5, 0.75, 0.975))

##  2.5%   25%   50%   75% 97.5%
## 0.496 0.582 0.632 0.678 0.769
```

Multilevel Predictions 1 –Existing Group

Predict log radon level for the first floor of a new house in Hennepin county. (county 26)
Assume all coefficients are estimated without error.

```
sigma.y.hat <- sigma.hat(lmer.fit2)$sigma$data
beta.hat <- as.matrix(coef(lmer.fit2)$county)[26, ]
y.tilde.floor1 <- rnorm(1000, beta.hat %*% c(1, 1, 0.908), sigma.y.hat)
exp(quantile(y.tilde.floor1, c(0.025, 0.5, 0.975)))

##   2.5%    50%  97.5%
##  0.933  4.201 17.896

## Simulate radon in 'average' Hennepin house
y.tilde.floor0 <- rnorm(1000, beta.hat %*% c(1, 0, 0.908), sigma.y.hat)
summary(0.9 * exp(y.tilde.floor0) + 0.1 * exp(y.tilde.floor1))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.9     4.9     7.9    10.0    12.4    84.3
```

Really should simulate draws of β and α as well, so this answer does not contain all the variance effects

Multilevel Predictions 2 – A New Group

Now we don't know α_j, must draw it from $N(\gamma_0 + \gamma_1 u_{\text{new}}, \sigma_\alpha^2)$

```
sigma.a.hat <- sigma.hat(lmer.fit2)$sigma$county
gamma.hat <- fixef(lmer.fit2)[-2]  ##omit floor effect
a.tilde <- rnorm(1000, gamma.hat[1] + gamma.hat[2] * 0.0226,
    sigma.a.hat)
y.tilde <- rnorm(1000, a.tilde + beta.hat[2] * 1, sigma.y.hat)
## U already accounted for in a.tilde
quantile(y.tilde, c(0.025, 0.25, 0.5, 0.75, 0.975))

##   2.5%    25%    50%    75%  97.5%
## -0.711  0.251  0.821  1.349  2.267

exp(quantile(y.tilde, c(0.025, 0.5, 0.975)))  ## back to pC/l

##  2.5%   50% 97.5%
## 0.491 2.273 9.652
```


Sample Sizes

With a small number of groups, J, it's hard to estimate σ_α. Tends to get over-estimated, making multilevel like no-pooling.

Need at least two observations in some groups to estimate σ_y.

This does not answer the sample size question: How big a sample do I need to take to detect an effect of size E with power of .80 (1 - chance of a type II error) at $\alpha = .05$ when $\sigma = 3$? With fixed effects models we can answer it using noncentral t or F distributions. In multilevel models we need to simulate.