THE UNIVERSITY of EDINBURGH

Multilevel Models

Data Analysis for Psychology in R 3

Josiah King, Umberto Noè, Tom Booth

Department of Psychology
The University of Edinburgh

AY 2021-2022

Part 1: LM to MLM

Part 2: Inference in MLM

Terminology

(size weighted by hits on google scholar)

Notation

Simple regression

for observation i

$$y_i = \beta_0 \cdot 1 \ + \ \beta_1 \cdot x_i + \varepsilon_i$$

Notation

Multi-level

for observation j in group i

Level 1:
$y_{ij} = \beta_{0i} \cdot \mathbf{1} + \beta_{1i} \cdot x_{ij} + \varepsilon_{ij}$
Level 2:
$\beta_{0i} = \gamma_{00} + \zeta_{0i}$
$\beta_{1i} = \gamma_{10} + \zeta_{1i}$

Notation

Multi-level

for observation j in group i

Level 1:
$y_{ij} = \beta_{0i} \cdot 1 + \beta_{1i} \cdot x_{ij} + \varepsilon_{ij}$
Level 2:
$\beta_{0i} = \gamma_{00} + \zeta_{0i}$
$\beta_{1i} = \gamma_{10} + \zeta_{1i}$

Where:
γ_{00} is the population intercept
and ζ_{0i} is the deviation of group i from γ_{00}

γ_{10} is the population slope,
and ζ_{1i} is the deviation of group i from γ_{10}

Notation

Multi-level

for observation j in group i

Level 1:
$y_{ij} = \beta_{0i} \cdot 1 + \beta_{1i} \cdot x_{ij} + \varepsilon_{ij}$
Level 2:
$\beta_{0i} = \gamma_{00} + \zeta_{0i}$
$\beta_{1i} = \gamma_{10} + \zeta_{1i}$

Where:
γ_{00} is the population intercept
and ζ_{0i} is the deviation of group i from γ_{00}

γ_{10} is the population slope,
and ζ_{1i} is the deviation of group i from γ_{10}

We are now assuming ζ_0, ζ_1, and ε to be normally distributed with a mean of 0, and we denote their variances as $\sigma^2_{\zeta_0}$, $\sigma^2_{\zeta_1}$, σ^2_{ε} respectively.

The ζ components also get termed the "random effects" part of the model, Hence names like "random effects model", etc.

Notation

Mixed-effects == Multi Level

Sometimes, you will see the levels collapsed into one equation, as it might make for more intuitive reading:

$$y_{ij} = \underbrace{(\gamma_{00} + \zeta_{0i})}_{\beta_{0i}} \cdot 1 + \underbrace{(\gamma_{10} + \zeta_{1i})}_{\beta_{1i}} \cdot x_{ij} + \varepsilon_{ij}$$

other notation to be aware of

- Many people use the symbol u in place of ζ
- Sometimes people use β_{00} instead of γ_{00}
- In various resources, you are likely to see α used to denote the intercept instead of β_0

Notation

Matrix form

And then we also have the condensed matrix form of the model, in which the Z matrix represents the grouping structure of the data, and ζ contains the estimated random deviations.

$$
\begin{bmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \\ y_{31} \\ y_{32} \end{bmatrix}
=
\begin{bmatrix} 1 & x_{11} \\ 1 & x_{12} \\ 1 & x_{21} \\ 1 & x_{22} \\ 1 & x_{31} \\ 1 & x_{32} \end{bmatrix}
\begin{bmatrix} \gamma_{00} \\ \beta_1 \end{bmatrix}
+
\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} \zeta_{01} \\ \zeta_{02} \\ \zeta_{03} \end{bmatrix}
+
\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{12} \\ \varepsilon_{21} \\ \varepsilon_{22} \\ \varepsilon_{31} \\ \varepsilon_{32} \end{bmatrix}
$$

$$
\boldsymbol{y} = \mathbf{X} \boldsymbol{\beta} + \mathbf{Z} \boldsymbol{\zeta} + \varepsilon
$$

"Fixed" vs "Random"

Level 1:

$y_{ij} = \beta_{0i} \cdot 1 + \beta_{1i} \cdot x_{ij} + \varepsilon_{ij}$

Level 2:

$\beta_{0i} = \underbrace{\gamma_{00}}_{\text{fixed}} + \underbrace{\zeta_{0i}}_{\text{random}}$

$\beta_{1i} = \underbrace{\gamma_{10}}_{\text{fixed}} + \underbrace{\zeta_{1i}}_{\text{random}}$

$$y_{ij} = (\underbrace{\gamma_{00}}_{\text{fixed}} + \underbrace{\zeta_{0i}}_{\text{random}}) \cdot 1 + (\underbrace{\gamma_{10}}_{\text{fixed}} + \underbrace{\zeta_{1i}}_{\text{random}}) \cdot x_{ij} + \varepsilon_{ij}$$

ζ_i is "random" because considered a random sample from larger population such that $\zeta_i \sim N(0, \sigma^2_{\zeta_i})$.

Fixed vs Random

What is the difference?

When specifying a random effects model, think about the data you have and how they fit in the following table:

Criterion:	**Repetition:** ***If the experiment were repeated:***	**Desired inference:** ***The conclusions refer to:***
Fixed effects	Same levels would be used	The levels used
Random effects	Different levels would be used	A population from which the levels used are just a (random) sample

Fixed vs Random

What is the difference?

When specifying a random effects model, think about the data you have and how they fit in the following table:

Criterion:	**Repetition:** *If the experiment were repeated:*	**Desired inference:** *The conclusions refer to:*
Fixed effects	Same levels would be used	The levels used
Random effects	Different levels would be used	A population from which the levels used are just a (random) sample

- Sometimes, there isn't much variability in a specific random effect and to allow your model to fit it is common to just model that variable as a fixed effect.
- Other times, you don't have sufficient data or levels to estimate the random effect variance, and you are forced to model it as a fixed effect.

Advantages of MLM

Multi-level models can be used to answer multi-level questions!

Do phenomena at Level X predict **outcomes** at Level Y?

Does population density in school district predict variation in scores in childrens' first year of school?

$$\text{score}_{ij} = \beta_{0i} + \beta_1 \text{school\_year}_j + \varepsilon_{ij}$$

$$\beta_{0i} = \gamma_{00} + \gamma_{01} \text{district\_pop\_dens}_i + \zeta_{0i}$$

Advantages of MLM

Multi-level models can be used to answer multi-level questions!

Do phenomena at Level X influence **effects** at Level Y?

Does amount of school funding influence childrens' improvement in scores over time?

$$\text{score}_{ij} = \beta_0 + \beta_{1i}\text{school\_year}_j + \varepsilon_{ij}$$

$$\beta_{1i} = \gamma_{10} + \gamma_{11}\text{school\_funding}_i + \zeta_{1i}$$

Advantages of MLM

Multi-level models can be used to answer multi-level questions!

Do random variances covary?

Do children who score higher at the start of school show greater improvements than those who start lower?

$$\text{score}_{ij} = \beta_{0i} + \beta_{1i}\text{school\_year}_j + \varepsilon_{ij}$$

$$\begin{aligned}\beta_{0i} &= \gamma_{00} + \zeta_{0i} \\ \beta_{1i} &= \gamma_{10} + \zeta_{1i}\end{aligned}$$

$$\begin{bmatrix} \sigma^2_{\zeta_0} & \\ \sigma_{\zeta_0,\zeta_1} & \sigma^2_{\zeta_1} \end{bmatrix}$$

lme4

- **lme4** package (many others are available, but **lme4** is most popular).
- `lmer()` function.
- syntax is similar to `lm()`, in that we specify:

  [outcome variable] ~ *[explanatory variables]*, data = *[name of dataframe]*

- in `lmer()`, we add to this the random effect structure in parentheses:

  [outcome variable] ~ *[explanatory variables]* + (*[vary this]* | *[by this grouping variable]*), data = *[name of dataframe]*, REML = *[TRUE/FALSE]*

take a break…

Part 1: LM to MLM Estimation

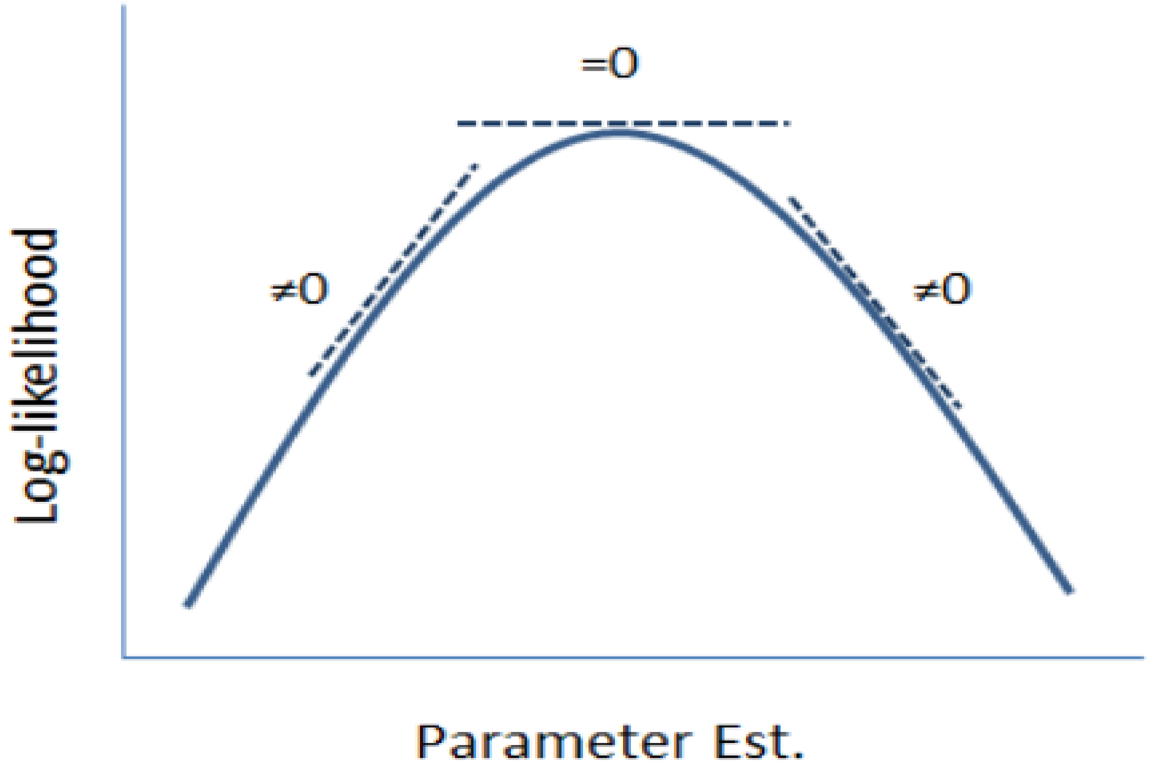
Part 2: Inference in MLM

Model Estimation

- For standard linear models, we can calculate the parameters using a *closed form solution*.
- Multilevel models are too complicated, we *estimate* all the parameters using an iterative procedure like Maximum Likelihood Estimation (MLE).

