Learning Objectives

1. Define a multilevel measurement model (MLMM)
2. Identify when an MLMM is needed
3. Describe and execute an MLMM in a multilevel modelling framework
4. Describe and execute an MLMM in a structural equation modelling framework

Module Overview

- 5 sections of content
  - Review of multilevel modelling
  - Review of measurement modelling
  - Multilevel measurement modelling (MLMM) overview
  - MLMM in MLM framework
  - MLMM in SEM framework

- Code and data for each module available to download

About the Authors

Mairead Shaw

- Ph.D. student in Quantitative Psychology at McGill University
- Research on effect sizes in multilevel modelling and measurement in replication studies
- Teaches multilevel modelling
  - Co-author of open-source teaching materials at www.learn-mlms.com

Jessica K. Flake, Ph.D.

- Assistant Professor of Quantitative Psychology at McGill University
- Assistant Director for Methods at the Psychological Science Accelerator
- Research on measurement practices and the appropriate and transparent use of latent variable models in psychological research
- Teaches courses on introductory statistics, measurement theory, and multilevel modelling
  - Co-author of open-source teaching materials at www.learn-mlms.com

Module Citation

Shaw, M. & Flake, J. K. (2023). Multilevel Measurement Models [Digital ITEMS Module 34]. *Educational Measurement: Issues and Practice, 34*(4), 82.

Multilevel Modelling Overview

1

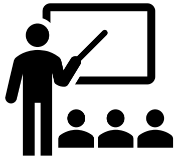
Section Learning Objectives

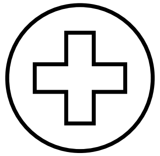
1 Multilevel Modelling Overview

- Understand when and why to use multilevel models
- Recognize when the model is cross-sectional or repeated measures
- Write and understand multilevel modelling equations
- Interpret output for fixed and random effects

Clustered Data Structures

- Common in educational and psychological research


Students within classes


Patients within clinicians


Trials within people (repeated measures)

Why Care About Clustering?

- At least two levels of variance: within and between clusters
  - Level 1 = students
  - Level 2 = class
- What if we neglect clustering?
  - Inflates sample size, deflates standard errors, increases likelihood of Type 1 Error
  - We miss out on interesting multilevel questions!

Example Equations

Outcome: student math achievement (math)

Level 1 predictor: hours spent studying (hours)

Level 2 predictor: teacher's years of experience (texp)

- Random intercept and random slope

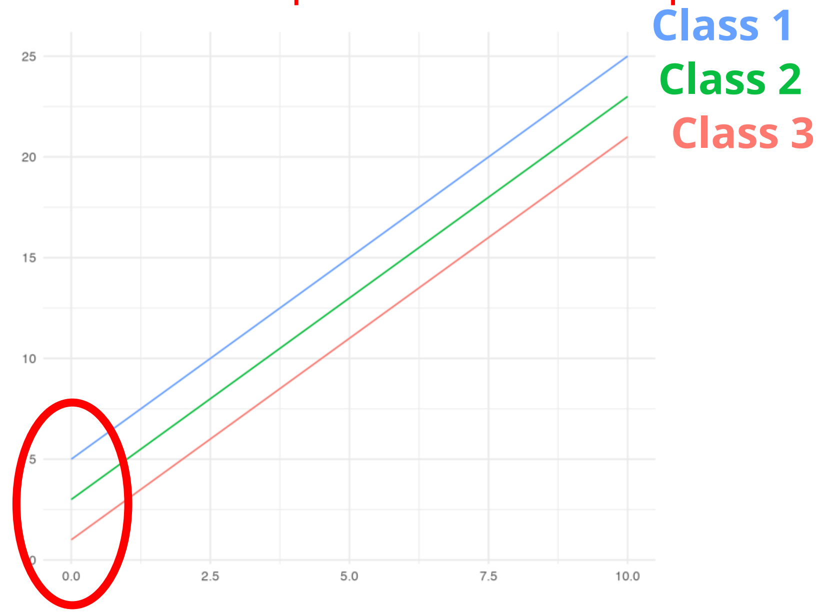
Random Intercept and Random Slope

Outcome: student math achievement (math)

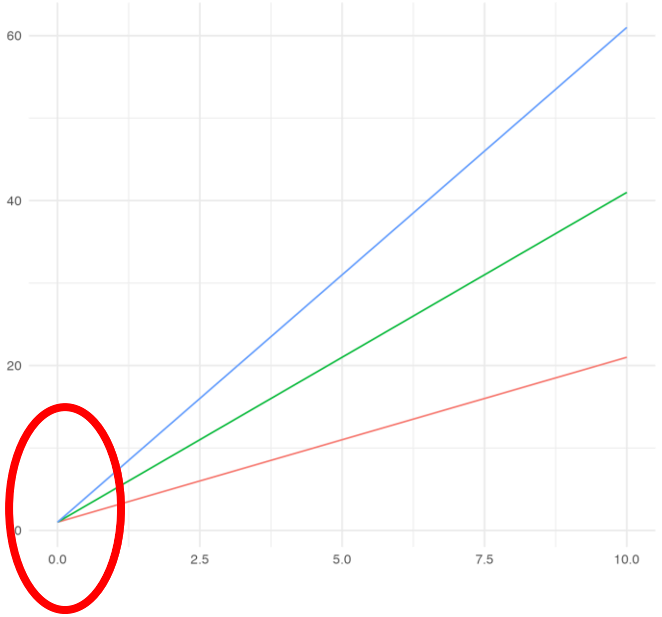
Level 1 predictor: hours spent studying (hours)

Level 2 predictor: teacher's years of experience (texp)

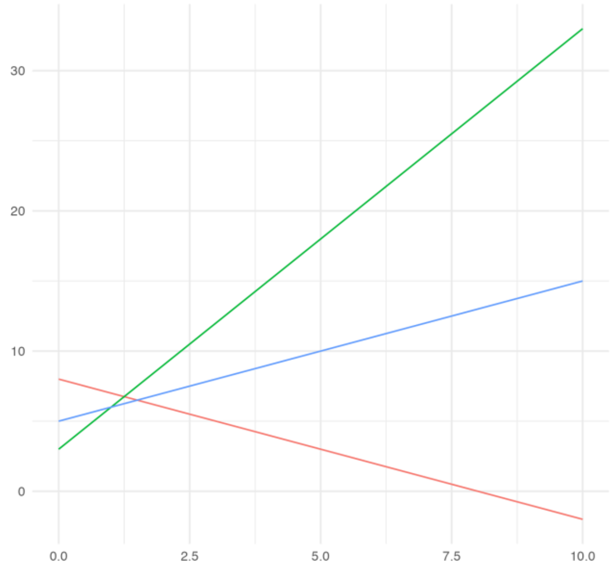
Random intercept, consistent slope



Random slope, consistent intercept



Random slope and intercept



Example Equations

Outcome: student math achievement (math)

Level 1 predictor: hours spent studying (hours)

Level 2 predictor: teacher's years of experience (texp)

- Random intercept and random slope

- Level 1 equation: $math_{ij} = \beta_{0j} + \beta_{1j} hours_{ij} + \epsilon_{ij}$
- Level 2 equations: $\beta_{0j} = \gamma_{00} + \gamma_{01} texp_j + U_{0j}$

- Total: $math_{ij} = \gamma_{00} + \gamma_{01} texp_j + \gamma_{10} hours_{ij} + U_{0j} + U_{1j} hours_{ij} + \epsilon_{ij}$

- Centering requires extra consideration in MLMs
- Without centering, effects are blends of within and between effects

Enders, C., & Tofighi, D. (2007). Centering Predictor Variables in Cross-Sectional Multilevel Models: A New Look at An Old Issue. *Psychological Methods, 12*, 121-138. doi:10.1037/1082-989X.12.2.121

Generalized Equation

Level 1: $y_{ij} = \beta_{0j} + \sum_{p=1}^{P} \beta_{pj} X_{pij} + \epsilon_{ij}$

Level 2 intercept: $\beta_{0j} = \gamma_{00} + \sum_{q=1}^{Q} \gamma_{0q} Z_{qj} + U_{0j}$

Level 2 slopes: $\beta_{pj} = \gamma_{p0} + \sum_{q=1}^{Q} \gamma_{pq} Z_{qj} + U_{pj}$

- $\epsilon \sim N(0, \sigma^2)$
- $Uj \sim MVN(0, \boldsymbol{\tau})$ → $\begin{bmatrix} \tau_{00} & \\ \tau_{01} & \tau_{11} \end{bmatrix}$

Repeated Measures

- Our example thus far has been cross-sectional, but sometimes data are repeated measures, with multiple responses on the outcome per person
- For example, students in a driver's ed class are shown 51 pictures, and they change slightly while they're looking
  - Outcome variable is log reaction time to notice a change
- Here, trials/measures are nested within students
- Level 1 variables are characteristics of the pictures
  - How big is the change? Is the picture related to driving? Etc.
- Level 2 variables are characteristics of the person
  - Gender? Age? etc.

Example From: Hoffman, L., & Rovine, M.J. (2007). Multilevel models for the experimental psychologist: Foundations and illustrative examples. *Behavior Research Methods, 39*(1), 101-117. https://doi.org/10.3758/bf03192848

Coding an MLM in R

- Data and example from Hoffman and Rovine (2007)
- Outcome of interest is log of reaction time for participants to detect a change to a picture
- Pictures varied on how salient the change in the picture was (salience)
- One of the primary questions was how age related to reaction time, given picture characteristics

Example From: Hoffman, L., & Rovine, M.J. (2007). Multilevel models for the experimental psychologist: Foundations and illustrative examples. *Behavior Research Methods, 39*(1), 101-117. https://doi.org/10.3758/bf03192848

Example Equations: Repeated Measures

Outcome: log of reaction time to notice difference between pictures (logRT)
Level 1 predictor: salience of change, centered around a constant (c_sal)
Level 2 predictor: age

- Random intercept and random slope
  - Random intercept = how a person's average RT deviates from mean across people, controlling for predictors
  - Random slope = how the relationship between salience and RT deviates from mean across people, controlling for age

Level 1 equation: $\mathrm{logRT_{ij}} = \beta_{0j} + \beta_{1j} c\_sal_{ij} + \epsilon_{ij}$
Level 2 intercept equation: $\beta_{0j} = \gamma_{00} + \gamma_{01} age_j + U_{0j}$
Level 2 slope equation: $\beta_{1j} = \gamma_{10} + U_{1j}$

Total: $\mathrm{logRT_{ij}} = \gamma_{00} + \gamma_{01} age_j + \gamma_{10} c\_sal_{ij} + U_{0j} + U_{1j} c\_sal_{ij} + \epsilon_{ij}$

Coding an MLM in R

- Repeated measures data, with many pictures per participant

id	sex	age	NAME	rt_sec	Item	meaning	salience	lg_rt	oldage	yrs65	c_mean	c_sal
1	1	20	rt_sec1	4.662	1	3.5	4.0	1.5394445	0	0	0.5	1.0
1	1	20	rt_sec2	6.660	2	0.0	3.0	1.8961195	0	0	-3.0	0.0
1	1	20	rt_sec3	6.602	3	4.0	2.0	1.8873726	0	0	1.0	-1.0
1	1	20	rt_sec4	1.332	4	4.0	4.0	0.2866816	0	0	1.0	1.0
1	1	20	rt_sec5	1.332	5	0.0	5.0	0.2866816	0	0	-3.0	2.0
1	1	20	rt_sec7	1.302	7	3.5	4.5	0.2639015	0	0	0.5	1.5
1	1	20	rt_sec8	2.601	8	5.0	1.5	0.9558960	0	0	2.0	-1.5
1	1	20	rt_sec9	23.287	9	0.0	5.0	3.1478953	0	0	-3.0	2.0
1	1	20	rt_sec10	3.330	10	4.0	3.5	1.2029723	0	0	1.0	0.5

Coding an MLM in R

- Data and example from Hoffman and Rovine (2007)

```
# package install, if needed
install.packages("lme4")

# Load dependency
library(lme4)

# Read data
hoffman2007 <- read.csv("hoffman2007.csv")

# Run model
model <- lmer(logRT ~ 1 + c_sal + age + (1 + c_sal|id),
              data = hoffman2007,
              REML = FALSE)

# Print output
summary(model)
```

Interpreting Output

Model input information

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: logRT ~ 1 + c_sal + age + (c_sal | id)
   Data: hoffman2007
```

Fit information

```
     AIC      BIC   logLik deviance df.resid
 16404.1  16452.7  -8195.0  16390.1     7639

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.5591 -0.7430 -0.1218  0.6220  4.1749
```

Random effects

```
Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 id       (Intercept) 0.0224610 0.14987
          c_sal       0.0004847 0.02201  -0.05
 Residual             0.4870854 0.69792
Number of obs: 7646, groups:  id, 153
```

Fixed effects