Model Estimation: MLE

Aim: find the values for the unknown parameters that maximize the probability of obtaining the observed data. How: This is done via finding values for the parameters that maximize the (log) likelihood function.

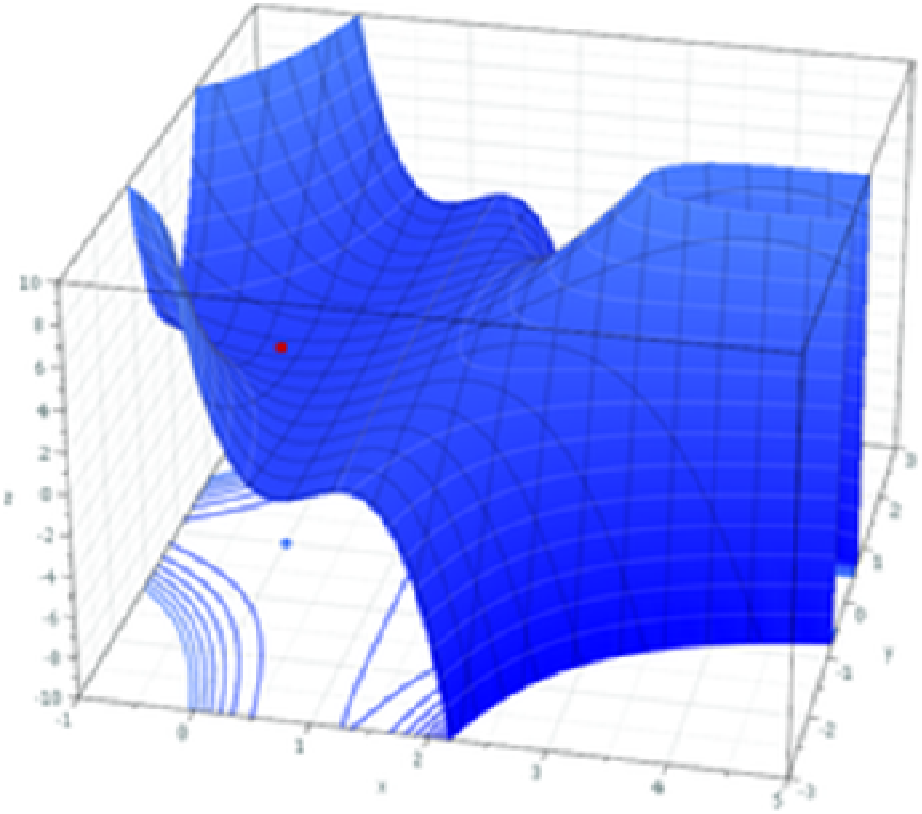


Model Estimation: (log)Likelihood

- Data = multiple observations: $1, \ldots, n$
- From our axioms of probability, we can combine these *i.i.d* by multiplication to get our likelihood of our parameters given our entire sample
- Instead of taking the **product** of the individual likelihoods, we can take the **summation** of the log-likelihoods
  - This is considerably easier to do, and can be achieved because multiplication is addition on a log scale.

Model Estimation: MLE

In multilevel models, our parameter space is more complex (e.g. both fixed effects and variance components).



Model Estimation: ML vs REML

- Standard ML results in biased estimates of variance components.

Model Estimation: ML vs REML

- Standard ML results in biased estimates of variance components.
- Restricted Maximum Likelihood (REML) is the default in `lmer()`.

Model Estimation: ML vs REML

- Standard ML results in biased estimates of variance components.
- Restricted Maximum Likelihood (REML) is the default in `lmer()`.
  - REML separates the estimation of fixed and random parts of the model, leading to less biased estimates of the variance components.

Model Estimation: ML vs REML

- Standard ML results in biased estimates of variance components.
- Restricted Maximum Likelihood (REML) is the default in `lmer()`.
    - REML separates the estimation of fixed and random parts of the model, leading to less biased estimates of the variance components.
- **Use ML to compare models that differ in their fixed effects.**

take a break...

Part 1: LM to MLM A Visual Explanation

Part 2: Inference in MLM

Data

> 200 pupils from 20 schools completed a survey containing the Emotion Dysregulation Scale (EDS) and the Child Routines Questionnaire (CRQ).

```
crq <- read_csv("https://uoepsy.github.io/data/crqdata.csv")
head(crq)
```

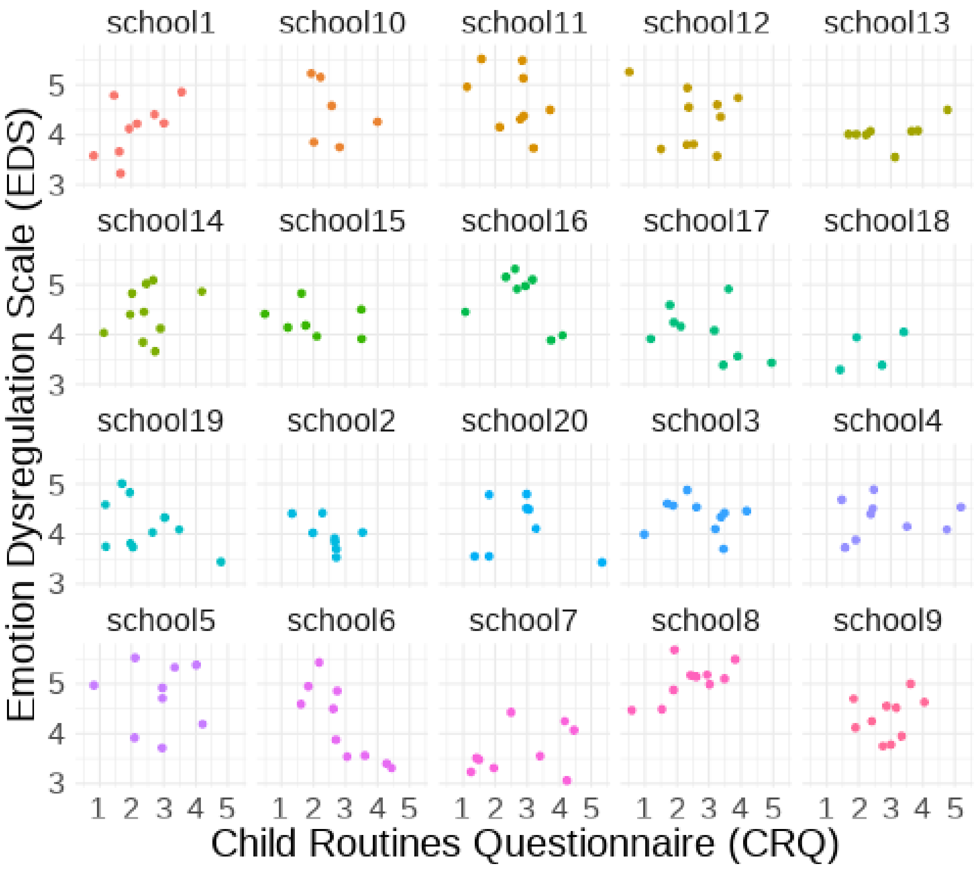
```
## # A tibble: 6 × 6
##   emot_dysreg   crq int       schoolid sleep   age
##         <dbl> <dbl> <chr>     <chr>    <chr> <dbl>
## 1        4.12  1.92 Treatment school1  <8hr     14
## 2        3.22  1.65 Treatment school1  <8hr     11
## 3        4.86  3.56 Treatment school1  <8hr     16
## 4        4.79  1.45 Treatment school1  8hr+     16
## 5        3.58  0.81 Treatment school1  <8hr     12
## 6        4.41  2.71 Treatment school1  <8hr     15
```

Data

> 200 pupils from 20 schools completed a survey containing the Emotion Dysregulation Scale (EDS) and the Child Routines Questionnaire (CRQ).

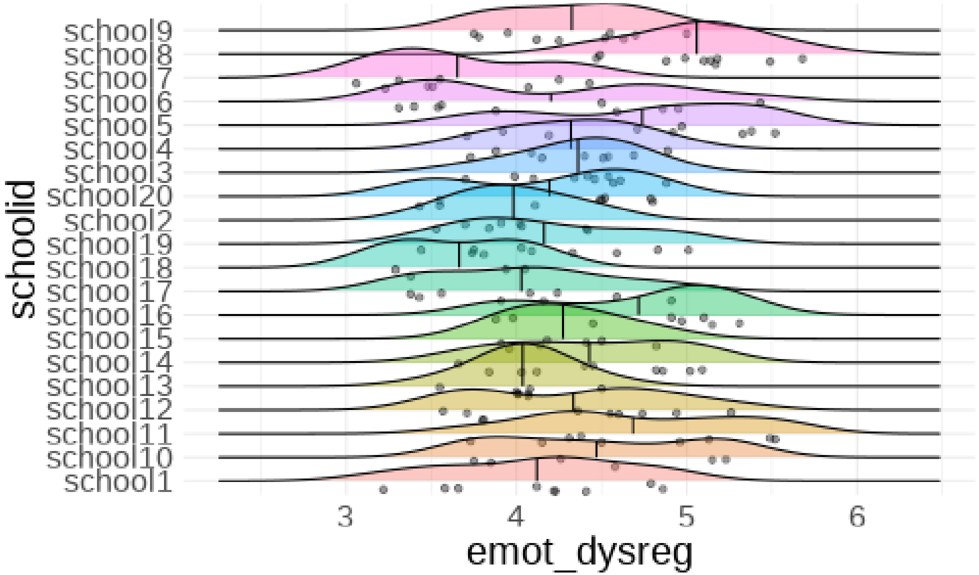
```
schoolplots <-
  ggplot(crq, aes(x = crq, y = emot_dysreg,
                  col = schoolid)) +
  geom_point()+
  facet_wrap(~schoolid) +
  guides(col = "none") +
  labs(x = "Child Routines Questionnaire (CRQ)",
       y = "Emotion Dysregulation Scale (EDS)") +
  themedapr3()
```

```
schoolplots
```



ICC

```r
library(ggridges)
ggplot(crq, aes(x = emot_dysreg, y = schoolid,
                fill = schoolid)) +
  geom_density_ridges(jittered_points = TRUE,
                      position = "raincloud", alpha = .4,
                      quantile_lines=TRUE,
                      quantile_fun=function(x,...) mean(x)) +
  guides(fill=FALSE) +
  themedapr3()
```




```r
library(ICC)
ICCbare(schoolid, emot_dysreg, data = crq)
```

```
## [1] 0.2443
```

Reminder: the Intraclass Correlation Coefficient is ratio of variance between clusters to the total variance (variance within + variance between).

R: fitting lm

```r
lm_mod <- lm(emot_dysreg ~ 1 + crq, data = crq)
summary(lm_mod)
```

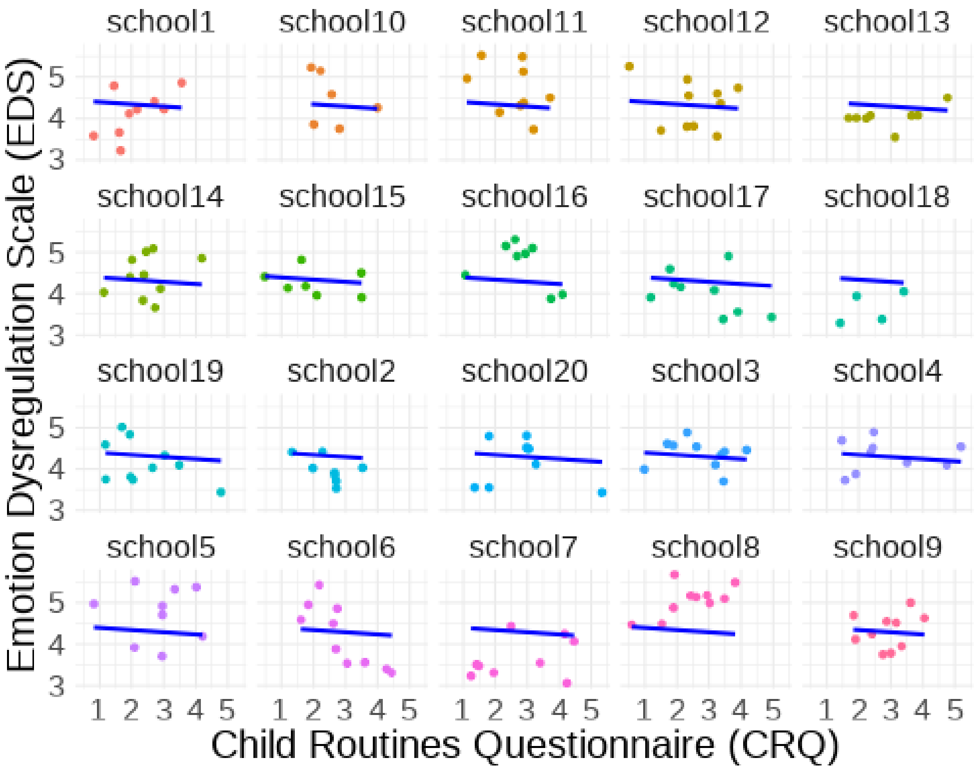
```
##
## Call:
## lm(formula = emot_dysreg ~ 1 + crq, data = crq)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1643 -0.4667  0.0158  0.4333  1.3338
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.4470     0.1259   35.31   <2e-16 ***
## crq          -0.0525     0.0448   -1.17     0.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.581 on 172 degrees of freedom
## Multiple R-squared:  0.00794,    Adjusted R-squared:  0.00217
## F-statistic: 1.38 on 1 and 172 DF,  p-value: 0.242
```

R: fitting lm

```
lm_mod <- lm(emot_dysreg ~ 1 + crq, data = crq)
summary(lm_mod)
```

```
##
## Call:
## lm(formula = emot_dysreg ~ 1 + crq, data = crq)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1643 -0.4667 0.0158  0.4333  1.3338
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.4470     0.1259   35.31   <2e-16 ***
## crq          -0.0525     0.0448   -1.17     0.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.581 on 172 degrees of freedom
## Multiple R-squared:  0.00794,    Adjusted R-squared:  0.00217
## F-statistic: 1.38 on 1 and 172 DF,  p-value: 0.242
```

```
schoolplots +
  geom_line(aes(y=fitted(lm_mod)), col = "blue", lwd=1)
```



R: Adding a random intercept

vary the intercept by schools.

```
library(lme4)
ri_mod <- lmer(emot_dysreg ~ 1 + crq +
                 (1 | schoolid), data = crq)
summary(ri_mod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: emot_dysreg ~ 1 + crq + (1 | schoolid)
##    Data: crq
##
## REML criterion at convergence: 290.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9203 -0.8709  0.0341  0.6536  2.3091
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  schoolid (Intercept) 0.0847   0.291
##  Residual             0.2578   0.508
## Number of obs: 174, groups:  schoolid, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   4.4299     0.1300   34.07
## crq          -0.0510     0.0402   -1.27
##
## Correlation of Fixed Effects:
```

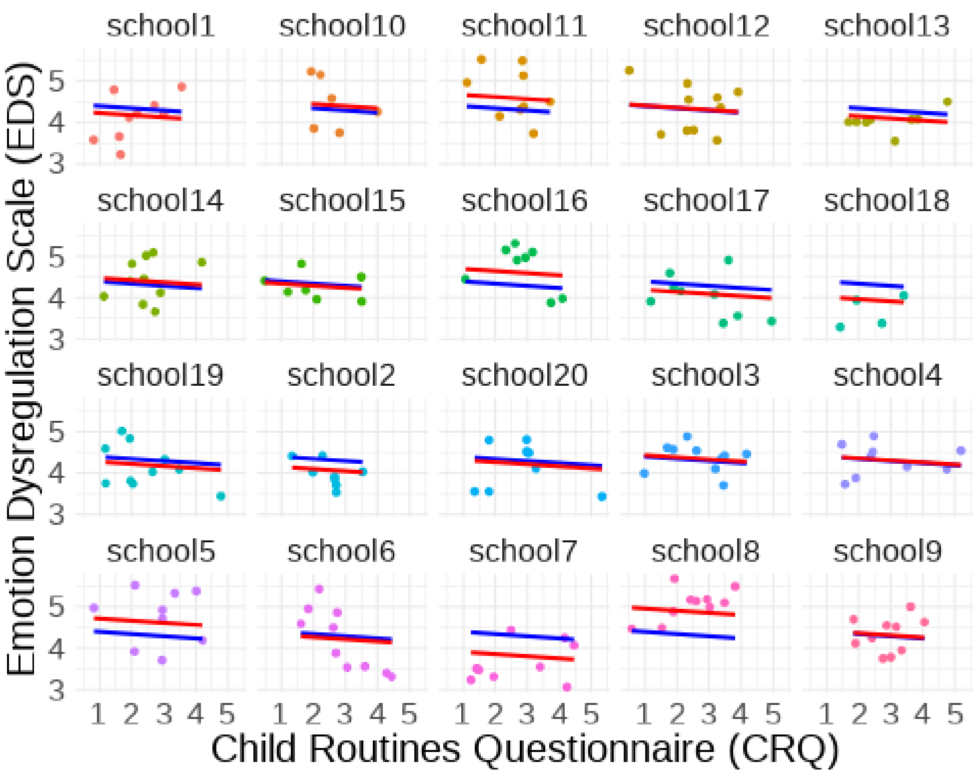
R: Adding a random intercept

vary the intercept by schools.

```
library(lme4)
ri_mod <- lmer(emot_dysreg ~ 1 + crq +
                 (1 | schoolid), data = crq)
summary(ri_mod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: emot_dysreg ~ 1 + crq + (1 | schoolid)
##    Data: crq
##
## REML criterion at convergence: 290.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9203 -0.8709  0.0341  0.6536  2.3091
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  schoolid (Intercept) 0.0847   0.291
##  Residual             0.2578   0.508
## Number of obs: 174, groups:  schoolid, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   4.4299     0.1300   34.07
## crq          -0.0510     0.0402   -1.27
##
## Correlation of Fixed Effects:
```

```
schoolplots +
  geom_line(aes(y=fitted(lm_mod)), col = "blue", lwd=1) +
  geom_line(aes(y=fitted(ri_mod)), col = "red", lwd=1)
```



R: Adding a random slope

vary the intercept and the effect (slope) of crq by schools

```
rs_mod <- lmer(emot_dysreg ~ crq +
                 (1 + crq | schoolid), data = crq)
summary(rs_mod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: emot_dysreg ~ crq + (1 + crq | schoolid)
##    Data: crq
##
## REML criterion at convergence: 288.6
##
## Scaled residuals:
##    Min      1Q Median     3Q    Max
## -1.879 -0.836  0.041  0.644  2.051
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  schoolid (Intercept) 0.242    0.492
##           crq         0.019    0.138    -0.80
##  Residual             0.239    0.489
## Number of obs: 174, groups:  schoolid, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   4.4377     0.1569   28.28
## crq          -0.0517     0.0506   -1.02
##
## Correlation of Fixed Effects:
```

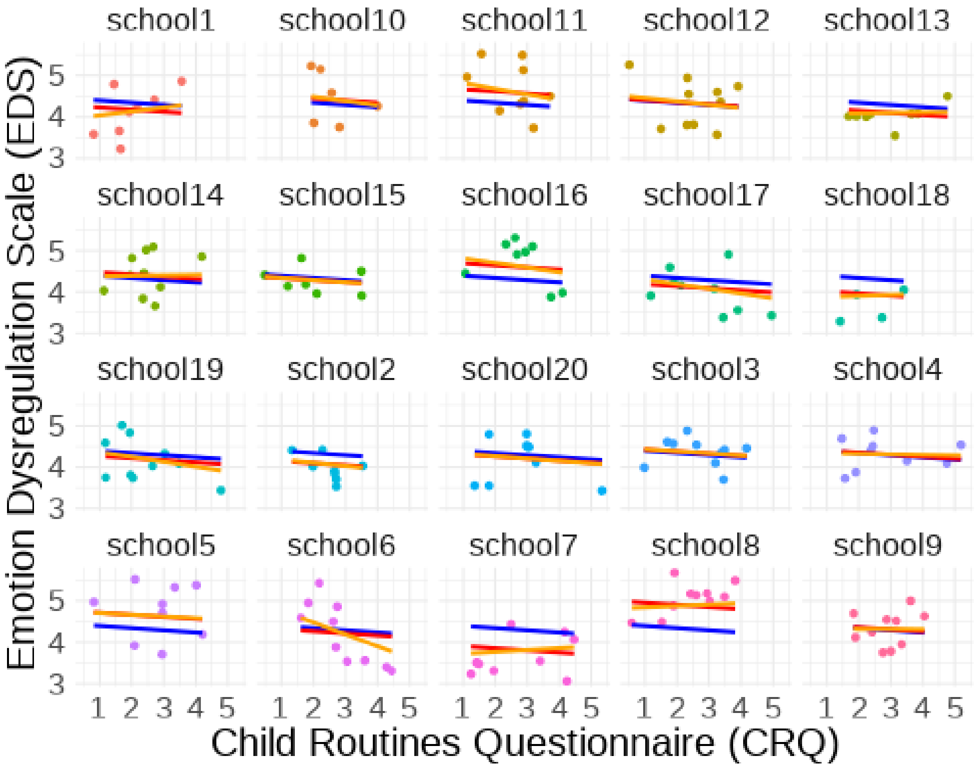
R: Adding a random slope

vary the intercept and the effect (slope) of crq by schools

```
rs_mod <- lmer(emot_dysreg ~ crq +
                  (1 + crq | schoolid), data = crq)
summary(rs_mod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: emot_dysreg ~ crq + (1 + crq | schoolid)
##    Data: crq
##
## REML criterion at convergence: 288.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.879 -0.836  0.041  0.644  2.051
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  schoolid (Intercept) 0.242    0.492
##           crq         0.019    0.138    -0.80
##  Residual             0.239    0.489
## Number of obs: 174, groups:  schoolid, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   4.4377     0.1569   28.28
## crq          -0.0517     0.0506   -1.02
##
## Correlation of Fixed Effects:
```

```
schoolplots +
  geom_line(aes(y=fitted(lm_mod)), col = "blue", lwd=1) +
  geom_line(aes(y=fitted(ri_mod)), col = "red", lwd=1) +
  geom_line(aes(y=fitted(rs_mod)), col = "orange", lwd=1)
```