```
Fixed effects:
              Estimate Std. Error t value
(Intercept)  1.0429887  0.0259398   40.21
c_sal       -0.1518363  0.0075316  -20.16
age          0.0143094  0.0005319   26.90
```

Fixed Effects

- (Intercept) = γ_{00} = average log reaction time across all people across all photos, controlling for predictors
- c_sal = γ_{10} = average effect of photo salience on reaction time across all people, controlling for age
- age = γ_{01} = effect of age on average log reaction time, controlling for c_sal

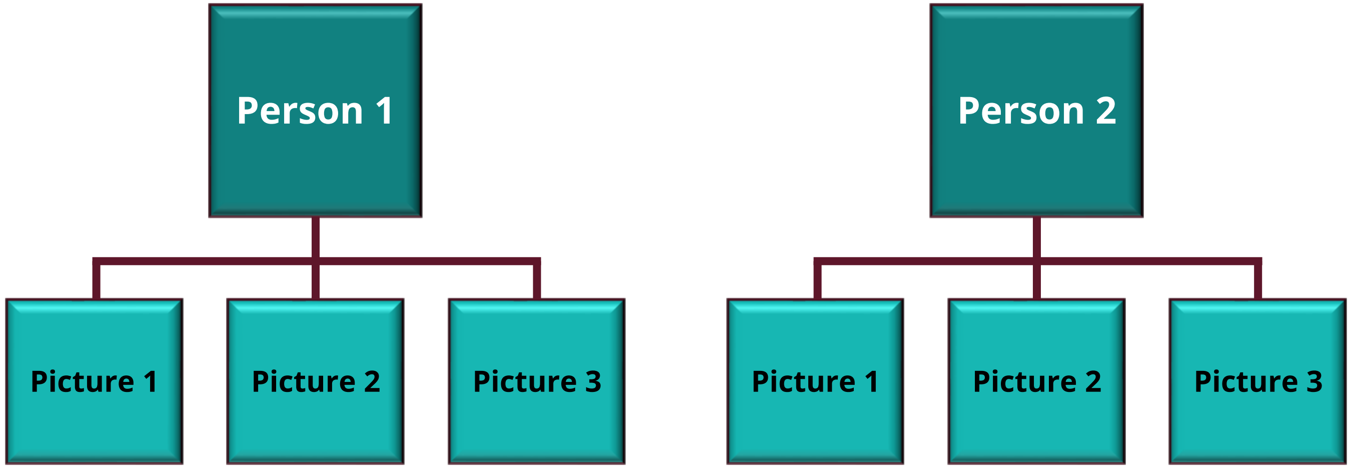
```
Fixed effects:
             Estimate Std. Error t value
(Intercept)  1.0429887  0.0259398   40.21
c_sal       -0.1518363  0.0075316  -20.16
age          0.0143094  0.0005319   26.90
```

Random Effects

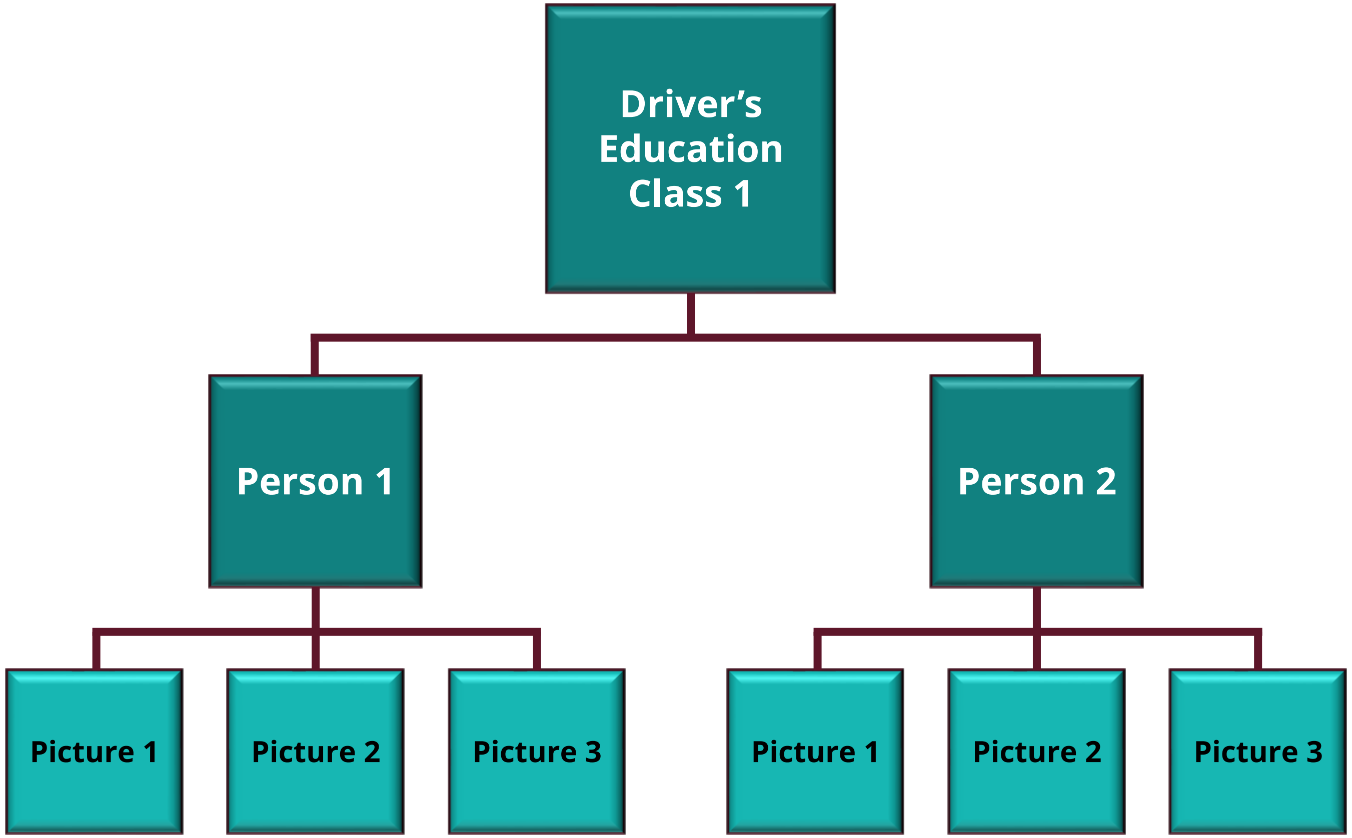
- (Intercept) = $\tau_0^2 = var(U_{0j})$ = variance describing how people's mean RT vary around grand mean intercept
- c_sal = $\tau_1^2 = var(U_{1j})$ = variance describing how people's relationship with salience and RT varies around grand mean slope
- Residual = $\sigma^2 = var(\epsilon_{ij})$ = residual describing how people's responses vary around their own mean

```
Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 id       (Intercept) 0.0224610 0.14987
          c_sal       0.0004847 0.02201  -0.05
 Residual             0.4870854 0.69792
Number of obs: 7646, groups:  id, 153
```

Two-Level MLM



Three-Level MLM



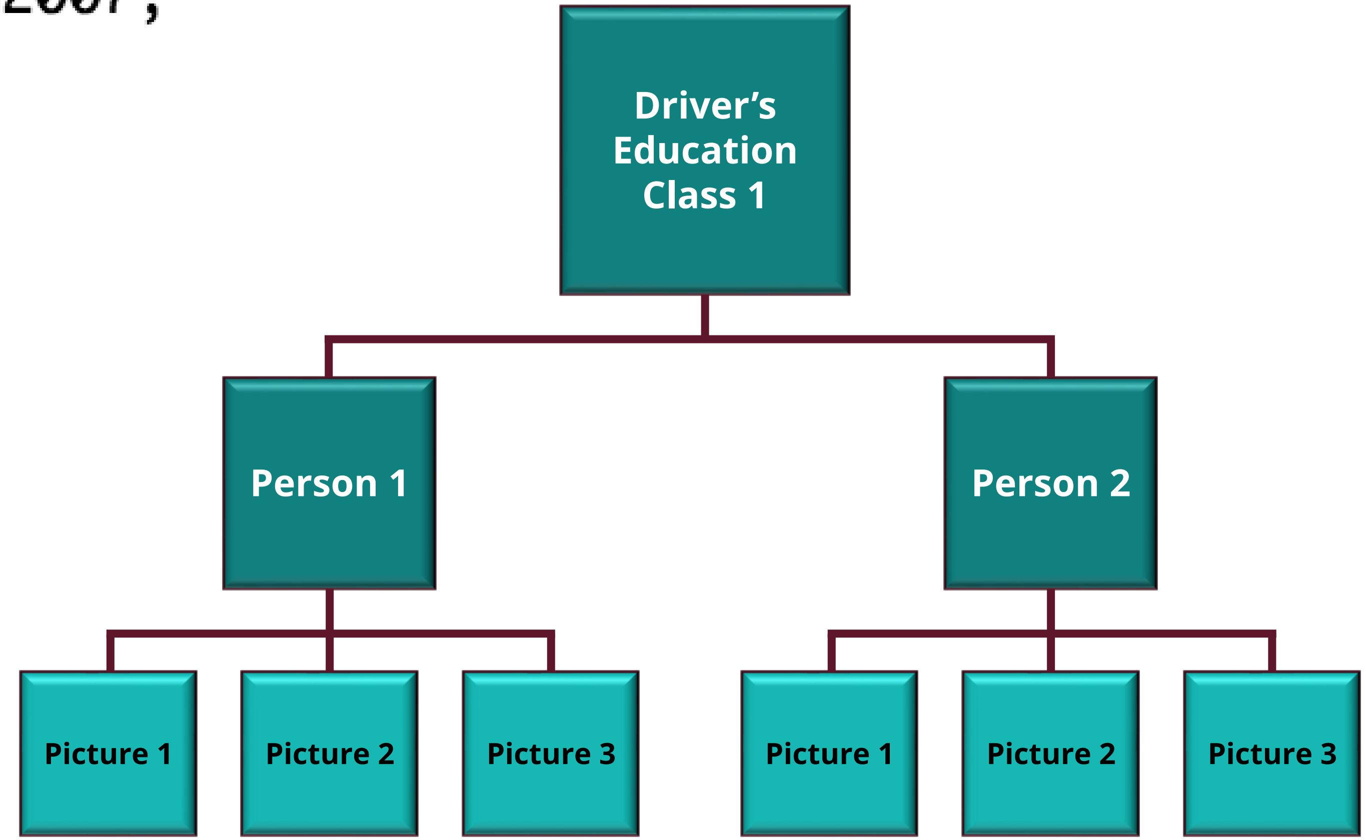
Three-Level MLM: Equations

Outcome: log of reaction time to notice difference between pictures (logRT)
Intercept-only model: no predictors yet
Level 1 equation: $\mathrm{logRT}_{\mathrm{ijk}} = \pi_{0jk} + \epsilon_{ijk}$
Level 2 intercept equation: $\pi_{0jk} = \beta_{00k} + r_{0jk}$
Level 3 intercept equation: $\beta_{00k} = \gamma_{000} + U_{00k}$
Total: $\mathrm{logRT}_{\mathrm{ijk}} = \gamma_{000} + U_{00k} + r_{0jk} + \epsilon_{ijk}$

- $\mathrm{logRT}_{\mathrm{ijk}}$ is log reaction time for picture *i*, person *j*, in class *k*
- γ_{000} is grand mean RT across all pictures, all people, all classes
- U_{00k} is residual term describing how class *k*'s average deviates from grand mean
- r_{0jk} is residual term describing how person *j* deviates from class mean
- ϵ_{ijk} is residual term describing how logRT to picture *i* deviates from person's mean

Three-Level MLM: Code

```r
# three-level null model
model3 <- lmer(logRT ~ 1 + (1|id) + (1|class),
               data = hoffman2007,
               REML = FALSE)
summary(model3)
```



Three-Level MLM: Output

```
Random effects:
 Groups   Name        Variance Std.Dev.
 id       (Intercept) 0.02556  0.1599
 class    (Intercept) 0.14045  0.3748
 Residual             0.51573  0.7181
Number of obs: 7646, groups:  id, 153; class, 12

Fixed effects:
            Estimate Std. Error      df t value      Pr(>|t|)
(Intercept)   1.5885     0.1093 12.0212   14.54 0.00000000544 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Long vs Wide Data

- We often work with “wide” data: one row per person

id	meaning1	meaning2	meaning3	salience1	salience2	salience3
1	3.5	0	4	4	3	2
2	3.5	0	4	4	3	2

- To run a repeated measures model, we need “long” data: one row per measurement

id	Item	meaning	salience
1	1	3.5	4
1	2	0.0	3
1	3	4.0	2
2	1	3.5	4
2	2	0.0	3
2	3	4.0	2

- In many cases, this will require transposing data

Summary

- Multilevel modelling is used in data structures where responses are clustered (e.g., students in classes, responses within person)
- It involves partitioning total variance to variance within and between clusters
- We reviewed formulae for cross-sectional, repeated measures, and three-level MLMs
- We implemented and interpreted a repeated measures and three-level MLM in R

Measurement Modelling Overview

2

2 Measurement Modelling Overview

Section Learning Objectives

Define measurement modelling

List the two elements of a structural equation model

Code a confirmatory factor analysis using *lavaan* in R

Interpret loadings, variances, and fit indices for CFA

What are Measurement Models?

- In educational and psychological research, we often work with latent constructs
  - Theoretical entities that account for characteristics or behaviors (Bandalos, 2018)
  - e.g., motivation, depression, anxiety
- We create items (prompts, questions) to capture elements of these constructs so we can theorize about them
- Measurement modelling is the process of relating these items to the latent variables they (supposedly) capture
- Any kind of relationship between an item and a latent variable is a measurement model
  - Sum scores
  - Single items
  - Factor analysis

Bandalos, D. L. (2018). *Measurement theory and applications for the social sciences.* New York, NY, US: The Guilford Press.

Structural Equation Modelling

- Combination of a type of measurement modelling (e.g., CFA) and analyzing relationships between variables of interest (path analysis)
- Three inputs, three outputs

In:

1. Theory of causal relationships between variables
2. Specification of relationships between variables of interest
3. Data

Out:

1. Numeric estimates of model parameters for hypothesized relationships between variables
2. Implications of the model not directly specified (e.g., W and Y unrelated, controlling for Z)
3. Fit indices reflecting degree to which testable implications are supported by data

Kline, R. B. (2015). *Principles and practice of structural equation modeling*, Fourth Edition: Guilford Publications.

Example CFA

- We'll now look at an example of a confirmatory factor analysis, a very common and useful measurement model
- First, we'll look at the data structure and discuss the models we'll run
- Then, we'll look at two different (equivalent) ways to identify the model and code it
- Finally, we'll look at output and interpret loadings, variances, and fit indices

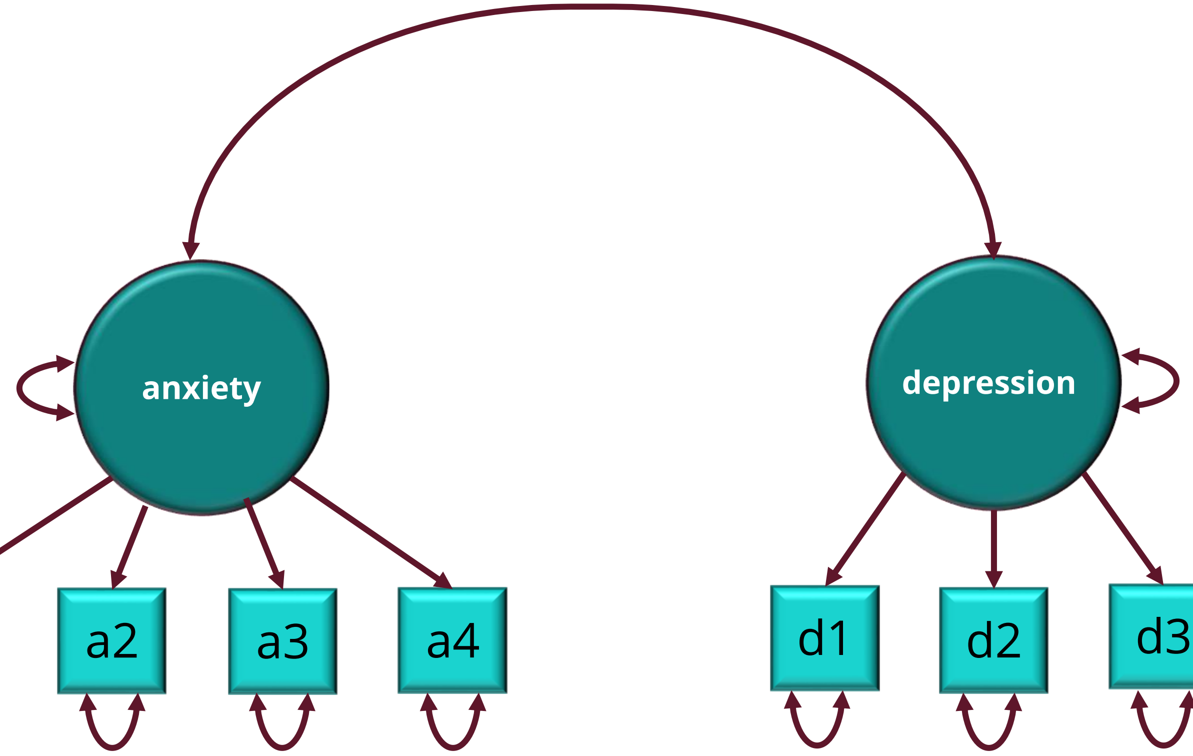
Example CFA: Data Structure

- Simulated data
- Responses range from 1 to 5 for questions about anxiety and depression symptoms

a1	a2	a3	a4	d1	d2	d3
1	2	1	1	3	2	2
1	2	1	2	2	1	2
3	1	1	1	1	1	2
3	1	1	4	2	2	1
1	2	1	2	3	1	3
1	1	2	2	2	2	1
5	2	2	2	2	1	2
2	1	1	1	1	1	2