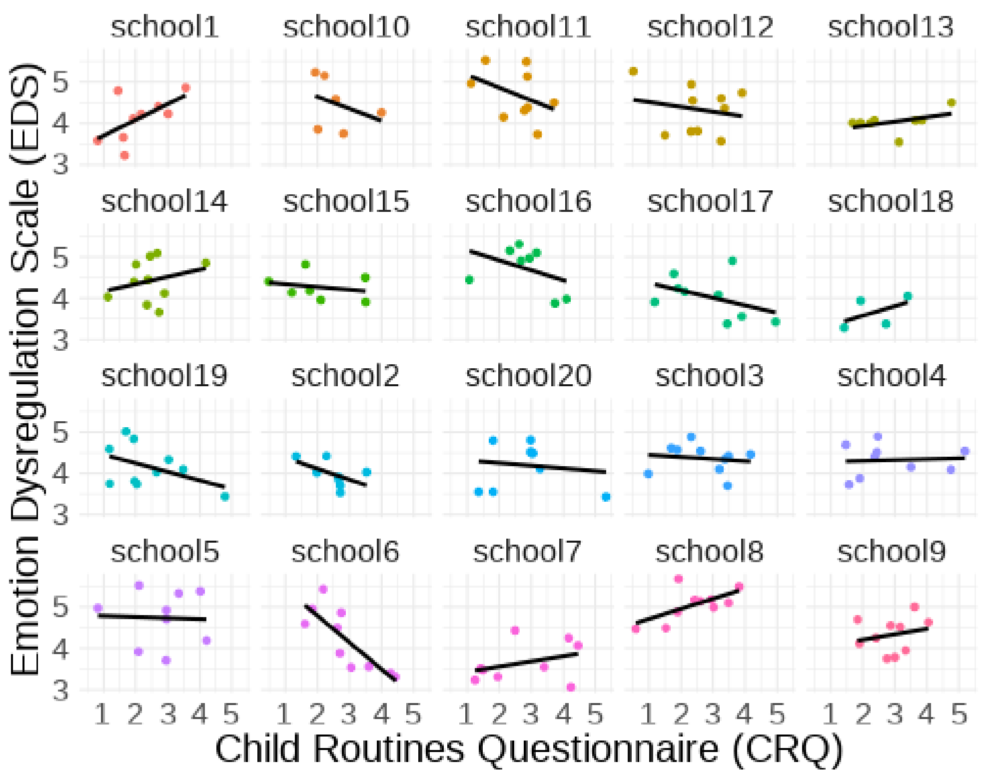


Partial Pooling vs No Pooling

Why not fit a fixed effect adjustment to the slope of x for each group?
`lm(y ~ x * group)`?

```
fe_mod <- lm(emot_dysreg ~ crq * schoolid, data = crq)
```

```
schoolplots +
  geom_line(aes(y=fitted(fe_mod)), col = "black", lwd=1)
```



Partial Pooling vs No Pooling

- We talked last week about how this results in a lot of output. With 20 schools, we get: intercept at reference school, adjustment for every other school, the effect of x at reference school, adjustment to effect of x for every other school.

```
length(coef(fe_mod))
```

```
## [1] 40
```

- information is not combined in anyway (data from school i contributes to differences from reference school to school i, but nothing else. No overall estimates)

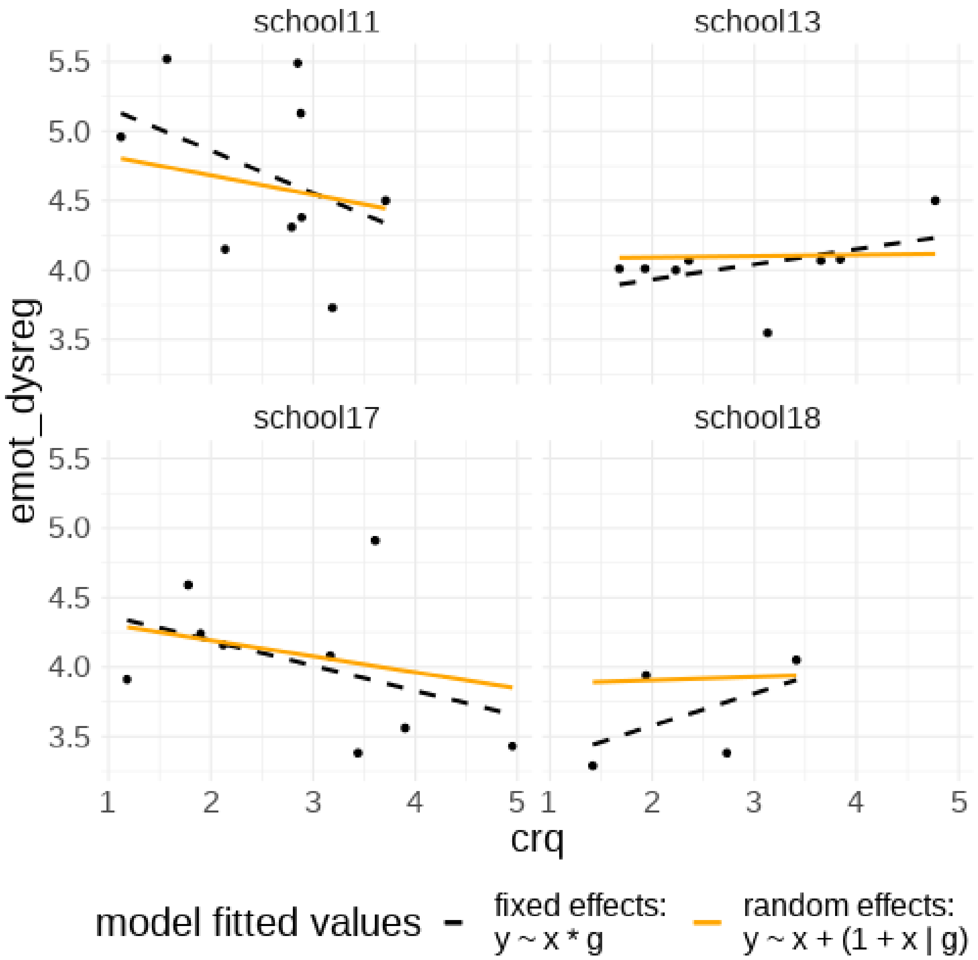
Partial Pooling vs No Pooling

- We talked last week about how this results in a lot of output. With 20 schools, we get: intercept at reference school, adjustment for every other school, the effect of x at reference school, adjustment to effect of x for every other school.

```
length(coef(fe_mod))
```

```
## [1] 40
```

- information is not combined in anyway (data from school i contributes to differences from reference school to school i, but nothing else. No overall estimates)



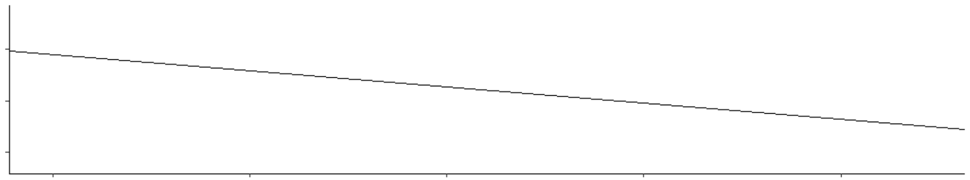
take a break...

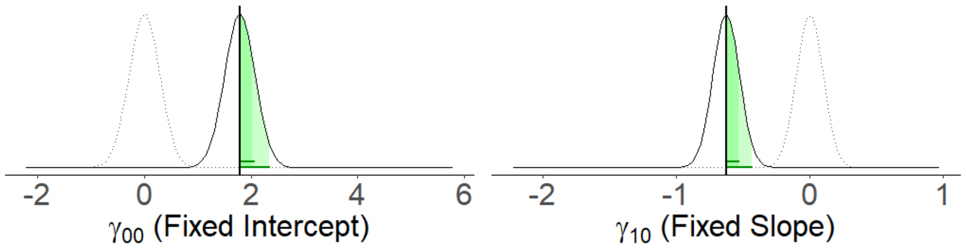
Part 1: LM to MLM lme4 Output

Part 2: Inference in MLM

Understanding MLM output

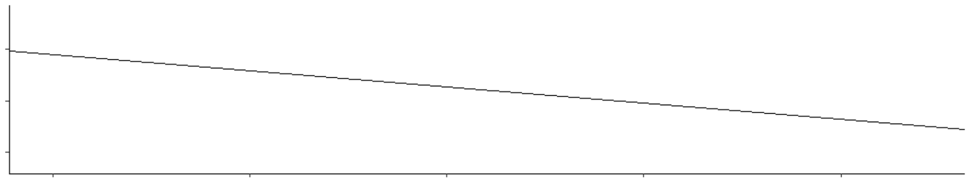
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ x + (1 + x | group)
##    Data: my_data
##
## REML criterion at convergence: 334.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1279 -0.7009  0.0414  0.6645  2.1010
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  group    (Intercept) 1.326    1.152
##           x           0.152    0.390    -0.88
##  Residual             0.262    0.512
## Number of obs: 170, groups:  group, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   1.7890     0.2858    6.26
## x            -0.6250     0.0996   -6.27
```

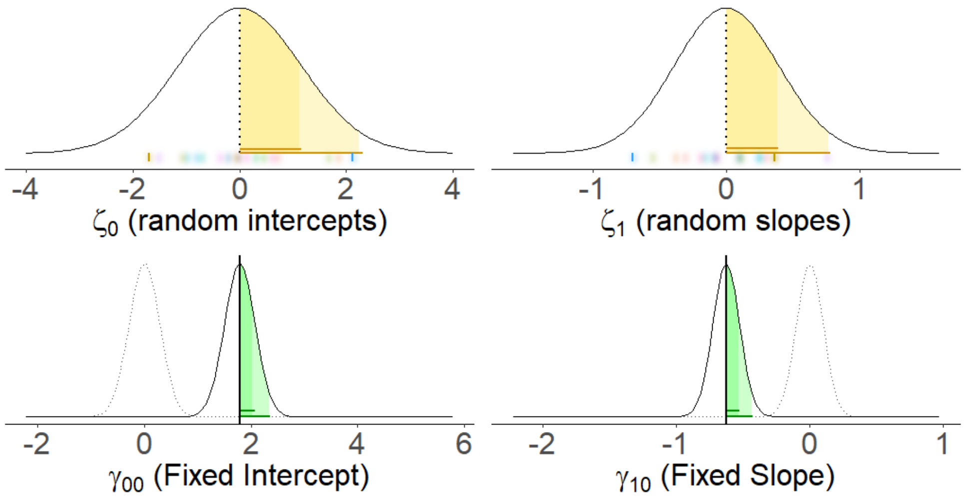




Understanding MLM output

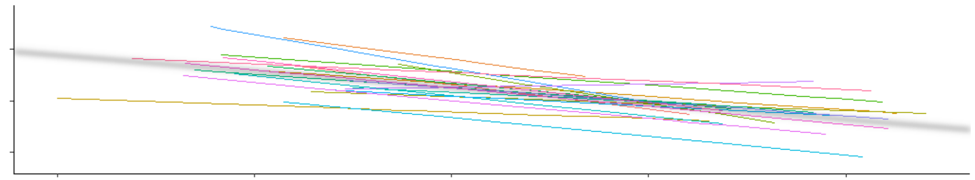
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ x + (1 + x | group)
##    Data: my_data
##
## REML criterion at convergence: 334.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1279 -0.7009  0.0414  0.6645  2.1010
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  group    (Intercept) 1.326    1.152
##           x           0.152    0.390    -0.88
##  Residual             0.262    0.512
## Number of obs: 170, groups:  group, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   1.7890     0.2858    6.26
## x            -0.6250     0.0996   -6.27
```

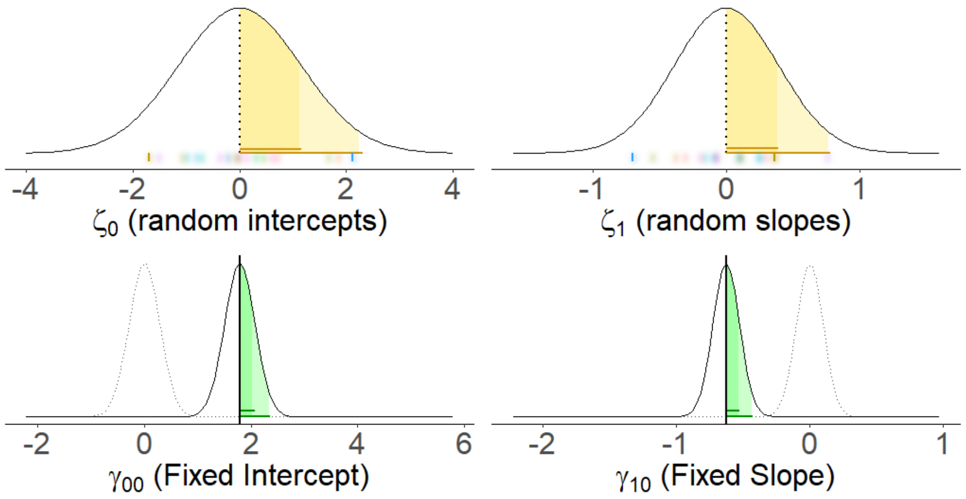




Understanding MLM output

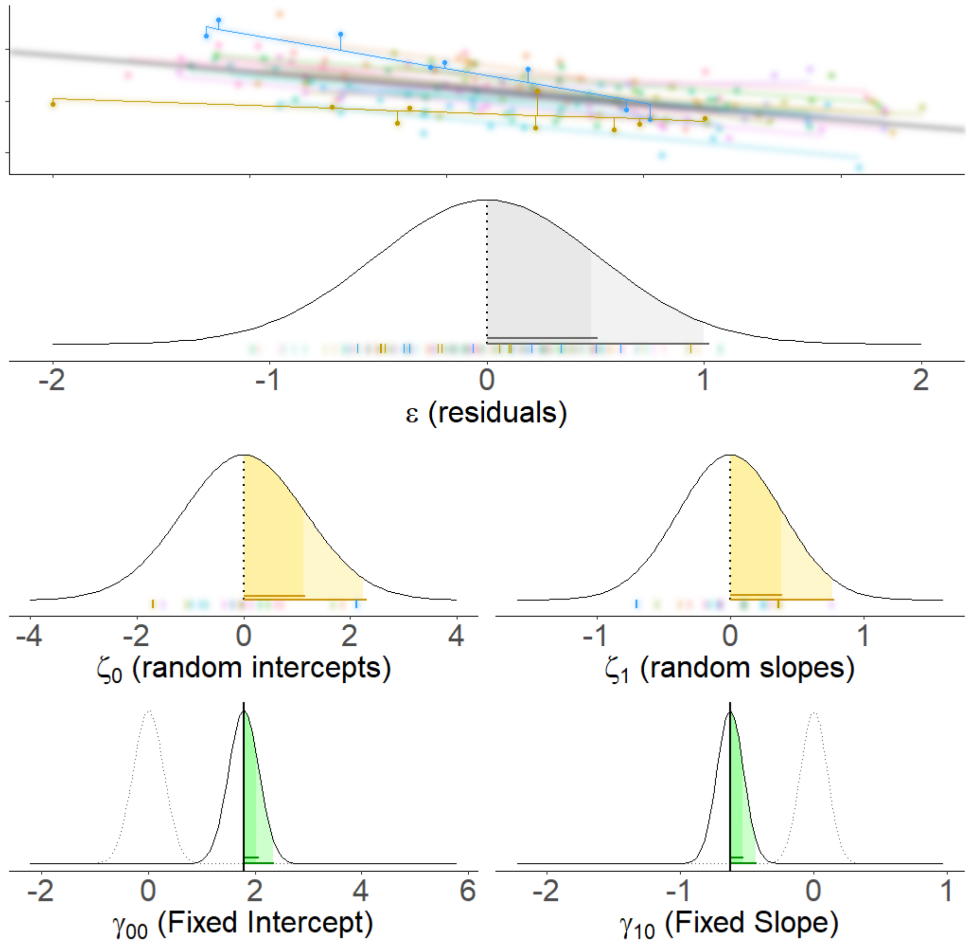
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ x + (1 + x | group)
##    Data: my_data
##
## REML criterion at convergence: 334.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1279 -0.7009  0.0414  0.6645  2.1010
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  group    (Intercept) 1.326    1.152
##           x           0.152    0.390    -0.88
##  Residual             0.262    0.512
## Number of obs: 170, groups:  group, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   1.7890     0.2858    6.26
## x            -0.6250     0.0996   -6.27
```





Understanding MLM output

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ x + (1 + x | group)
##    Data: my_data
##
## REML criterion at convergence: 334.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1279 -0.7009  0.0414  0.6645  2.1010
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  group    (Intercept) 1.326    1.152
##           x           0.152    0.390    -0.88
##  Residual             0.262    0.512
## Number of obs: 170, groups:  group, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   1.7890     0.2858    6.26
## x            -0.6250     0.0996   -6.27
```



Extracting MLM output

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ x + (1 + x | group)
##    Data: my_data
##
## REML criterion at convergence: 334.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1279 -0.7009  0.0414  0.6645  2.1010
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  group    (Intercept) 1.326    1.152
##           x           0.152    0.390    -0.88
##  Residual             0.262    0.512
## Number of obs: 170, groups:  group, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   1.7890     0.2858    6.26
## x            -0.6250     0.0996   -6.27
```

```
fixef(model)
```

```
## (Intercept)           x
##       1.789      -0.625
```

```
ranef(model)
```

```
##          (Intercept)       x
## school1       0.7019 -0.3113
## school10      1.8388 -0.3828
## school11     -0.0781  0.1098
## school12     -1.7005  0.3658
## school13     -1.0825   0.355
## ...              ...     ...
```

```
coef(model)
```

```
##          (Intercept)       x
## school1        2.491 -0.9363
## school10      3.6278 -1.0078
## school11       1.711 -0.5152
## school12      0.0885 -0.2592
## school13      0.7065   -0.27
## ...              ...     ...
```

ICC in lmer

```r
base_mod <- lmer(emot_dysreg ~ 1 + (1 | schoolid), data = crq
summary(base_mod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: emot_dysreg ~ 1 + (1 | schoolid)
##    Data: crq
##
## REML criterion at convergence: 287.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8585 -0.7964  0.0012  0.7119  2.3705
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  schoolid (Intercept) 0.0845   0.291
##  Residual             0.2588   0.509
## Number of obs: 174, groups:  schoolid, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   4.2960     0.0759    56.6
```

```r
0.0845 / (0.0845 + 0.2588)
```

```
## [1] 0.2461
```

Note: ICC is conditional on zero values of random-effects covariates. In other words, it has computed the ICC based on a value of zero for the random slope variable(s), so any interpretation of the ICC is also based on a value of zero for the slope variable(s).

Explained Variance in MLM

R^2

- Recall R^2 is proportion of variance explained
- In MLM, multiple variance components (not just ε). Do random effects "explain" variance?
    - "marginal R^2" = variance explained due to fixed effects
    - "conditional R^2" = variance explained due to fixed + random

```
library(MuMIn)
mod1 <- lmer(emot_dysreg ~ 1 + crq + (1 | schoolid), data = c
r.squaredGLMM(mod1)
```

```
##              R2m    R2c
## [1,] 0.007321 0.2529
```

Explained Variance in MLM

R^2

- Recall R^2 is proportion of variance explained
- In MLM, multiple variance components (not just ε). Do random effects "explain" variance?
  - "marginal R^2" = variance explained due to fixed effects
  - "conditional R^2" = variance explained due to fixed + random

```
library(MuMIn)
mod1 <- lmer(emot_dysreg ~ 1 + crq + (1 | schoolid), data = c
r.squaredGLMM(mod1)
```

```
##              R2m    R2c
## [1,] 0.007321 0.2529
```

Proportional Reduction in Variance (PRV)

- $PRV = \frac{\text{var}_{m0} - \text{var}_{m1}}{\text{var}_{m0}}$
- where var_{m0} and var_{m1} are variance components from models with and without a parameter.

take a break…

Part 1: LM to MLM **Example**

Part 2: Inference in MLM

MLM Example

Researchers are interested in how cognition changes over time.

```
cogtime <- read_csv("https://uoepsy.github.io/data/cogtimerpm
cogtime <- cogtime %>%
  mutate(across(c(participant, sexFemale, alc), factor))
head(cogtime, 12L)
```

```
## # A tibble: 12 × 6
##    visit_n sexFemale   cog y_bin participant alc
##      <dbl> <fct>     <dbl> <dbl> <fct>       <fct>
##  1       1 1          56.1     1 1           1
##  2       2 1          71.5     1 1           1
##  3       3 1          68.9     1 1           0
##  4       4 1          73.0     1 1           0
##  5       5 1          59.4     1 1           0
##  6       6 1          76.4     1 1           1
##  7       7 1          72.1     1 1           1
##  8       8 1          64.2     1 1           1
##  9       9 1          74.3     1 1           0
## 10      10 1          69.7     1 1           1
## 11       1 1          82.2     1 2           1
## 12       2 1          65.1     1 2           0
```

MLM Example

Researchers are interested in how cognition changes over time.