Example CFA: Model Specification

- Two-factor model: anxiety and depression
- Four items assessing anxiety: a1 - a4
- Three items assessing depression: d1 - d3



Example CFA: Code

```
# 0 Load dependencies and data ------------------------------------------

library(lavaan)
data <- read.csv("sim_cfa.csv")

# 1 CFA -------------------------------------------------------------------

model <- '
          # Item loadings
          ANX =~ NA*a1 + a2 + a3 + a4
          DEP =~ NA*d1 + d2 + d3

          # Factor co/variances
          ANX ~~ 1*ANX
          DEP ~~ 1*DEP
          ANX ~~ DEP
'

model_cfa <- cfa(model, data)

summary(model_cfa, fit.measures = TRUE, standardized = TRUE)
```

Example CFA: Loadings and Variances

Latent Variables:

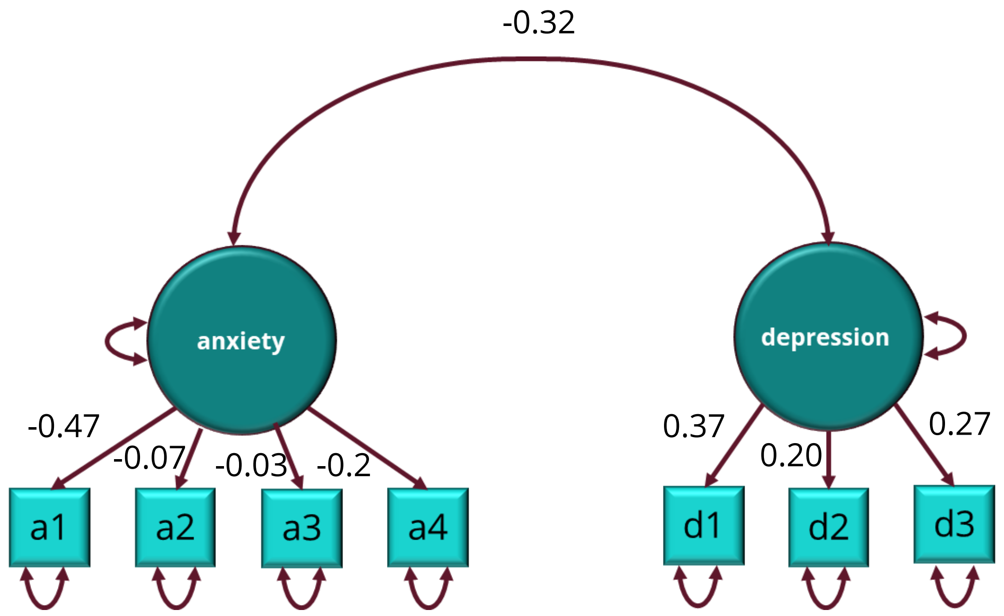
	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
ANX =~						
a1	-0.467	0.140	-3.345	0.001	-0.467	-0.549
a2	-0.073	0.040	-1.800	0.072	-0.073	-0.112
a3	-0.026	0.038	-0.691	0.489	-0.026	-0.041
a4	-0.200	0.064	-3.116	0.002	-0.200	-0.287
DEP =~						
d1	0.370	0.072	5.106	0.000	0.370	0.454
d2	0.195	0.047	4.191	0.000	0.195	0.273
d3	0.274	0.057	4.839	0.000	0.274	0.371

Covariances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
ANX ~~						
DEP	-0.324	0.120	-2.693	0.007	-0.324	-0.324

Variances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
ANX	1.000				1.000	1.000
DEP	1.000				1.000	1.000
.a1	0.507	0.130	3.888	0.000	0.507	0.699
.a2	0.418	0.021	19.489	0.000	0.418	0.988
.a3	0.415	0.021	19.940	0.000	0.415	0.998
.a4	0.445	0.033	13.680	0.000	0.445	0.918
.d1	0.528	0.056	9.393	0.000	0.528	0.794
.d2	0.474	0.028	16.779	0.000	0.474	0.926
.d3	0.469	0.036	12.903	0.000	0.469	0.862



Example CFA: Fit Indices

```
Model Test User Model:

  Test statistic                                10.084
  Degrees of freedom                                13
  P-value (Chi-square)                           0.687

Model Test Baseline Model:

  Test statistic                                84.604
  Degrees of freedom                                21
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    1.000
  Tucker-Lewis Index (TLI)                       1.074
```

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -6121.951
  Loglikelihood unrestricted model (H1)      -6116.909

  Akaike (AIC)                               12273.902
  Bayesian (BIC)                             12344.171
  Sample-size adjusted Bayesian (BIC)        12296.538

Root Mean Square Error of Approximation:

  RMSEA                                          0.000
  90 Percent confidence interval - lower         0.000
  90 Percent confidence interval - upper         0.028
  P-value RMSEA <= 0.05                          1.000

Standardized Root Mean Square Residual:

  SRMR                                           0.021
```

McNeish & Wolf (2021)
https://dynamicfit.app/__landing__/

Example CFA: Code

```
# Unit loading identification

model <- '
          # Item loadings
          ANX =~ a1 + a2 + a3 + a4
          DEP =~ d1 + d2 + d3

          # Factor co/variances
          ANX ~~ ANX
          DEP ~~ DEP
          ANX ~~ DEP
'

model_cfa <- cfa(model, data)

summary(model_cfa, fit.measures = TRUE, standardized = TRUE)
```

Example CFA: Loadings and Variances

Latent Variables:

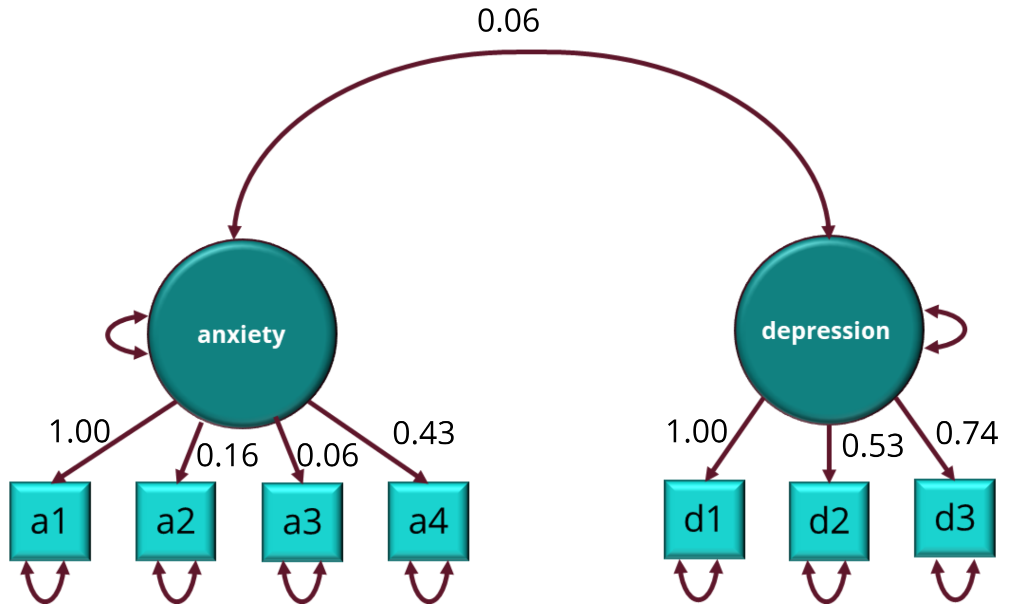
	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
ANX =~						
a1	1.000				0.467	0.549
a2	0.155	0.110	1.416	0.157	0.073	0.112
a3	0.056	0.085	0.662	0.508	0.026	0.041
a4	0.428	0.247	1.732	0.083	0.200	0.287
DEP =~						
d1	1.000				0.370	0.454
d2	0.528	0.180	2.932	0.003	0.195	0.273
d3	0.741	0.255	2.906	0.004	0.274	0.371

Covariances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
ANX ~~						
DEP	0.056	0.021	2.691	0.007	0.324	0.324

Variances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
ANX	0.218	0.130	1.673	0.094	1.000	1.000
DEP	0.137	0.054	2.553	0.011	1.000	1.000
.a1	0.507	0.130	3.888	0.000	0.507	0.699
.a2	0.418	0.021	19.489	0.000	0.418	0.988
.a3	0.415	0.021	19.940	0.000	0.415	0.998
.a4	0.445	0.033	13.680	0.000	0.445	0.918
.d1	0.528	0.056	9.393	0.000	0.528	0.794
.d2	0.474	0.028	16.779	0.000	0.474	0.926
.d3	0.469	0.036	12.903	0.000	0.469	0.862



From CFA to SEM

- Once you have your measurement model sorted (in our case, the CFA), you can move on to SEM
- You can add more variables and use the latent variables from the measurement model as predictors or outcomes for other latent variables or observed variables
- Our focus is on the measurement model, so we won't cover SEM further

Summary

- Measurement modelling is the process of relating items to latent constructs
- Confirmatory factor analysis is one kind of measurement model
- Structural equation modelling combines measurement modelling (e.g., CFA) with structural modelling of relationships between latent variables
- We implemented a confirmatory factor analysis using lavaan in R

Multilevel Measurement Modelling

3

Section Learning Objectives

3 Multilevel Measurement Modelling

Define a multilevel measurement model (MLMM)

Understand when to use an MLMM

Describe two different approaches to multilevel measurement modelling

State two issues that arise from not using an MLMM

What is a Multilevel Measurement Model?

So far, we've introduced and reviewed two modelling frameworks

1. Multilevel modelling for clustered data
2. Measurement modelling in single-level data

We use multilevel measurement models (MLMMs) when these circumstances collide, i.e. when:

- We want to conduct measurement modelling (linking items to factors)
- In clustered data structures

Why use an MLMM?

MLMMs are a mix of measurement modelling and clustered data modelling

If you don't use one, you run into two main issues:

No Measurement Modelling

no accounting for measurement error

No Multilevel Modelling

no accounting for clustered data

Dyer, N., Hanges, P., & Hall, R. (2005). Applying multilevel confirmatory factor analysis techniques to the study of leadership. *The Leadership Quarterly, 16*, 149-167. doi:10.1016/j.leaqua.2004.09.009

Huang, F. (2017). *Conducting Multilevel Confirmatory Factor Analysis Using R.*

Muthén, B. O. (1994). Multilevel Covariance Structure Analysis. *Sociological Methods & Research, 22*(3), 376-398. doi:10.1177/0049124194022003006

No Measurement Modelling

- Assumes perfect measurement
- In our MLM example, we used hours studied and teacher salary, which could plausibly have no measurement error
- In our SEM example, we talked about anxiety and depression which are more clearly latent constructs

Why use an MLMM?

MLMMs are a mix of measurement modelling and clustered data modelling

If you don't use one, you run into two main issues:

No Measurement Modelling

no accounting for measurement error

No Multilevel Modelling

no accounting for clustered data

Dyer, N., Hanges, P., & Hall, R. (2005). Applying multilevel confirmatory factor analysis techniques to the study of leadership. *The Leadership Quarterly, 16*, 149-167. doi:10.1016/j.leaqua.2004.09.009

Huang, F. (2017). *Conducting Multilevel Confirmatory Factor Analysis Using R.*

Muthén, B. O. (1994). Multilevel Covariance Structure Analysis. *Sociological Methods & Research, 22*(3), 376-398. doi:10.1177/0049124194022003006

No Multilevel Modelling

- Without MLM, we don't account for clustered data
- Violates assumption of independence
- Can bias parameter estimates, standard errors, and model fit
- Denies ability to ask multilevel questions
  - Some constructs have different factor structures across levels
  - Some constructs only exist at the group level, e.g., “classroom environment”

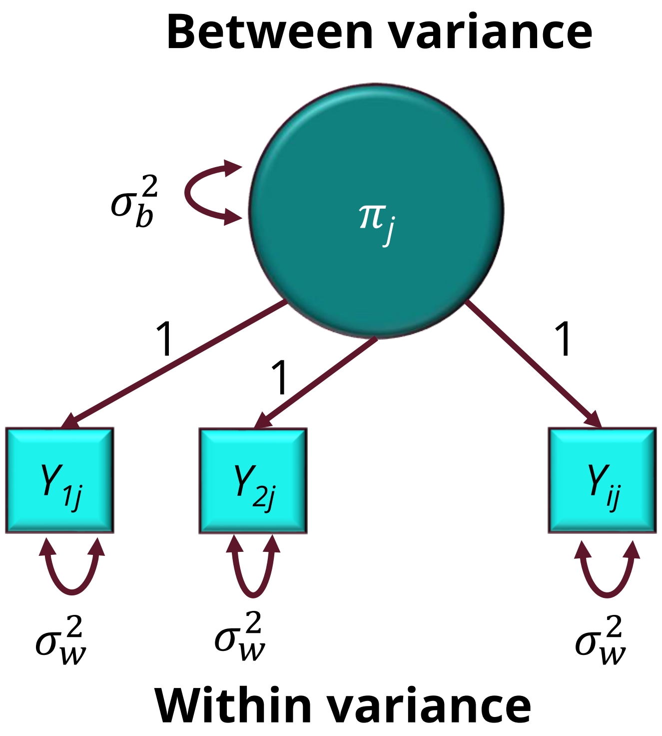
MLM and SEM: Analogous Frameworks

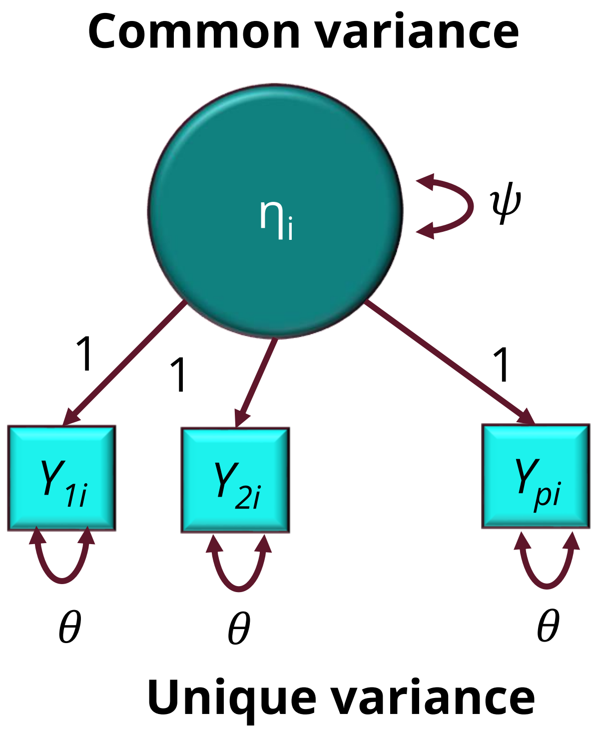
MLM

$$Y_{ij} = \beta + U_j + \epsilon_{ij}$$

SEM

$$Y_{pi} = v_p + \lambda_{pq}\eta_{qi} + \epsilon_{pi}$$



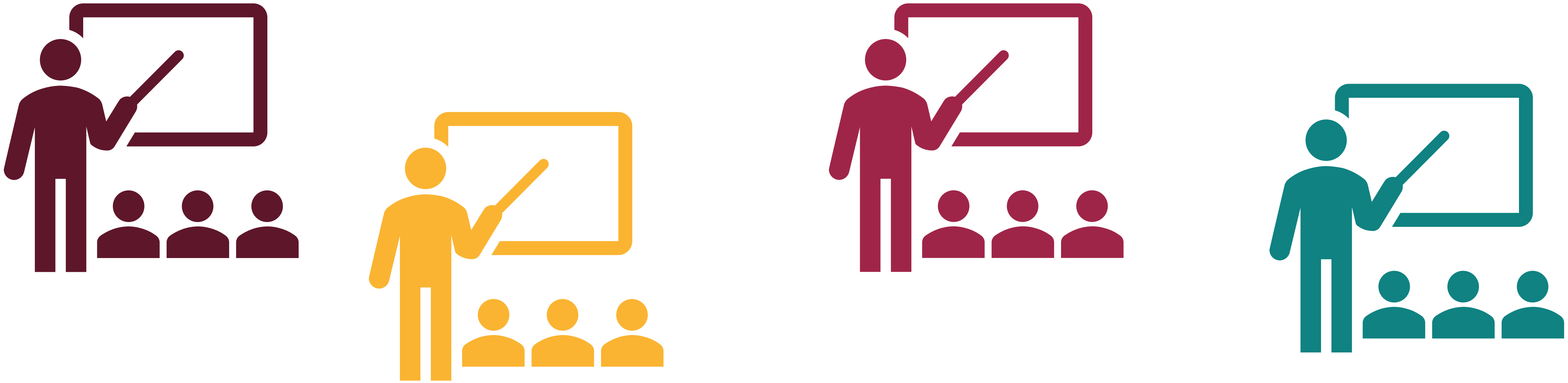


Mehta & Neale (2005)

Two Frameworks: MLM or SEM

We can adapt each of the two frameworks reviewed to MLMM

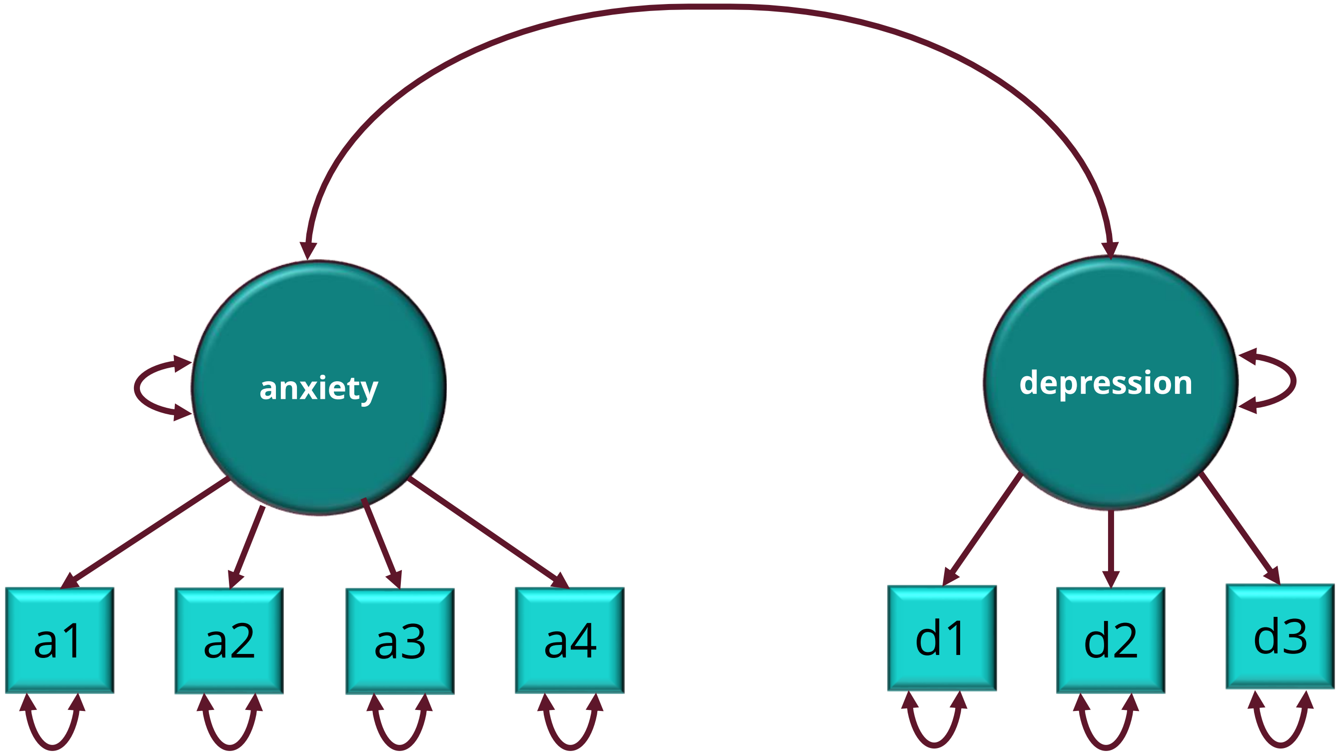
MLM: Use multilevel modelling for measurement modelling



Two Frameworks: MLM or SEM

We can adapt each of the two frameworks reviewed to MLMM

SEM: Add multilevel modelling to measurement modelling



Unclustered Factor Model: MLM vs SEM

- Let's run a factor model in each framework assuming no clustering, to compare and set the stage for adding another level
- We're not getting into data preparation (that's in the next section) or model details, just looking at code and output

Example: Teacher perception of student engagement

- One factor
- Six items
  - e.g., "Students generally like this school."
- Data from:

Huang F.L. & Cornell D.G. (2015). Factor structure of the high school teacher version of the authoritative school climate survey. *Journal of Psychoeducational Assessment*, 1557-5144. doi:10.1177/0734282915621439.

Unclustered Factor Model: MLM vs SEM

MLM data structure

	response	x1	x2	x3	x4	x5	x6	teacherID
1	4	1	0	0	0	0	0	1
2	4	0	1	0	0	0	0	1
3	3	0	0	1	0	0	0	1
4	4	0	0	0	1	0	0	1
5	4	0	0	0	0	1	0	1
6	3	0	0	0	0	0	1	1
7	6	1	0	0	0	0	0	2
8	6	0	1	0	0	0	0	2

```r
data_t <- data %>%
  rename(schoolID = sid) %>%
  mutate(teacherID = 1:nrow(data)) %>% # add row for teacher ID
  pivot_longer(
    cols = starts_with("x"),
    names_to = "item",
    names_prefix = "x",
    values_to = "response"
  ) %>%
  mutate(
    x1 = ifelse(item == 1, 1, 0),
    x2 = ifelse(item == 2, 1, 0),
    x3 = ifelse(item == 3, 1, 0),
    x4 = ifelse(item == 4, 1, 0),
    x5 = ifelse(item == 5, 1, 0),
    x6 = ifelse(item == 6, 1, 0)
  ) %>%
  select(response, x1:x6, item, teacherID, schoolID)
```

SEM data structure

	x1	x2	x3	x4	x5	x6	sid
1	4	4	3	4	4	3	55
2	6	6	4	6	4	4	55
3	5	5	2	6	4	4	55
4	4	4	5	5	3	3	55
5	4	4	4	4	3	2	55
6	4	4	2	4	2	2	55
7	5	4	3	5	3	3	41
8	3	2	2	3	3	2	41

Unclustered Factor Model: MLM vs SEM

MLM code: lme4

```
mlm <- lmer(response ~ 1 + x2 + x3 + x4 + x5 + x6 + (1|teacherID),
            data = data_t,
            REML = TRUE)
summary(mlm)
```

SEM code

```
sem <- '

      engagement =~ NA*x1 + x2 + x3 + x4 + x5 + x6

      engagement ~~ 1*engagement

'
sem_fit <- cfa(model = sem, data = data, meanstructure = TRUE)
summary(sem_fit,
        fit.measures = TRUE,
        standardized = TRUE)
```

Unclustered Factor Model: MLM vs SEM

MLM results: lme4

```
Random effects:
 Groups    Name        Variance Std.Dev.
 teacherID (Intercept) 0.5620   0.7497
 Residual              0.5496   0.7414
Number of obs: 23364, groups:  teacherID, 3894

Fixed effects:
            Estimate Std. Error t value
(Intercept)  4.7774     0.0169  282.75
x2          -0.1721     0.0168  -10.24
x3          -1.2360     0.0168  -73.56
x4          -0.3983     0.0168  -23.71
x5          -0.5354     0.0168  -31.87
x6          -0.8318     0.0168  -49.51
```

SEM results

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  engagement =~
    x1                0.736    0.012   62.120    0.000    0.736    0.840
    x2                0.840    0.013   63.372    0.000    0.840    0.851
    x3                0.700    0.018   39.239    0.000    0.700    0.599
    x4                0.550    0.016   33.425    0.000    0.550    0.524
    x5                0.804    0.016   50.571    0.000    0.804    0.728
    x6                0.821    0.016   51.505    0.000    0.821    0.738

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                4.777    0.014  340.099    0.000    4.777    5.450
   .x2                4.605    0.016  291.129    0.000    4.605    4.665
   .x3                3.541    0.019  189.049    0.000    3.541    3.030
   .x4                4.379    0.017  260.388    0.000    4.379    4.173
   .x5                4.242    0.018  239.665    0.000    4.242    3.841
   .x6                3.946    0.018  221.289    0.000    3.946    3.546
    engagement        0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    engagement        1.000                               1.000    1.000
   .x1                0.226    0.007   30.699    0.000    0.226    0.294
   .x2                0.268    0.009   29.443    0.000    0.268    0.275
   .x3                0.876    0.021   41.111    0.000    0.876    0.641
   .x4                0.799    0.019   42.095    0.000    0.799    0.725
   .x5                0.573    0.015   37.956    0.000    0.573    0.469
   .x6                0.564    0.015   37.582    0.000    0.564    0.455
```

MLM: nlme approach

SEM code

```
sem <- '

      engagement =~ NA*x1 + x2 + x3 + x4 + x5 + x6

      engagement ~~ 1*engagement

'
sem_fit <- cfa(model = sem, data = data, meanstructure = TRUE)
summary(sem_fit,
        fit.measures = TRUE,
        standardized = TRUE)
```

MLM code: nlme

```
mlm_nlme <- lme(
  response ~ 0 + x1 + x2 + x3 + x4 + x5 + x6,
  data = data_t,
  random = ~ 1 | teacherID,
  weights = varIdent(form = ~ 1 | item)
)
summary(mlm_nlme)

# level-1 variance components
(c(1.000, coef(mlm_nlme$modelStruct$varStruct, unconstrained = FALSE))*mlm_nlme$sigma)^2

# level-2 error variance (factor variance)
.7604**2
```

Unclustered Factor Model: MLM vs SEM

MLM Results: nlme

```
Random effects:
 Formula: ~1 | teacherID
        (Intercept)  Residual
StdDev:   0.7603665 0.4741239

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | item
 Parameter estimates:
       1        2        3        4        5        6
1.000000 1.144862 1.967339 1.893319 1.607276 1.600786
Fixed effects: response ~ 0 + x1 + x2 + x3 + x4 + x5 + x6
      Value  Std.Error    DF  t-value p-value
x1 4.777350 0.01435974 19465 332.6907       0
x2 4.605290 0.01497126 19465 307.6087       0
x3 3.541346 0.01928487 19465 183.6334       0
x4 4.379045 0.01885230 19465 232.2817       0
x5 4.241911 0.01725123 19465 245.8903       0
x6 3.945557 0.01721636 19465 229.1749       0
```

Level 1 error variances rescaled:

```
                  2         3         4         5         6
0.2247934 0.2946390 0.8700454 0.8058071 0.5807167 0.5760366
```

SEM results

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  engagement =~
    x1                0.736    0.012   62.120    0.000    0.736    0.840
    x2                0.840    0.013   63.372    0.000    0.840    0.851
    x3                0.700    0.018   39.239    0.000    0.700    0.599
    x4                0.550    0.016   33.425    0.000    0.550    0.524
    x5                0.804    0.016   50.571    0.000    0.804    0.728
    x6                0.821    0.016   51.505    0.000    0.821    0.738

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                4.777    0.014  340.099    0.000    4.777    5.450
   .x2                4.605    0.016  291.129    0.000    4.605    4.665
   .x3                3.541    0.019  189.049    0.000    3.541    3.030
   .x4                4.379    0.017  260.388    0.000    4.379    4.173
   .x5                4.242    0.018  239.665    0.000    4.242    3.841
   .x6                3.946    0.018  221.289    0.000    3.946    3.546
    engagement        0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    engagement        1.000                               1.000    1.000
   .x1                0.226    0.007   30.699    0.000    0.226    0.294
   .x2                0.268    0.009   29.443    0.000    0.268    0.275
   .x3                0.876    0.021   41.111    0.000    0.876    0.641
   .x4                0.799    0.019   42.095    0.000    0.799    0.725
   .x5                0.573    0.015   37.956    0.000    0.573    0.469
   .x6                0.564    0.015   37.582    0.000    0.564    0.455
```

MLM: nlme approach

SEM code

```
# SEM with loadings constrained to 1
sem <- '

      engagement =~ 1*x1 + 1*x2 + 1*x3 + 1*x4 + 1*x5 + 1*x6

      engagement ~~ engagement

'
sem_fit <- cfa(model = sem, data = data, meanstructure = TRUE)
summary(sem_fit,
        fit.measures = TRUE,
        standardized = TRUE)
```

MLM code: nlme

```
mlm_nlme <- lme(
  response ~ 0 + x1 + x2 + x3 + x4 + x5 + x6,
  data = data_t,
  random = ~ 1 | teacherID,
  weights = varIdent(form = ~ 1 | item)
)
summary(mlm_nlme)

# level-1 variance components
(c(1.000, coef(mlm_nlme$modelStruct$varStruct, unconstrained = FALSE))*mlm_nlme$sigma)^2

# level-2 error variance (factor variance)
.7604**2
```

Latent Variable Variance

MLM Results: nlme

```
Random effects:
 Formula: ~1 | teacherID
        (Intercept)  Residual
StdDev:   0.7603665 0.4741239

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | item
 Parameter estimates:
       1        2        3        4        5        6
1.000000 1.144862 1.967339 1.893319 1.607276 1.600786
Fixed effects: response ~ 0 + x1 + x2 + x3 + x4 + x5 + x6
      Value  Std.Error    DF  t-value p-value
x1 4.777350 0.01435974 19465 332.6907       0
x2 4.605290 0.01497126 19465 307.6087       0
x3 3.541346 0.01928487 19465 183.6334       0
x4 4.379045 0.01885230 19465 232.2817       0
x5 4.241911 0.01725123 19465 245.8903       0
x6 3.945557 0.01721636 19465 229.1749       0
```

Level 2 error variance:

SEM results

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  engagement =~
    x1                1.000                               0.760    0.849
    x2                1.000                               0.760    0.814
    x3                1.000                               0.760    0.632
    x4                1.000                               0.760    0.646
    x5                1.000                               0.760    0.706
    x6                1.000                               0.760    0.708

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                4.777    0.014  332.734    0.000    4.777    5.332
   .x2                4.605    0.015  307.647    0.000    4.605    4.930
   .x3                3.541    0.019  183.655    0.000    3.541    2.943
   .x4                4.379    0.019  232.311    0.000    4.379    3.723
   .x5                4.242    0.017  245.920    0.000    4.242    3.941
   .x6                3.946    0.017  229.202    0.000    3.946    3.673
    engagement        0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    engagement        0.578    0.015   39.098    0.000    1.000    1.000
   .x1                0.225    0.007   31.058    0.000    0.225    0.280
   .x2                0.295    0.009   34.130    0.000    0.295    0.338
   .x3                0.870    0.021   40.802    0.000    0.870    0.601
   .x4                0.806    0.020   40.535    0.000    0.806    0.582
   .x5                0.581    0.015   39.121    0.000    0.581    0.501
   .x6                0.576    0.015   39.079    0.000    0.576    0.499
```