```
cogtime <- read_csv("https://uoepsy.github.io/data/cogtimerpm
cogtime <- cogtime %>%
  mutate(across(c(participant, sexFemale, alc), factor))
head(cogtime, 12L)
```

```
ggplot(cogtime, aes(x=visit_n, y = cog, col=participant))+
  geom_line(alpha = 0.5)+
  guides(col=FALSE)+
  scale_x_continuous(breaks=1:10)+
  themedapr3()
```

```
## # A tibble: 12 × 6
##    visit_n sexFemale   cog y_bin participant alc
##      <dbl> <fct>     <dbl> <dbl> <fct>       <fct>
##  1       1 1          56.1     1 1           1
##  2       2 1          71.5     1 1           1
##  3       3 1          68.9     1 1           0
##  4       4 1          73.0     1 1           0
##  5       5 1          59.4     1 1           0
##  6       6 1          76.4     1 1           1
##  7       7 1          72.1     1 1           1
##  8       8 1          64.2     1 1           1
##  9       9 1          74.3     1 1           0
## 10      10 1          69.7     1 1           1
## 11       1 1          82.2     1 2           1
## 12       2 1          65.1     1 2           0
```

MLM Example

determining our random effect structure

- multiple data-points per participant: **1 | participant**

MLM Example

determining our random effect structure

- multiple data-points per participant: **1 | participant**
- explanatory variable of interest (`visit_n`) varies *within* participants: **visit_n | participant**

MLM Example

determining our random effect structure

- multiple data-points per participant: **1 | participant**
- explanatory variable of interest (`visit_n`) varies *within* participants: **visit_n | participant**
- allow by-participant intercepts to correlate with by-participant slopes: **1 + visit_n | participant** (more on this in future weeks)

MLM Example

determining our random effect structure

- multiple data-points per participant: **1 | participant**
- explanatory variable of interest (`visit_n`) varies *within* participants: **visit_n | participant**
- allow by-participant intercepts to correlate with by-participant slopes: **1 + visit_n | participant** (more on this in future weeks)

fitting the model

```
cogtime_model <- lmer(cog ~ visit_n + (1 + visit_n | participant), data = cogtime)
```

MLM Example

model output

```
summary(cogtime_model)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: cog ~ visit_n + (1 + visit_n | participant)
##    Data: cogtime
##
## REML criterion at convergence: 1357
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.274 -0.663 -0.091  0.577  3.227
##
## Random effects:
##  Groups      Name        Variance Std.Dev. Corr
##  participant (Intercept) 10.06    3.17
##              visit_n      1.22    1.11     0.69
##  Residual                37.93    6.16
## Number of obs: 200, groups:  participant, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    68.56       1.18    58.2
## visit_n        -1.22       0.29    -4.2
##
## Correlation of Fixed Effects:
##         (Intr)
## visit_n -0.019
```

raw data

```
ggplot(cogtime, aes(x=visit_n, y = cog, col=participant))+
  geom_path(alpha = 0.5)+
  guides(col=FALSE)+
  scale_x_continuous(breaks=1:10)+
  themedapr3()
```

MLM Example: Plotting the model

sjPlot::plot_model()

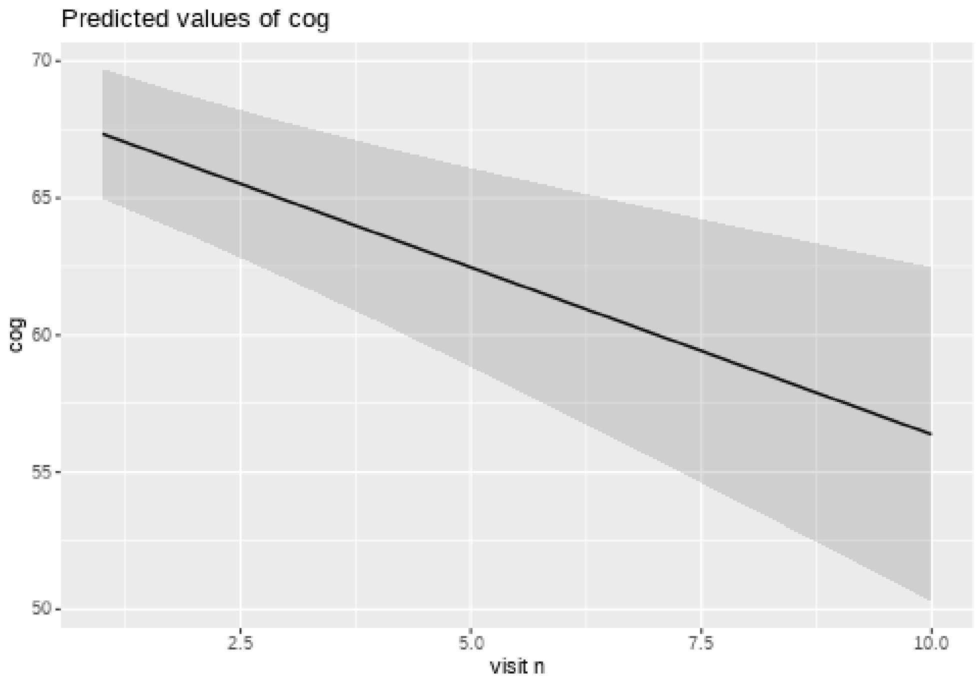
```
library(sjPlot)
plot_model(cogtime_model, type="pred")
```

MLM Example: Plotting the model

sjPlot::plot_model()

```
library(sjPlot)
plot_model(cogtime_model, type="pred")
```

```
## $visit_n
```



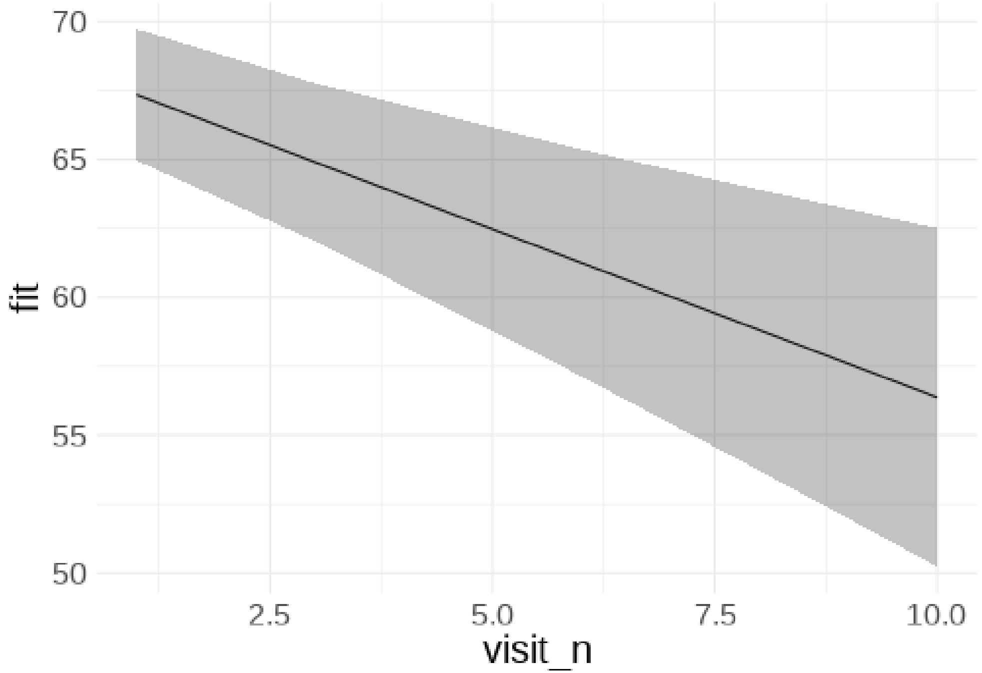
MLM Example: Plotting the model

effects::effect()

```
library(effects)
as.data.frame(effect("visit_n",cogtime_model))
```

```
##   visit_n   fit    se lower upper
## 1       1 67.34 1.208 64.96 69.72
## 2       3 64.90 1.452 62.04 67.77
## 3       6 61.25 2.083 57.14 65.35
## 4       8 58.81 2.582 53.72 63.90
## 5      10 56.37 3.110 50.24 62.50
```

```
as.data.frame(effect("visit_n",cogtime_model)) %>%
  ggplot(.,aes(x=visit_n, y=fit))+
  geom_line()+
  geom_ribbon(aes(ymin=lower,ymax=upper), alpha=.3)+
  themedapr3()
```



MLM Example: Plotting the model

broom.mixed::augment() for cluster-specific fits

```
library(broom.mixed)
augment(cogtime_model)
```

```
## # A tibble: 200 × 14
##      cog visit_n participant .fitted  .resid   .hat  .cooksd .fixed   .mu .offset
##    <dbl>   <dbl> <fct>         <dbl>   <dbl>  <dbl>    <dbl>  <dbl> <dbl>   <dbl>
##  1  56.1       1 1              69.7 -13.6   0.0782 0.225      67.3  69.7       0
##  2  71.5       2 1              69.6   1.93  0.0684 0.00386    66.1  69.6       0
##  3  68.9       3 1              69.5  -0.611 0.0653 0.000369   64.9  69.5       0
##  4  73.0       4 1              69.3   3.62  0.0690 0.0138     63.7  69.3       0
##  5  59.4       5 1              69.2  -9.79  0.0793 0.118      62.5  69.2       0
##  6  76.4       6 1              69.1   7.32  0.0964 0.0833     61.2  69.1       0
##  7  72.1       7 1              69.0   3.16  0.120  0.0204     60.0  69.0       0
##  8  64.2       8 1              68.8  -4.64  0.151  0.0594     58.8  68.8       0
##  9  74.3       9 1              68.7   5.65  0.188  0.120      57.6  68.7       0
## 10  69.7      10 1              68.6   1.16  0.232  0.00695    56.4  68.6       0
## # … with 190 more rows, and 4 more variables: .sqrtXwt <dbl>, .sqrtrwt <dbl>,
## #   .weights <dbl>, .wtres <dbl>
```