Summary

- A multilevel measurement model (MLMM) is used to do measurement modelling in clustered data
- We outlined two approaches: (1) using an MLM framework and (2) using an SEM framework
- We ran an unclustered (i.e., single-level) measurement model in each framework to demonstrate the analogy and preview the similarities and differences

Multilevel Measurement Models in a Multilevel Modelling Framework

4

Section Learning Objectives

4 MLMMs in MLM Framework

State the 5-step process for conducting an MLMM in MLM framework

Restructure data to be used with MLM framework

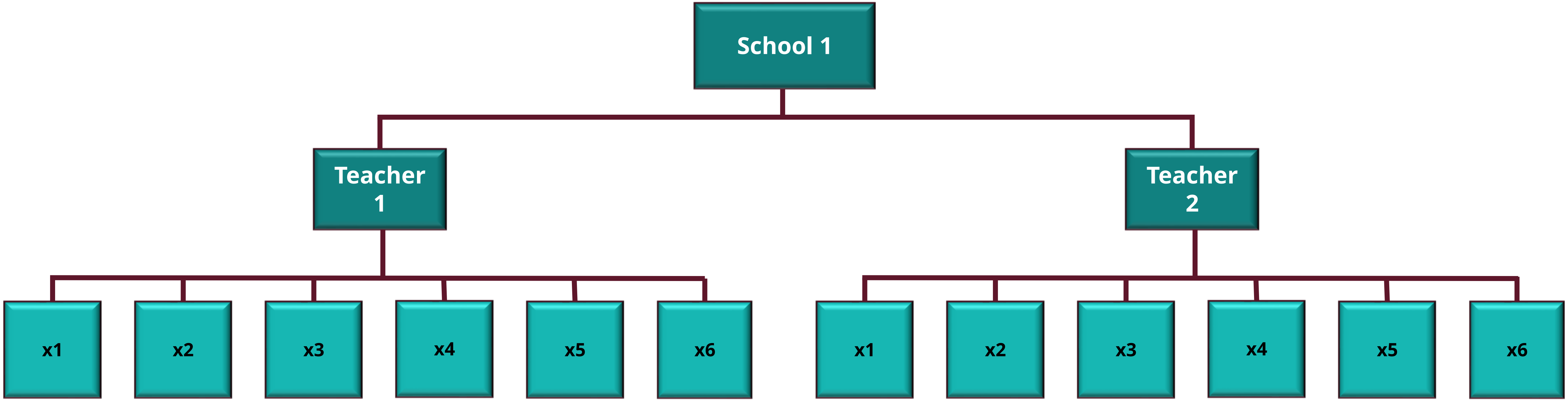
List and interpret equations for multilevel measurement models in MLM framework

Execute multilevel measurement model in R package *nlme* and interpret output

Our Example

- Data structure: teachers within schools
  - Between 5 and 50 teachers per school
  - 254 schools
- Outcome: perception of student engagement
- Outcome variance can be decomposed into three parts
  - Within teachers: variance across item responses
  - Between teachers: variance across teachers within schools
  - Between schools: variance across schools
- 6 items assessing student engagement
  - 1 = strongly disagree, 2 = disagree, 3 = somewhat disagree, 4 = somewhat agree, 5 = agree, 6 = strongly agree
- Random subset of data from Huang and Cornell (2015)
  - https://github.com/flh3/pubdata/blob/main/MLCFA/raw.csv

Our Example



- x1: Students generally like this school.
- x2: Students are proud to be at this school.
- x3: Students finish their homework at this school.
- x4: Students hate going to school. (reverse-coded)
- x5: Getting good grades is very important to most students here.
- x6: Most students want to learn as much as they can at this school.

MLMM: Equations

- **Outcome:** response to item
- **Level 1 equation:** $\mathrm{Y_{ijk}} = \pi_{0jk} + \pi_{1jk}X1_{ijk} + \cdots + \pi_{5jk}X6_{ijk} + \epsilon_{ijk}$
- **Level 2 equations:** $\pi_{0jk} = \beta_{00k} + r_{0jk}$

$$\pi_{1jk} = \beta_{10k}$$

$$\ldots$$

$$\pi_{5jk} = \beta_{50k}$$

- **Level 3 equation:** $\beta_{00k} = \gamma_{000} + U_{00k}$

$$\beta_{10k} = \gamma_{100}$$

$$\ldots$$

$$\beta_{50k} = \gamma_{500}$$

- **Total:** $\mathrm{Y_{ijk}} = \gamma_{100}X1_{ijk} + \cdots + \gamma_{500}X6_{ijk} + U_{00k} + r_{0jk} + \epsilon_{ijk}$
  - γ_{000} set to zero

MLMM: What the Model Means

- We are specifying a measurement model for the items, providing a student engagement factor score for each teacher and for each school
- Model results:
  - Intercept for each item that we assume holds across all teachers and schools (testable assumption)
  - Variance across teachers in student engagement factor scores
  - Variance across schools in student engagement factor scores
  - Factor loadings for each item are assumed to be 1 across teachers and schools

Bauer, D.J. (2003). Estimating multilevel linear models as structural equation models. *Journal of Educational and Behavioral Statistics, 28,* 135-167. doi:10.3102/10769986028002135

Steps for Running MLMM

Step 1	Load data
Step 2	Prepare data for MLMM
Step 3	Check extent of clustering
Step 4	MLMM
Step 5	Interpret output

Step 1 Load data

```
library(RCurl) # for extracting data from web
library(dplyr) # for data manipulation
library(tidyr) # for pivoting data
library(lme4) # for MLMs
library(performance) # for ICCs

data <- read.csv(text = getURL("https://raw.githubusercontent.com/flh3/pubdata/main/MLCFA/raw.csv"))
```

Step 2 Prepare data for MLMM

- Our structure is responses to items, clustered within teachers, clustered within schools
- We need long data when working with repeated measures
- That means transposing our dataset
- Our model outcome will be the response to a given item
- Each row is one item, with dummy coded variables

Step 2 Prepare data for MLMM

Before

	x1	x2	x3	x4	x5	x6	schoolID
1	4	4	3	4	4	3	55
2	6	6	4	6	4	4	55
3	5	5	2	6	4	4	55
4	4	4	5	5	3	3	55
5	4	4	4	4	3	2	55
6	4	4	2	4	2	2	55
7	5	4	3	5	3	3	41
8	3	2	2	3	3	2	41
9	4	4	2	4	1	1	41
10	4	4	2	5	2	2	41

After

	response	x1	x2	x3	x4	x5	x6	teacherID	schoolID
1	4	1	0	0	0	0	0	1	55
2	4	0	1	0	0	0	0	1	55
3	3	0	0	1	0	0	0	1	55
4	4	0	0	0	1	0	0	1	55
5	4	0	0	0	0	1	0	1	55
6	3	0	0	0	0	0	1	1	55
7	6	1	0	0	0	0	0	2	55
8	6	0	1	0	0	0	0	2	55
9	4	0	0	1	0	0	0	2	55
10	6	0	0	0	1	0	0	2	55

Step 2 Prepare data for MLMM

Code

```
data_t <- data %>%
  rename(schoolID = sid) %>%
  mutate(teacherID = 1:nrow(data)) %>% # add row for teacher ID
  pivot_longer(
    cols = starts_with("x"),
    names_to = "item",
    names_prefix = "x",
    values_to = "response"
  ) %>%
  mutate(
    x1 = ifelse(item == 1, 1, 0),
    x2 = ifelse(item == 2, 1, 0),
    x3 = ifelse(item == 3, 1, 0),
    x4 = ifelse(item == 4, 1, 0),
    x5 = ifelse(item == 5, 1, 0),
    x6 = ifelse(item == 6, 1, 0)
  ) %>%
  select(response, x1:x6, item, teacherID, schoolID)
```

Step 3 Check extent of clustering

- We check the extent of clustering to ascertain whether we need to go through this multi-step process
- We do this with the intraclass correlation coefficient (ICC), a ratio of variance between clusters to total variance
- With three levels, you can consider variance explained by clustering at level 2:

$$ICC_{L2} = \frac{\tau_0^2}{\phi_0^2 + \tau_0^2 + \sigma^2}$$

- Or variance explained by clustering at level 3:

$$ICC_{L3} = \frac{\phi_0^2}{\phi_0^2 + \tau_0^2 + \sigma^2}$$

- Ranges from 0 to 1, with higher numbers indicating that a larger proportion of item response variance is between clusters
- If the data aren't very clustered (ICC below .05), may not be necessary to take the extra steps accounting for clustering

Step 3 Check degree of clustering

nlme: Unconditional ICCs

```
null_nlme <- lme(
  response ~ 1,
  data = data_t,
  method = "REML",
  random = ~ 1 | schoolID/teacherID/item,
  control = lmeControl(optimizer = "Nelder-Mead")
)
summary(null_nlme)
```

nlme: Conditional ICCs

```
# conditional
conditional_nlme <- lme(
  response ~ 0 + x1 + x2 + x3 + x4 + x5 + x6,
  data = data_t,
  method = "REML",
  random = ~ 1 | schoolID/teacherID/item,
  control = lmeControl(optimizer = "Nelder-Mead")
)
summary(conditional_nlme)
```

Step 3 Check degree of clustering

- nlme outputs standard deviations, so we need to square them to get variances
- item %in% teacherID %in% school ID indicates level-1 variance, σ^2
- teacherID %in% schoolID indicates level-2 variance, τ_0^2
- schoolID indicates level-3 variance, ϕ_0^2

Unconditional

```
Random effects:
 Formula: ~1 | schoolID
        (Intercept)
StdDev:   0.4075208

 Formula: ~1 | teacherID %in% schoolID
        (Intercept)
StdDev:     0.60208

 Formula: ~1 | item %in% teacherID %in% schoolID
        (Intercept)  Residual
StdDev:   0.8006477 0.3341296
```

Conditional

```
Random effects:
 Formula: ~1 | schoolID
        (Intercept)
StdDev:   0.4181061

 Formula: ~1 | teacherID %in% schoolID
        (Intercept)
StdDev:   0.6386338

 Formula: ~1 | item %in% teacherID %in% schoolID
        (Intercept)  Residual
StdDev:   0.3570088 0.3177386
```

Step 3 Check degree of clustering

- $ICC_{L2} = \frac{\tau_0^2}{\phi_0^2 + \tau_0^2 + \sigma^2}$
  - $Unconditional = \frac{0.60^2}{0.60^2 + 0.41^2 + 0.80^2} = 0.30$
  - $Conditional = \frac{0.64^2}{0.64^2 + 0.42^2 + 0.36^2} = 0.57$
- $ICC_{L3} = \frac{\phi_0^2}{\phi_0^2 + \tau_0^2 + \sigma^2}$
  - $Unconditional = \frac{0.41^2}{0.60^2 + 0.41^2 + 0.80^2} = 0.14$
  - $Conditional = \frac{0.42^2}{0.64^2 + 0.42^2 + 0.36^2} = 0.25$

Unconditional

```
Random effects:
 Formula: ~1 | schoolID
        (Intercept)
StdDev:   0.4075208

 Formula: ~1 | teacherID %in% schoolID
        (Intercept)
StdDev:     0.60208

 Formula: ~1 | item %in% teacherID %in% schoolID
        (Intercept)  Residual
StdDev:   0.8006477 0.3341296
```

Conditional

```
Random effects:
 Formula: ~1 | schoolID
        (Intercept)
StdDev:   0.4181061

 Formula: ~1 | teacherID %in% schoolID
        (Intercept)
StdDev:   0.6386338

 Formula: ~1 | item %in% teacherID %in% schoolID
        (Intercept)  Residual
StdDev:   0.3570088 0.3177386
```

Step 4 MLMM

Total: $Y_{ijk} = \gamma_{100}X1_{ijk} + \cdots + \gamma_{500}X6_{ijk} + U_{00k} + r_{0jk} + \epsilon_{ijk}$

- The MLM framework allows us to model the relationships between items and latent student engagement factor, and obtain standing of teachers and schools on this factor
- Partitions total variance into 3 levels: within-teacher, between-teacher, and between-schools
- Yields fixed effects for items (average responses to items across all teachers and schools) and random effects for U_{00k} (how school mean varies from grand mean), r_{0jk} (how teacher's mean response varies from school mean), and e_{ijk} (how teacher’s response to item deviates from their own mean response across items controlling for L1 predictors)

Step 4 MLMM

Total: $Y_{ijk} = \gamma_{100}X1_{ijk} + \cdots + \gamma_{500}X6_{ijk} + U_{00k} + r_{0jk} + \epsilon_{ijk}$

```
mlmm <- lme(
  response ~ 0 + x1 + x2 + x3 + x4 + x5 + x6,
  data = data_t,
  random = ~ 1 | schoolID/teacherID/item,
  weights = varIdent(form = ~ 1 | item)
)
summary(mlmm)
```

Step 5

Interpret output

Total: $Y_{ijk} = \gamma_{100}X1_{ijk} + \cdots + \gamma_{500}X6_{ijk} + U_{00k} + r_{0jk} + \epsilon_{ijk}$

```
Random effects:
 Formula: ~1 | schoolID
        (Intercept)
StdDev:   0.4181061

 Formula: ~1 | teacherID %in% schoolID
        (Intercept)
StdDev:   0.6386338

 Formula: ~1 | item %in% teacherID %in% schoolID
        (Intercept)  Residual
StdDev:   0.3570088 0.3177386

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | item
 Parameter estimates:
       1        2        3        4        5        6
1.000000 1.273703 2.708233 2.593207 2.100547 2.124616
Fixed effects: response ~ 0 + x1 + x2 + x3 + x4 + x5 + x6
      Value  Std.Error    DF  t-value p-value
x1 4.741659 0.02977883 19465 159.2292       0
x2 4.569599 0.03004852 19465 152.0740       0
x3 3.505655 0.03241930 19465 108.1348       0
x4 4.343354 0.03217453 19465 134.9935       0
x5 4.206220 0.03122896 19465 134.6897       0
x6 3.909866 0.03127114 19465 125.0311       0
```

Step 5 Interpret output

Total: $\mathrm{Y_{ijk}} = \gamma_{100} X1_{ijk} + \cdots + \gamma_{500} X6_{ijk} + U_{00k} + r_{0jk} + \epsilon_{ijk}$

```
> # level-1 variance components
> (c(1.000, coef(mlmm$modelStruct$varStruct, unconstrained = FALSE))*mlmm$sigma)^2
                  2         3         4         5         6
0.1009578 0.1637858 0.7404776 0.6789130 0.4454560 0.4557227
> # level-2 error variance (teacher-level student engagement factor variance)
> 0.638**2
[1] 0.407044
> # level-3 error variance (school-level student engagement factor variance)
> 0.418**2
[1] 0.174724
```

Summary

- We ran a multilevel measurement model in an MLM framework, allowing us to recover item means and factor variances simultaneously
- The MLM framework is useful for obtaining latent trait values at multiple levels, conducting uniform differential item functioning analyses, investigating whether item-level predictors vary across clusters and explain response patterns, etc.

Multilevel Measurement Models in an SEM Framework

5

5

MLMMs in SEM Framework

Section Learning Objectives

State the six-step process for conducting an MLMM in an SEM framework

Specify a multilevel CFA

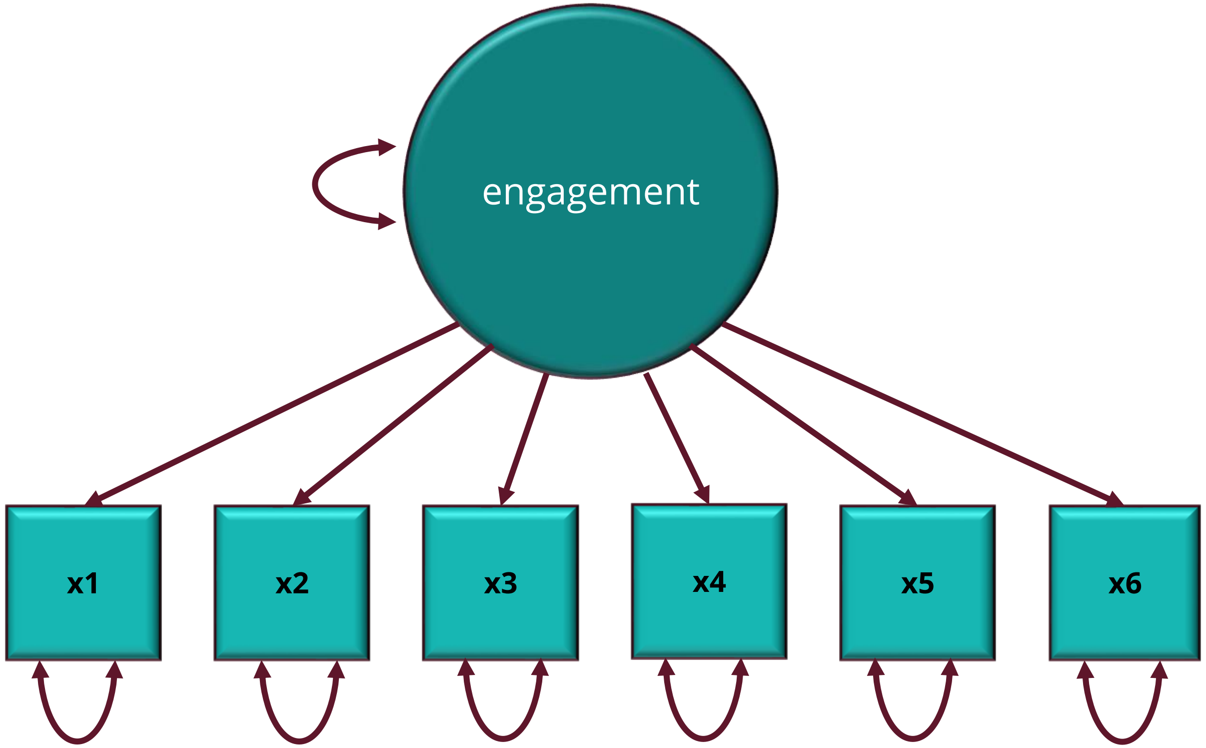
Execute a MLMM in an SEM framework using R package *lavaan* and interpret the output

Compare a single-level CFA to a multilevel CFA

Our Example

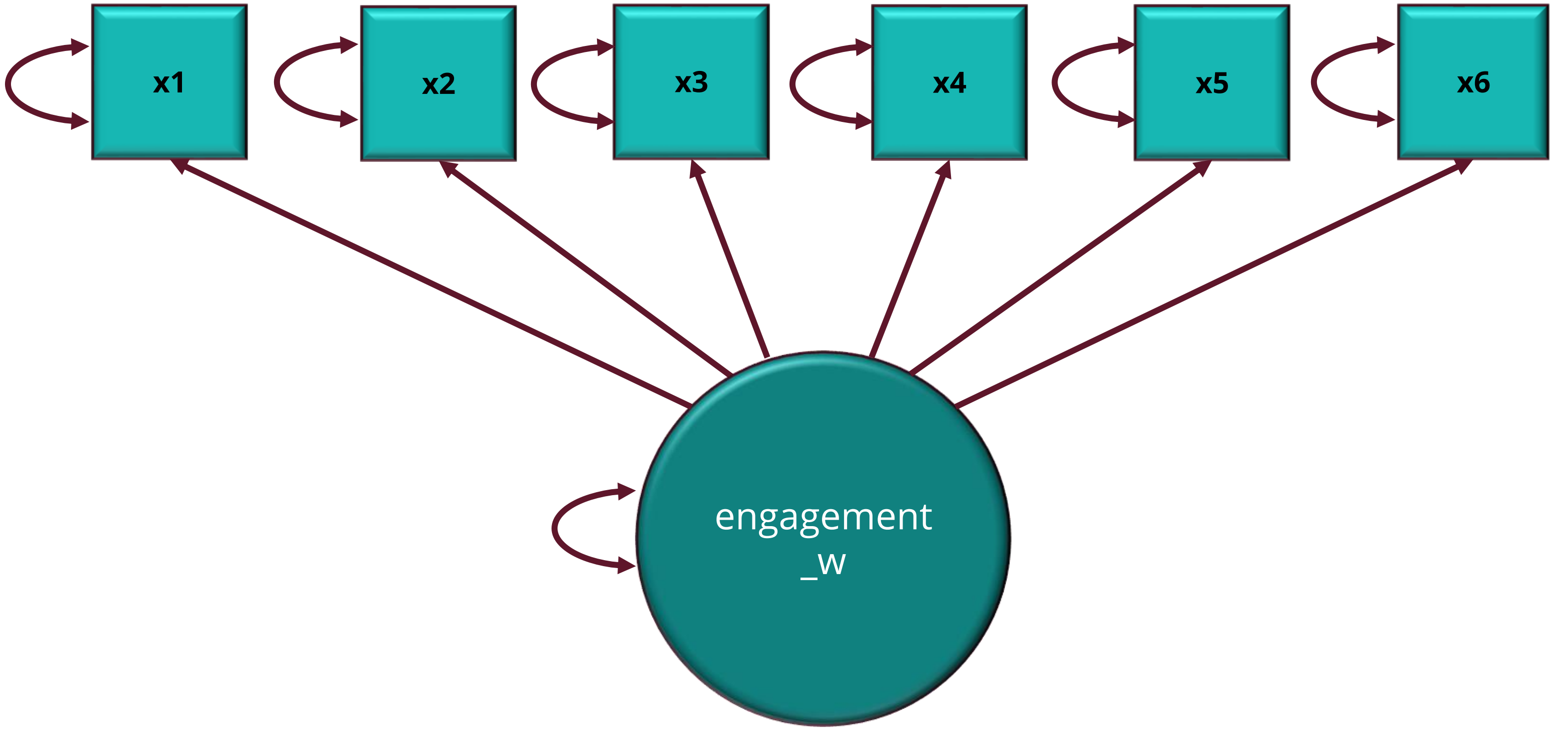
- Data structure: teachers within schools
  - Between 5 and 50 teachers per school
  - 254 schools
- Outcome: perception of student engagement
  - Within: teacher's perception of engagement
  - Between: school-level factor of general engagement
- 6 items assessing student engagement
  - 1 = strongly disagree, 2 = disagree, 3 = somewhat disagree, 4 = somewhat agree, 5 = agree, 6 = strongly agree
- Random subset of data from Huang and Cornell (2015)
  - https://github.com/flh3/pubdata/blob/main/MLCFA/raw.csv

MLMM in SEM Framework



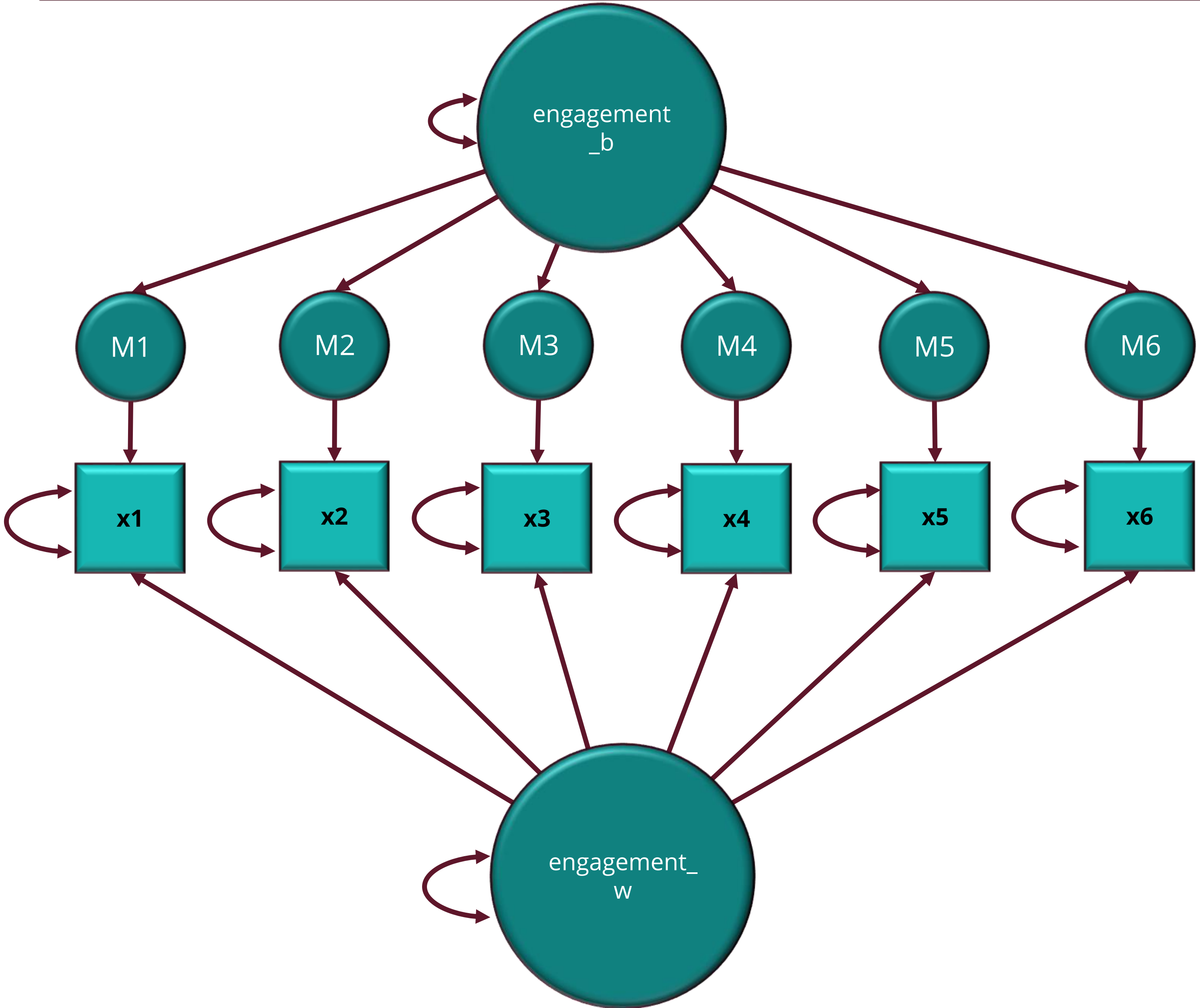
- A regular CFA (pictured) would tell us how our items relate to perceptions of student engagement via the loadings, accounting for measurement variance and factor variance
- This model assumes overall item variances and covariances, doesn't account for potential clustering
- What if our data are clustered?
  - Parameter estimates, standard errors, and model fit might be biased
  - Student engagement might have different meanings at individual and cluster level

MLMM in SEM Framework



- In an MLMM in the SEM framework, the “within” portion of the model looks just like a single-level CFA: items load directly to a within latent variable
- This "within" measurement model captures how the items reflect teacher perceptions of student engagement
- Higher loading = item related to perception of higher student engagement

MLMM in SEM Framework



- Then we add the “between” level, calculating group means that load onto a between latent variable
- The "between" measurement model captures how item means reflect school-level perception of student engagement
- Higher loading = item related to higher levels of school-level perception of engagement
- Within and between covariance matrices estimated simultaneously, partitions mixed total variance into within and between

Steps for Running

Step 1 Prepare Data

Step 2 Check degree of clustering

Step 3 Conventional CFA

Step 4 Within CFA

Step 5 Between CFA

Step 6 Multilevel CFA

- Steps 4 and 5 are helpful if you're having convergence issues, generate suggested starting points
- Full data required
  - i.e., not just variance-covariance matrix

Dyer (2005)
Kaplan (2017)
Muthén (1994)

Step 1 Prepare Data

```r
library(RCurl) # for extracting data from web
library(lavaan) # for SEM
library(dplyr) # renaming variables
library(lme4) # for ICCs
library(performance) # for ICCs

data <- read.csv(text = getURL("https://raw.githubusercontent.com/flh3/pubdata/main/MLCFA/raw.csv")) %>%
  rename(schoolID = sid)
```

Step 2 Check degree of clustering

- We check the extent of clustering to ascertain whether we need to go through this multi-step process
- If the data aren't very clustered, it might be unnecessary to take the extra steps accounting for clustering
- The intraclass correlation coefficient (ICC) from the multilevel modelling framework is useful here

$$ICC = \frac{\phi_0^2}{\phi_0^2 + \tau_0^2}$$

- With our items as the outcome, the ICC reflects how much variance in item responses is between cluster
- Ranges from 0 to 1, with higher numbers indicating that a larger proportion of item response variance is between clusters
- With ICC $< .05$, multilevel approach may not be needed

Dyer (2005)

Step 2 Check degree of clustering

```r
icc_x1 <- lmer(x1 ~ 1 + (1|schoolID), data = data, REML = F)
performance::icc(icc_x1)

icc_x2 <- lmer(x2 ~ 1 + (1|schoolID), data = data, REML = F)
performance::icc(icc_x2)

icc_x3 <- lmer(x3 ~ 1 + (1|schoolID), data = data, REML = F)
performance::icc(icc_x3)

icc_x4 <- lmer(x4 ~ 1 + (1|schoolID), data = data, REML = F)
performance::icc(icc_x4)

icc_x5 <- lmer(x5 ~ 1 + (1|schoolID), data = data, REML = F)
performance::icc(icc_x5)

icc_x6 <- lmer(x6 ~ 1 + (1|schoolID), data = data, REML = F)
performance::icc(icc_x6)
```