```
ggplot(augment(cogtime_model),
       aes(x=visit_n, y=.fitted,
           col=participant))+
  geom_line() +
  guides(col=FALSE)+
  themedapr3()
```

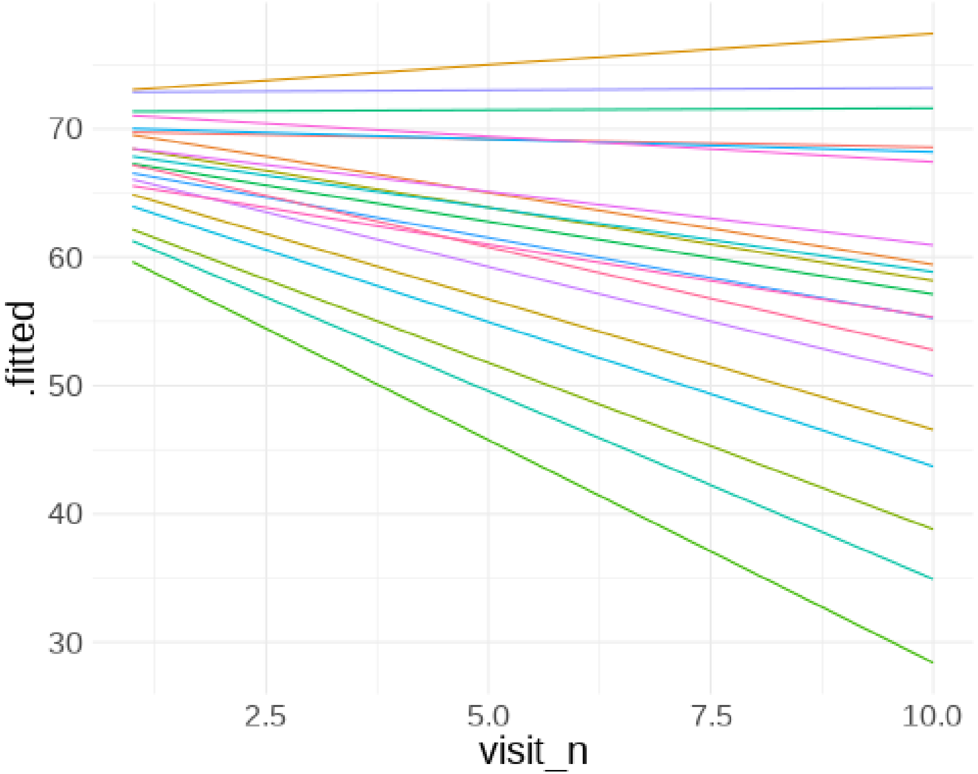
MLM Example: Plotting the model

broom.mixed::augment() for cluster-specific fits

```
library(broom.mixed)
augment(cogtime_model)
```

```
## # A tibble: 200 × 14
##      cog visit_n participant .fitted  .resid   .hat  .cooksd .fixed
##    <dbl>   <dbl> <fct>         <dbl>   <dbl>  <dbl>    <dbl>  <dbl>
##  1  56.1       1 1              69.7 -13.6   0.0782 0.225      67.3
##  2  71.5       2 1              69.6   1.93  0.0684 0.00386    66.1
##  3  68.9       3 1              69.5  -0.611 0.0653 0.000369   64.9
##  4  73.0       4 1              69.3   3.62  0.0690 0.0138     63.7
##  5  59.4       5 1              69.2  -9.79  0.0793 0.118      62.5
##  6  76.4       6 1              69.1   7.32  0.0964 0.0833     61.2
##  7  72.1       7 1              69.0   3.16  0.120  0.0204     60.0
##  8  64.2       8 1              68.8  -4.64  0.151  0.0594     58.8
##  9  74.3       9 1              68.7   5.65  0.188  0.120      57.6
## 10  69.7      10 1              68.6   1.16  0.232  0.00695    56.4
## # … with 190 more rows, and 4 more variables: .sqrtXwt <dbl>, .sqrt
## #   .weights <dbl>, .wtres <dbl>
```

```
ggplot(augment(cogtime_model),
       aes(x=visit_n, y=.fitted,
           col=participant))+
  geom_line() +
  guides(col=FALSE)+
  themedapr3()
```



MLM Example: Tables

```
library(sjPlot)
tab_model(cogtime_model)
```

	cog		
Predictors	*Estimates*	*CI*	*p*
(Intercept)	68.56	66.25 – 70.87	**<0.001**
visit_n	-1.22	-1.79 – -0.65	**<0.001**
Random Effects			
σ^2	37.93		
τ_{00} participant	10.06		
τ_{11} participant.visit_n	1.22		
ρ_{01} participant	0.69		
ICC	0.69		
N participant	20		
Observations	200		
Marginal R^2 / Conditional R^2	0.092 / 0.717		

Summary

- We can extend our linear model equation to model certain parameters as random cluster-level adjustments around a fixed center.

- $y_i = \beta_0 \cdot 1 \ + \ \beta_1 \cdot x_i + \varepsilon_i$
  becomes
  $y_{ij} = \beta_{0i} \cdot 1 + \beta_{1i} \cdot x_{ij} + \varepsilon_{ij}$
  $\beta_{0i} = \gamma_{00} + \zeta_{0i}$

- We can express this as one equation if we prefer: $y_{ij} = \underbrace{(\gamma_{00} + \zeta_{0i})}_{\beta_{0i}} \cdot 1 + \beta_{1i} \cdot x_{ij} + \varepsilon_{ij}$

- This allows us to model cluster-level variation around the intercept ("random intercept") and around slopes ("random slope").

- We can fit this using the **lme4** package in R

End of Part 1

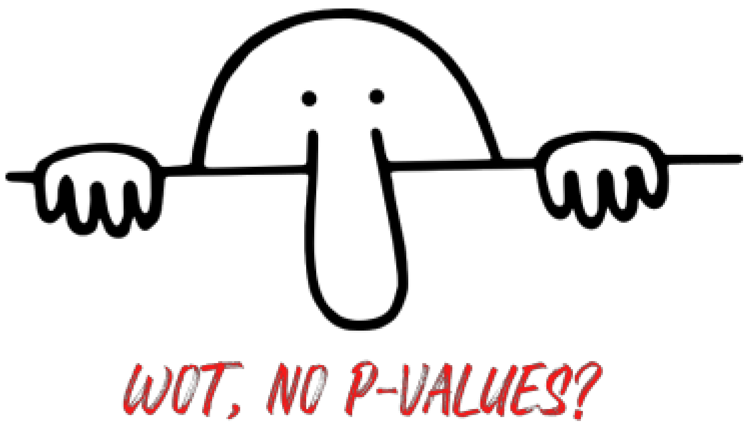
Part 1: LM to MLM

Part 2: Inference in MLM

you might have noticed...

```
summary(cogtime_model)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: cog ~ visit_n + (1 + visit_n | participant)
##    Data: cogtime
##
## REML criterion at convergence: 1357
##
## Scaled residuals:
##    Min      1Q Median      3Q     Max
## -2.274 -0.663 -0.091  0.577  3.227
##
## Random effects:
##  Groups      Name        Variance Std.Dev. Corr
##  participant (Intercept) 10.06    3.17
##              visit_n      1.22    1.11     0.69
##  Residual                37.93    6.16
## Number of obs: 200, groups:  participant, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    68.56       1.18    58.2
## visit_n        -1.22       0.29    -4.2
##
## Correlation of Fixed Effects:
##         (Intr)
## visit_n -0.019
```



Why no p-values?

Extensive debate about how best to test parameters from MLMs.

Why no p-values?

Extensive debate about how best to test parameters from MLMs.

In simple LM, we test the reduction in residual SS (sums of squares), which follows an F distribution with a known df.

$$F = \frac{MS_{model}}{MS_{residual}} = \frac{SS_{model}/df_{model}}{SS_{residual}/df_{residual}}$$

$$\begin{aligned} df_{model} &= k \\ df_{residual} &= n - k - 1 \end{aligned}$$

Why no p-values?

Extensive debate about how best to test parameters from MLMs.

In simple LM, we test the reduction in residual SS (sums of squares), which follows an F distribution with a known df.

$$F = \frac{MS_{model}}{MS_{residual}} = \frac{SS_{model}/df_{model}}{SS_{residual}/df_{residual}}$$

$$\begin{aligned} df_{model} &= k \\ df_{residual} &= n - k - 1 \end{aligned}$$

The t-statistic for a coefficient in a simple regression model is the square root of F ratio between models with and without that parameter.

- Such F will have 1 numerator degree of freedom (and $n - k - 1$ denominator degrees of freedom).
- The analogous t-distribution has $n - k - 1$ degrees of freedom

Why no p-values?

In MLM, the distribution of a test statistic when the null hypothesis is true is **unknown.**

Why no p-values?

In MLM, the distribution of a test statistic when the null hypothesis is true is **unknown.**

Under very specific conditions (normally distributed outcome variable, perfectly balanced designs), we can use an F distribution and correctly determine the denominator df.

But for most situations:

- unclear how to calculate denominator df
- unclear whether the test statistics even follow an F distribution

Options for inference

1. df approximations
2. Likelihood Ratio Tests
3. Bootstrap

Satterthwaite df approximation

- There are some suggested approaches to approximating the denominator df.

- Loading the package **lmerTest** will fit your models and print the summary with p-values approximated by the Satterthwaite method.

```
library(lmerTest)
full_model <- lmer(cog ~  1 + visit_n + (1 + visit_n | partic
summary(full_model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: cog ~ 1 + visit_n + (1 + visit_n | participant)
##    Data: cogtime
##
## REML criterion at convergence: 1357
##
## Scaled residuals:
##     Min      1Q Median      3Q     Max
## -2.274 -0.663 -0.091  0.577  3.227
##
## Random effects:
##  Groups      Name        Variance Std.Dev. Corr
##  participant (Intercept) 10.06    3.17
##              visit_n      1.22    1.11     0.69
##  Residual                37.93    6.16
## Number of obs: 200, groups:  participant, 20
##
## Fixed effects:
##             Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)    68.56       1.18 19.00    58.2  < 2e-16 ***
## visit_n        -1.22       0.29 19.00    -4.2  0.00048 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr)
## visit_n -0.019
```

Kenward-Rogers df approximations

- The **pbkrtest** package implements the slightly more reliable Kenward-Rogers method.

```
library(pbkrtest)
restricted_model <- lmer(cog ~ 1 + (1 + visit_n | participant), data = cogtime)
full_model <- lmer(cog ~ 1 + visit_n + (1 + visit_n | participant), data = cogtime)
KRmodcomp(full_model, restricted_model)
```

```
## large : cog ~ 1 + visit_n + (1 + visit_n | participant)
## small : cog ~ 1 + (1 + visit_n | participant)
##        stat  ndf  ddf F.scaling p.value    
## Ftest 17.7  1.0 19.0         1 0.00048 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Likelihood ratio tests

We can also conduct a Likelihood Ratio Test (LRT).

```
anova(restricted_model, full_model, test = "LRT")
```

```
## Data: cogtime
## Models:
## restricted_model: cog ~ 1 + (1 + visit_n | participant)
## full_model: cog ~ 1 + visit_n + (1 + visit_n | participant)
##                  npar  AIC  BIC logLik deviance Chisq Df Pr(>Chisq)
## restricted_model    5 1382 1398   -686     1372
## full_model          6 1370 1390   -679     1358  13.2  1    0.00029 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Compares the log-likelihood of two competing models.

- what is the "likelihood"?
  - a function that associates to a parameter the probability (or probability density) of observing the given sample data.

- ratio of two likelihoods is asymptotically χ^2-square distributed.
  - **this means for small samples it may be unreliable**

Options for inference

1. df approximations - assumes F-distributed just with unknown ddf.

2. Likelihood Ratio Tests - differences in logLik are only asymptotically `$\chi^2$` distributed.

3. Bootstrap

    - Parametric Bootstrap
      assumes that explanatory variables are fixed and that model specification and the distributions such as $\zeta_i \sim N(0, \sigma_\zeta)$ and $\varepsilon_i \sim N(0, \sigma_\varepsilon)$ are correct.

    - Case-based Bootstrap
      minimal assumptions - we just need to ensure that we correctly specify the hierarchical dependency of data.
      requires decision of at which levels to resample.
      (discussed more next week)

Parametric Bootstrap

The idea here is that in order to do inference on the effect of a (set of) predictor(s), you

1. fit the reduced model (without the predictors) to the data.

Parametric Bootstrap

The idea here is that in order to do inference on the effect of a (set of) predictor(s), you

1. fit the reduced model (without the predictors) to the data.

2. Do many times:

    - simulate data from the reduced model
    - fit both the reduced and the full model to the simulated (null) data
    - compute some statistic(s), e.g. likelihood ratio.

Parametric Bootstrap

The idea here is that in order to do inference on the effect of a (set of) predictor(s), you

1. fit the reduced model (without the predictors) to the data.

2. Do many times:

    - simulate data from the reduced model
    - fit both the reduced and the full model to the simulated (null) data
    - compute some statistic(s), e.g. likelihood ratio.

3. Compare the parameter estimates obtained from fitting models to the **data**, to the "null distribution" constructed in step 2.

Parametric Bootstrap

The idea here is that in order to do inference on the effect of a (set of) predictor(s), you

1. fit the reduced model (without the predictors) to the data.

2. Do many times:

    - simulate data from the reduced model
    - fit both the reduced and the full model to the simulated (null) data
    - compute some statistic(s), e.g. likelihood ratio.

3. Compare the parameter estimates obtained from fitting models to the **data**, to the "null distribution" constructed in step 2.

Easy to do with `PBmodcomp()` in the **pbkrtest** package.