Output

- x1: 0.203
- x2: 0.258
- x3: 0.143
- x4: 0.109
- x5: 0.240
- x6: 0.127

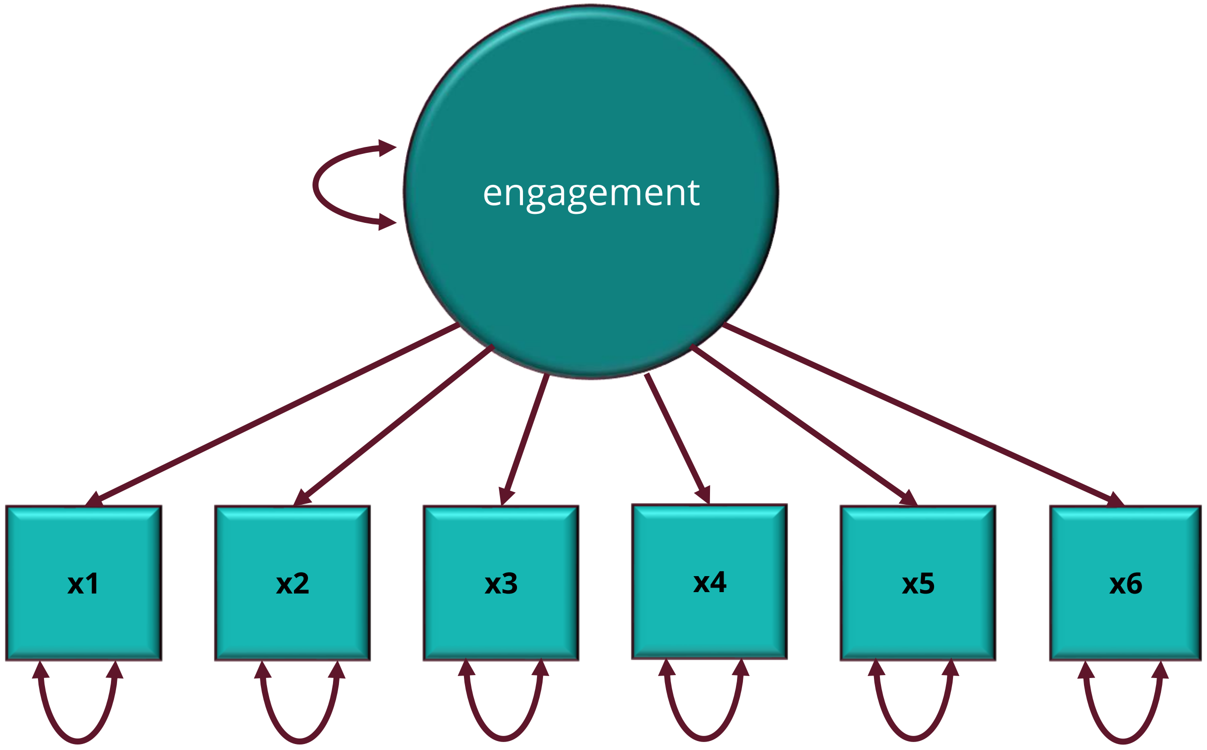
Step 3 Conventional CFA

- What is it: factor analysis of the total covariance matrix, which results in biased model fit and loading estimates because the total covariance matrix is an uninterpretable blend of within and between effects
- If the results are biased, why conduct this step?
- If your ICC is greater than .05, you don't need to conduct the conventional CFA, this model is more to illustrate the effect of accounting for the clustered structure
- If your ICC is less than .05, you might run a conventional CFA as normal in lieu of using the multilevel approach
  - With insufficient clustering, there is not a lot of mean level variance across clusters, so estimation might be difficult/impossible

Dyer (2005)

Step 3 Conventional CFA

```
regular <- '

      engagement =~ NA*x1 + x2 + x3 + x4 + x5 + x6
      engagement ~~ 1*engagement

'
regular_fit <- cfa(model = regular, data = data)
summary(regular_fit, fit.measures = TRUE, standardized = TRUE)
```



Step 3 Conventional CFA

```
Model Test User Model:

  Test statistic                              2226.990
  Degrees of freedom                                 9
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                             12323.196
  Degrees of freedom                                15
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.820
  Tucker-Lewis Index (TLI)                       0.700
```

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -29138.673
  Loglikelihood unrestricted model (H1)     -28025.178

  Akaike (AIC)                               58301.346
  Bayesian (BIC)                             58376.552
  Sample-size adjusted Bayesian (BIC)        58338.421

Root Mean Square Error of Approximation:

  RMSEA                                          0.252
  90 Percent confidence interval - lower         0.243
  90 Percent confidence interval - upper         0.260
  P-value RMSEA <= 0.05                          0.000

Standardized Root Mean Square Residual:

  SRMR                                           0.079
```

Step 3 Conventional CFA

Latent Variables:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
engagement =~						
x1	0.736	0.012	62.120	0.000	0.736	0.840
x2	0.840	0.013	63.372	0.000	0.840	0.851
x3	0.700	0.018	39.239	0.000	0.700	0.599
x4	0.550	0.016	33.425	0.000	0.550	0.524
x5	0.804	0.016	50.571	0.000	0.804	0.728
x6	0.821	0.016	51.505	0.000	0.821	0.738

Variances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
engagement	1.000				1.000	1.000
.x1	0.226	0.007	30.699	0.000	0.226	0.294
.x2	0.268	0.009	29.443	0.000	0.268	0.275
.x3	0.876	0.021	41.111	0.000	0.876	0.641
.x4	0.799	0.019	42.095	0.000	0.799	0.725
.x5	0.573	0.015	37.956	0.000	0.573	0.469
.x6	0.564	0.015	37.582	0.000	0.564	0.455

S_{PW} and S_B

- It is possible to run level-specific models
- To do so, you need to manually partition the total variance-covariance matrix into within and between matrices
- S_{PW} (pooled within) is an unbiased estimate of population within covariance matrix
- S_B is a biased estimator of the population between covariance matrix, but can be adjusted with information about average cluster size and S_{PW}

Huang (2017)
Muthén (1994)

S_{PW} and S_B: Equations

- We'll be using the function written by Huang (2017) to get our partitioned covariance matrices

  https://github.com/flh3/mcfa/blob/main/02_syntax/mcfa2.R

- Equations for your reference

$$\mathbf{S_{PW}} = (n - G)^{-1} \sum_{g=1}^{G} \sum_{i=1}^{n_g} (\mathbf{y_{ig}} - \bar{\mathbf{y}}_{\mathbf{g}})(\mathbf{y_{ig}} - \bar{\mathbf{y}}_{\mathbf{g}})'$$

$$\mathbf{S_B} = (G - 1)^{-1} \sum_{g=1}^{G} n_g (\mathbf{y_g} - \bar{\mathbf{y}})(\mathbf{y_g} - \bar{\mathbf{y}})'$$

$$c. = [n^2 - \sum_{g=1}^{G} n_g^2][n(G-1)]^{-1}$$

- $\Sigma_B = \frac{S_B - S_{PW}}{c}$

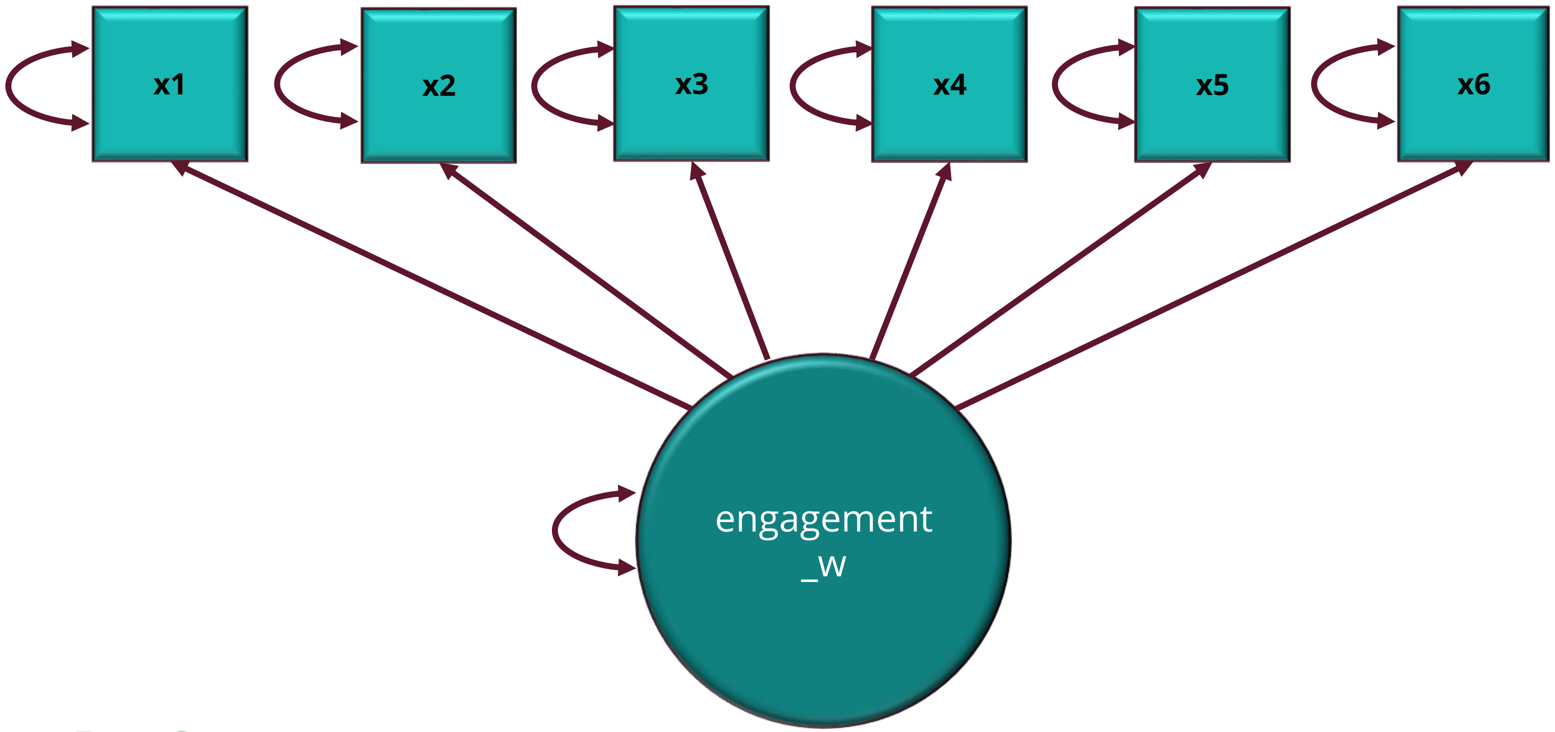
- n is total sample size
- G is number of groups
- y_{ig} is score of observation I nested in group g
- $\bar{y}_g$ is cluster mean in group g
- $\bar{y}$ is overall grand mean
- c is average cluster size

Huang (2017)
Muthén (1994)

Step 4 Within CFA

- What is it: Factor analysis of within-level using pooled within covariance matrix, S_{PW}
- Why do it: Evaluates the appropriateness of the within structure
- Yields fit information and factor loadings at the within-only level
- If fit is poor, may want to re-evaluate the within model (and document that you did so)
- If you want an unbiased within-only model (i.e., you want to account for clustering but aren't interested in the between model), you can use the output from this model
- Can be used to inform starting values for the MLMM if there are convergence issues (more on that later)

Step 4 Within CFA




```
within <- '

    engagement_w =~ NA*x1 + x2 + x3 + x4 + x5 + x6
    engagement_w ~~ 1*engagement_w

'
within_fit <- cfa(within, sample.cov = x$pw.cov, sample.nobs = x$n)
summary(within_fit, fit.measures = TRUE, standardized = TRUE)
```

Step 4 Within CFA

Model Test User Model:

Test statistic	2109.180
Degrees of freedom	9
P-value (Chi-square)	0.000

Model Test Baseline Model:

Test statistic	10000.088
Degrees of freedom	15
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.790
Tucker-Lewis Index (TLI)	0.649

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-27924.172
Loglikelihood unrestricted model (H1)	-26869.582
Akaike (AIC)	55872.344
Bayesian (BIC)	55947.550
Sample-size adjusted Bayesian (BIC)	55909.420

Root Mean Square Error of Approximation:

RMSEA	0.245
90 Percent confidence interval - lower	0.236
90 Percent confidence interval - upper	0.254
P-value RMSEA <= 0.05	0.000

Standardized Root Mean Square Residual:

SRMR	0.081

Step 4 Within CFA

Latent Variables:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
engagement_w =~						
x1	0.583	0.011	50.692	0.000	0.583	0.743
x2	0.642	0.013	51.300	0.000	0.642	0.749
x3	0.633	0.017	37.314	0.000	0.633	0.585
x4	0.450	0.016	27.759	0.000	0.450	0.454
x5	0.710	0.014	50.542	0.000	0.710	0.741
x6	0.800	0.015	53.102	0.000	0.800	0.768

Variances:

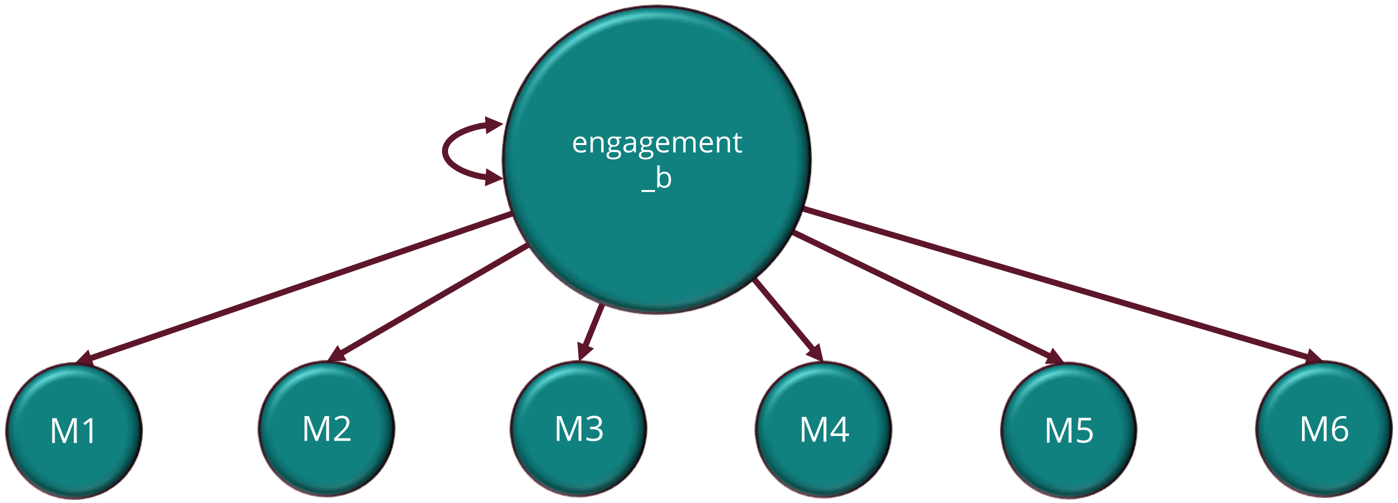
	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
engagement_w	1.000				1.000	1.000
.x1	0.276	0.008	34.740	0.000	0.276	0.448
.x2	0.322	0.009	34.356	0.000	0.322	0.439
.x3	0.769	0.019	40.216	0.000	0.769	0.658
.x4	0.780	0.018	42.190	0.000	0.780	0.794
.x5	0.414	0.012	34.833	0.000	0.414	0.451
.x6	0.444	0.013	33.110	0.000	0.444	0.410

Step 5 Between CFA

- What is it: Factor analysis of between level using between-group covariance matrix, Σ_B
- Why do it: Evaluates appropriateness of between structure
- Yields model fit and loadings at between level
- Might consider changing structure if fit is poor (and, again, documenting that you did so)
  - EFA with between-group covariance matrix
- Like within-only model, can be used to inform starting values for the MLMM if there are convergence issues

Step 5

Between CFA