```
library(pbkrtest)
PBmodcomp(full_model, restricted_model)
```

```
## Bootstrap test; time: 27.43 sec; samples: 1000; extremes: 1;
## Requested samples: 1000 Used samples: 989 Extremes: 1
## large : cog ~ 1 + visit_n + (1 + visit_n | participant)
## cog ~ 1 + (1 + visit_n | participant)
##         stat df p.value
## LRT     13.1  1 0.00029 ***
## PBtest 13.1    0.00202 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Summary

- Lots of debate around how best to conduct statistical inferences based on multi-level models.
    - denominator degrees of freedom can't be calculated, so traditional F tests cannot be conducted
- Lots of other options (approximations for df, likelihood ratio tests, bootstrap)
    - The choice is yours, but we recommend bootstrapping (of which there are also many different approaches!)

take a break…

Part 1: LM to MLM

Part 2: Inference in MLM Examples

Data

```
nursedf <- read_csv("https://uoepsy.github.io/data/nurse_stress.csv")
nursedf <- nursedf %>%
  mutate(across(c(hospital, expcon, gender, wardtype, hospsize), factor))
head(nursedf)
```

```
## # A tibble: 6 × 10
##   hospital expcon nurse   age gender experien stress Zstress wardtype   hospsize
##   <fct>    <fct>  <dbl> <dbl> <fct>     <dbl>  <dbl>   <dbl> <fct>      <fct>
## 1 1        1          1    36 0            11      7    2.07 general c… large
## 2 1        1          2    45 0            20      7    2.07 general c… large
## 3 1        1          3    32 0             7      7    2.07 general c… large
## 4 1        1          4    57 1            25      6    1.04 general c… large
## 5 1        1          5    46 1            22      6    1.04 general c… large
## 6 1        1          6    60 1            22      6    1.04 general c… large
```

The files nurses.csv contains three-level simulated data from a hypothetical study on stress in hospitals. The data are from nurses working in wards nested within hospitals. It is a cluster-randomized experiment. In each of 25 hospitals, four wards are selected and randomly assigned to an experimental and a control condition. In the experimental condition, a training program is offered to all nurses to cope with job-related stress. After the program is completed, a sample of about 10 nurses from each ward is given a test that measures job-related stress. Additional variables are: nurse age (years), nurse experience (years), nurse gender (0 = male, 1 = female), type of ward (0 = general care, 1 = special care), and hospital size (0 = small, 1 = medium, 2 = large).
(From https://multilevel-analysis.sites.uu.nl/datasets/)

test of a single parameter

> After accounting for nurses' age, gender and experience, does having been offered a training program to cope with job-related stress appear to reduce levels of stress, and if so, by how much?

test of a single parameter

> After accounting for nurses' age, gender and experience, does having been offered a training program to cope with job-related stress appear to reduce levels of stress, and if so, by how much?

```
mod1 <- lmer(Zstress ~ 1 + experien + age + gender + expcon + (1 | hospital), data = nursedf)
summary(mod1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Zstress ~ 1 + experien + age + gender + expcon + (1 | hospital)
##    Data: nursedf
##
## REML criterion at convergence: 2218
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.950 -0.672  0.026  0.645  3.192
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  hospital (Intercept) 0.296    0.544
##  Residual             0.484    0.696
## Number of obs: 1000, groups:  hospital, 25
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  0.90568    0.14386    6.30
## experien    -0.05912    0.00638   -9.26
## age          0.01965    0.00320    6.15
## gender1     -0.46214    0.05045   -9.16
## expcon1     -0.76007    0.04403  -17.26
##
```

test of a single parameter

> After accounting for nurses' age, gender and experience, does having been offered a training program to cope with job-related stress appear to reduce levels of stress, and if so, by how much?

Likelihood Ratio Test:

```
mod0 <- lmer(Zstress ~ 1 + experien + age + gender + (1 | hospital), data = nursedf)
mod1 <- lmer(Zstress ~ 1 + experien + age + gender + expcon + (1 | hospital), data = nursedf)
anova(mod0, mod1, test="Chisq")
```

```
## Data: nursedf
## Models:
## mod0: Zstress ~ 1 + experien + age + gender + (1 | hospital)
## mod1: Zstress ~ 1 + experien + age + gender + expcon + (1 | hospital)
##      npar  AIC  BIC logLik deviance Chisq Df Pr(>Chisq)
## mod0    6 2461 2490  -1224     2449
## mod1    7 2202 2236  -1094     2188   261  1     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

test of a single parameter

> After accounting for nurses' age, gender and experience, does having been offered a training program to cope with job-related stress appear to reduce levels of stress, and if so, by how much?

Parametric Bootstrap

```
mod0 <- lmer(Zstress ~ 1 + experien + age + gender + (1 | hospital), data = nursedf)
mod1 <- lmer(Zstress ~ 1 + experien + age + gender +  expcon + (1 | hospital), data = nursedf)
PBmodcomp(mod1, mod0)
```

```
## Bootstrap test; time: 23.01 sec; samples: 1000; extremes: 0;
## large : Zstress ~ 1 + experien + age + gender + expcon + (1 | hospital)
## Zstress ~ 1 + experien + age + gender + (1 | hospital)
##         stat df p.value
## LRT      261  1  <2e-16 ***
## PBtest   261      0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

test of a single parameter

> After accounting for nurses' age, gender and experience, does having been offered a training program to cope with job-related stress appear to reduce levels of stress, and if so, **by how much?**

Parametric Bootstrap Confidence Intervals

```
mod1 <- lmer(Zstress ~ 1 + experien + age + gender +  expcon + (1 | hospital), data = nursedf)
confint(mod1, method="boot")
```

```
##                 2.5 %   97.5 %
## .sig01        0.39476  0.70600
## .sigma        0.66009  0.72655
## (Intercept)  0.61711  1.18782
## experien    -0.07269 -0.04748
## age          0.01368  0.02616
## gender1     -0.55448 -0.36142
## expcon1     -0.83685 -0.67703
```

test of a single parameter

> After accounting for nurses' age, gender and experience, does having been offered a training program to cope with job-related stress appear to reduce levels of stress, and if so, by how much?

Attendance of training programs on job-related stress was found to predict stress levels of nurses in 25 hospitals, beyond individual nurses' years of experience, age and gender (Parametric Bootstrap Likelihood Ratio Test statistic = 260.919, p<.001). Having attended the training program was associated with a decrease in -0.7601 (Bootstrap 95% CI [-0.84, -0.68]) standard deviations on the measure of job-related stress.

testing that several parameters are simultaneously zero

> Do ward type and hospital size influence levels of stress in nurses beyond the effects of age, gender, training and experience?

Likelihood Ratio Test

```
mod0 <- lmer(Zstress ~ experien + age + gender + expcon + (1 | hospital), data = nursedf)
mod1 <- lmer(Zstress ~ experien + age + gender + expcon + wardtype + hospsize + (1 | hospital), data = nursedf)
anova(mod0, mod1, test="Chisq")
```

```
## Data: nursedf
## Models:
## mod0: Zstress ~ experien + age + gender + expcon + (1 | hospital)
## mod1: Zstress ~ experien + age + gender + expcon + wardtype + hospsize + (1 | hospital)
##      npar  AIC  BIC logLik deviance Chisq Df Pr(>Chisq)
## mod0    7 2202 2236  -1094     2188
## mod1   10 2194 2243  -1087     2174    14  3     0.0029 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

testing that several parameters are simultaneously zero

> Do ward type and hospital size influence levels of stress in nurses beyond the effects of age, gender, training and experience?

Kenward-Rogers df-approximation

```
mod0 <- lmer(Zstress ~ experien + age + gender + expcon + (1 | hospital), data = nursedf)
mod1 <- lmer(Zstress ~ experien + age + gender + expcon + wardtype + hospsize + (1 | hospital), data = nursedf)
KRmodcomp(mod1, mod0)
```

```
## large : Zstress ~ experien + age + gender + expcon + wardtype + hospsize +
##     (1 | hospital)
## small : Zstress ~ experien + age + gender + expcon + (1 | hospital)
##         stat   ndf   ddf F.scaling p.value
## Ftest  4.99  3.00 40.27     0.988  0.0049 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

testing that several parameters are simultaneously zero

> Do ward type and hospital size influence levels of stress in nurses beyond the effects of age, gender, training and experience?

Parametric Bootstrap

```
mod0 <- lmer(Zstress ~ experien + age + gender + expcon + (1 | hospital), data = nursedf)
mod1 <- lmer(Zstress ~ experien + age + gender + expcon + wardtype + hospsize + (1 | hospital), data = nursedf)
PBmodcomp(mod1, mod0)
```

```
## Bootstrap test; time: 22.05 sec; samples: 1000; extremes: 3;
## large : Zstress ~ experien + age + gender + expcon + wardtype + hospsize +
##     (1 | hospital)
## Zstress ~ experien + age + gender + expcon + (1 | hospital)
##         stat df p.value
## LRT     13.8  3  0.0032 **
## PBtest 13.8     0.0040 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

testing random effects

are you sure you want to?

- Justify the random effect structure based on study design, theory, and practicalities more than tests of significance.
- If needed, the **RLRsim** package can test a single random effect (e.g. `lm()` vs `lmer()`).

```
library(RLRsim)
mod0 <- lm(stress ~ expcon + experien + age + gender + wardtype + hospsize, data = nursedf)
mod1 <- lmer(stress ~ expcon + experien + age + gender + wardtype + hospsize +
               (1 | hospital), data = nursedf)
exactLRT(m = mod1, m0 = mod0)
```

```
##
##      simulated finite sample distribution of LRT. (p-value based on 10000
##      simulated values)
##
## data:
## LRT = 240, p-value <2e-16
```

End