```
between <- '

    engagement_b =~ NA*x1 + x2 + x3 + x4 + x5 + x6
    engagement_b ~~ 1*engagement_b

'

between_fit <- cfa(between, sample.cov = x$b.cov, sample.nobs = x$G)
summary(between_fit, fit.measures = T, standardized = T)
```

Step 5 Between CFA

```
Model Test User Model:

  Test statistic                               257.623
  Degrees of freedom                                 9
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                              1706.279
  Degrees of freedom                                15
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.853
  Tucker-Lewis Index (TLI)                       0.755
```

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -2450.816
  Loglikelihood unrestricted model (H1)      -2322.005

  Akaike (AIC)                                4925.632
  Bayesian (BIC)                              4968.080
  Sample-size adjusted Bayesian (BIC)         4930.038

Root Mean Square Error of Approximation:

  RMSEA                                          0.330
  90 Percent confidence interval - lower         0.296
  90 Percent confidence interval - upper         0.365
  P-value RMSEA <= 0.05                          0.000

Standardized Root Mean Square Residual:

  SRMR                                           0.071
```

Step 5

Between CFA

Latent Variables:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
engagement_b =~						
x1	1.643	0.080	20.553	0.000	1.643	0.956
x2	2.027	0.097	20.910	0.000	2.027	0.965
x3	1.564	0.109	14.318	0.000	1.564	0.764
x4	1.260	0.090	14.030	0.000	1.260	0.753
x5	1.901	0.122	15.568	0.000	1.901	0.809
x6	1.573	0.094	16.763	0.000	1.573	0.848

Variances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
engagement_b	1.000				1.000	1.000
.x1	0.253	0.037	6.846	0.000	0.253	0.086
.x2	0.303	0.052	5.870	0.000	0.303	0.069
.x3	1.747	0.162	10.760	0.000	1.747	0.417
.x4	1.212	0.112	10.793	0.000	1.212	0.433
.x5	1.913	0.181	10.578	0.000	1.913	0.346
.x6	0.963	0.093	10.322	0.000	0.963	0.280

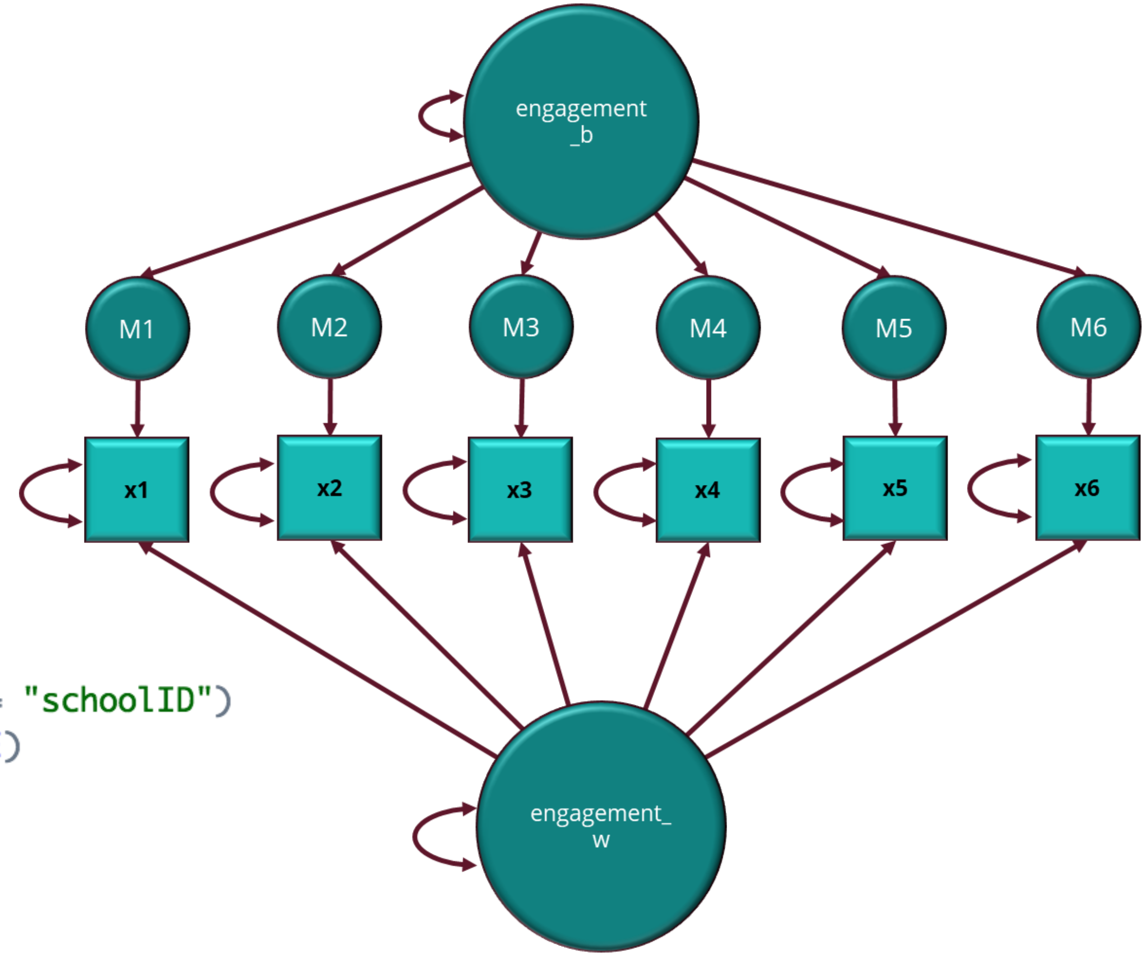
Step 6 Multilevel CFA

- What is it: simultaneous factor analysis of both within- and between-group covariance matrices
- Why do it: model building all in one place is easier, and allows for ease of building on a structural model
- What will it tell you: loadings and fit information for within and between models simultaneously

Step 6

Multilevel CFA

```
multilevel <- '

    level: 1
        engagement_w =~ NA*x1 + x2 + x3 + x4 + x5 + x6
        engagement_w ~~ 1*engagement_w

    level: 2
        engagement_b =~ NA*x1 + x2 + x3 + x4 + x5 + x6
        engagement_b ~~ 1*engagement_b

'
multilevel_fit <- cfa(model = multilevel, data = data, cluster = "schoolID")
summary(multilevel_fit, fit.measures = TRUE, standardized = TRUE)
```



Step 6 Multilevel CFA

```
Model Test User Model:

  Test statistic                              2207.215
  Degrees of freedom                                18
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                             10988.837
  Degrees of freedom                                30
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.800
  Tucker-Lewis Index (TLI)                       0.667
```

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -28526.495
  Loglikelihood unrestricted model (H1)     -27422.888

  Akaike (AIC)                               57112.991
  Bayesian (BIC)                             57301.007
  Sample-size adjusted Bayesian (BIC)        57205.680

Root Mean Square Error of Approximation:

  RMSEA                                          0.177
  90 Percent confidence interval - lower         0.171
  90 Percent confidence interval - upper         0.183
  P-value RMSEA <= 0.05                          0.000

Standardized Root Mean Square Residual (corr metric):

  SRMR (within covariance matrix)                0.082
  SRMR (between covariance matrix)               0.083
```

Step 6

Multilevel CFA

Level 1 [within]:

Latent Variables:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
engagement_w =~						
x1	0.554	0.015	37.942	0.000	0.554	0.709
x2	0.618	0.016	39.443	0.000	0.618	0.721
x3	0.664	0.019	35.735	0.000	0.664	0.609
x4	0.437	0.017	25.388	0.000	0.437	0.440
x5	0.744	0.016	45.134	0.000	0.744	0.770
x6	0.837	0.018	46.600	0.000	0.837	0.797

Intercepts:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
.x1	0.000				0.000	0.000
.x2	0.000				0.000	0.000
.x3	0.000				0.000	0.000
.x4	0.000				0.000	0.000
.x5	0.000				0.000	0.000
.x6	0.000				0.000	0.000
engagement_w	0.000				0.000	0.000

Variances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
engagement_w	1.000				1.000	1.000
.x1	0.303	0.012	24.479	0.000	0.303	0.497
.x2	0.353	0.015	24.274	0.000	0.353	0.480
.x3	0.746	0.020	36.689	0.000	0.746	0.629
.x4	0.799	0.020	40.303	0.000	0.799	0.807
.x5	0.381	0.015	24.598	0.000	0.381	0.407
.x6	0.403	0.019	21.618	0.000	0.403	0.365

Step 6 Multilevel CFA

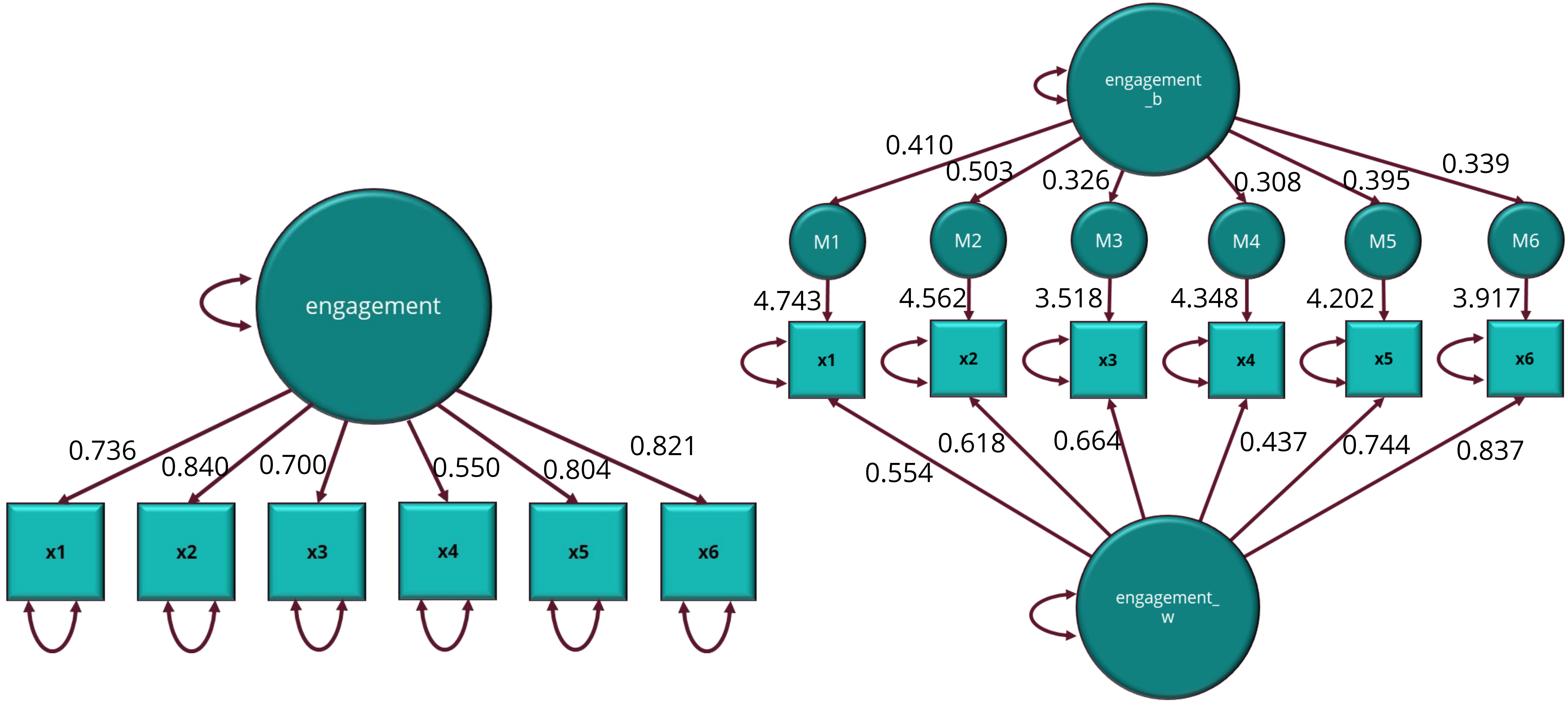
```
Level 2 [schoolID]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  engagement_b =~
    x1                0.410    0.023   17.766    0.000    0.410    1.006
    x2                0.503    0.028   18.274    0.000    0.503    1.006
    x3                0.326    0.031   10.528    0.000    0.326    0.802
    x4                0.308    0.025   12.268    0.000    0.308    0.911
    x5                0.395    0.034   11.783    0.000    0.395    0.799
    x6                0.339    0.027   12.377    0.000    0.339    0.922

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                4.743    0.029  163.740    0.000    4.743   11.636
   .x2                4.562    0.035  130.713    0.000    4.562    9.117
   .x3                3.518    0.032  110.771    0.000    3.518    8.672
   .x4                4.348    0.027  160.013    0.000    4.348   12.877
   .x5                4.202    0.035  118.453    0.000    4.202    8.504
   .x6                3.917    0.029  133.737    0.000    3.917   10.655
    engagement_b      0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    engagement_b      1.000                               1.000    1.000
   .x1               -0.002    0.002   -1.066    0.286   -0.002   -0.012
   .x2               -0.003    0.003   -1.033    0.302   -0.003   -0.012
   .x3                0.059    0.011    5.459    0.000    0.059    0.356
   .x4                0.019    0.008    2.540    0.011    0.019    0.170
   .x5                0.088    0.011    7.771    0.000    0.088    0.361
   .x6                0.020    0.006    3.497    0.000    0.020    0.150
```

Variances can't be negative, these are likely near-zero

Comparing CFA Results

Models	χ^2	df	CFI	RMSEA	SRMR
Regular	2226.990	9	0.820	0.700	0.079
Within	2109.180	9	0.790	0.245	0.081
Between	257.623	9	0.853	0.330	0.071
Multilevel	2207.215	18	0.800	0.177	W = 0.082 B = 0.083

Comparing Regular and Multilevel CFA Results



Setting Starting Values

- As mentioned, you might want to run the individual-level models if your initial MLMM doesn't converge to set starting values
- To set starting values, you use the special *start()* function in your model specification

```
multilevel <- '

    level: 1
        engagement_w =~ NA*x1 + start(0.642)*x2 + start(0.633)*x3 +
                        start(0.450)*x4 + start(0.710)*x5 + start(0.800)*x6
        engagement_w ~~ 1*engagement_w

    level: 2
        engagement_b =~ NA*x1 + start(2.027)*x2 + start(1.564)*x3 +
                        start(1.260)*x4 + start(1.901)*x5 + start(1.573)*x6
        engagement_b ~~ 1*engagement_b

'
```

- The values input to *start()* are those from the individual-level models

Summary

- Multilevel measurement modelling in an SEM framework is more flexible than in an MLM framework
- A multi-step process is recommended for understanding your modelling: check clustering, regular CFA, within-only, between-only, multilevel
- To conduct a multilevel CFA, the total covariance matrix is partitioned into within and between covariance matrices, either manually or automatically with software
- There is a lot of room for decision-making, so documenting your process is important

You have reached the end of this section…

Shaw, M. & Flake, J. K. (2023). Multilevel Measurement Models [Digital ITEMS Module 34]. *Educational Measurement: Issues and Practice, 34*(4), 82.