Multiple Membership Models: A tutorial

Laura Lambert

2023-05-02

Contents

Chapter 1

About

This is a fictional scenario and data set designed for the sole purpose of illustrating the coding and interpretation of multiple membership models using the R2MLwiN package. These data were simulated by myself, and were designed to loosely resemble a population of students at JMU. These data were not actually gathered from JMU students. This is designed to be tutorial-style, with background information about the models presented prior to walking through model building and output interpretation.

1.1 R packages used

Calculations and data manipulations performed in this tutorial made use of the following R packages: Bookdown (Xie [2023])
knitr (Xie [2022])
Rmarkdown (Allaire et al. [2022])
Tidyverse (Wickham [2021])
ggplot2 (Wickham et al. [2022])
misty (Yanagida [2023]) R2MLwiN (Zhang et al. [2023])

The software MLwiN, v3.06 [Charlton et al.] was also used, with Bayesian MCMC estimation [Browne]

1.2 Acknowledgements

Much of the background information about multiple membership models was gathered from the learning resources provided by the University of Bristol's Center for Multilevel Modelling (LEM).

1.3 A Disclaimer

The data used in this tutorial were simulated by me, and I am certain that a better job could have been done to better reflect some dependencies that should be in the data. One known omission is variance components were not specified prior to estimating the outcome. However, these data were simulated to reflect a student population at JMU to the best of my ability at the time and no intentional errors or misrepresentations were intended. Data-generating code is included in the Appendix.

Chapter 2

Introduction

When evaluating data in the educational field and in many other fields, researchers often run into the problem of their data being nested or clustered. When this occurs, they are no longer able to use multiple regression, as they would be violating the independent observations assumption, and may instead turn to multilevel modeling. With multilevel modeling, a researcher is able to handle students nested in schools and patients nested in nurses, as well as longitudinal models where time points are nested within individuals. However, a key facet to all of these models is the strict hierarchical nature of the data: each level one unit (students, patients, or time points) is nested within one, and only one, level 2 unit (school, nurse, individual). If we were to draw this relationship, it would look like Figure 1: there are no lines that are crossed, and each level one unit is associated with only one level two unit. This can also be represented in table form; a true hierarchical model will only have one row.

Two level strict hierarchical models can be written as two separate equations, each representing a level of the model.

$$y_{ij} = \beta_{0j} + r_{ij}$$

$$\beta_{0j} = \gamma_{00} + u_{0j}$$

,

Table 2.1: (#tab:true hierarchical table)Data distribution indicating a true hierarchical model

student	Math_Teacher_1	Math_Teacher_2	Math_Teacher_3
student	1, 2	3, 4	5, 6

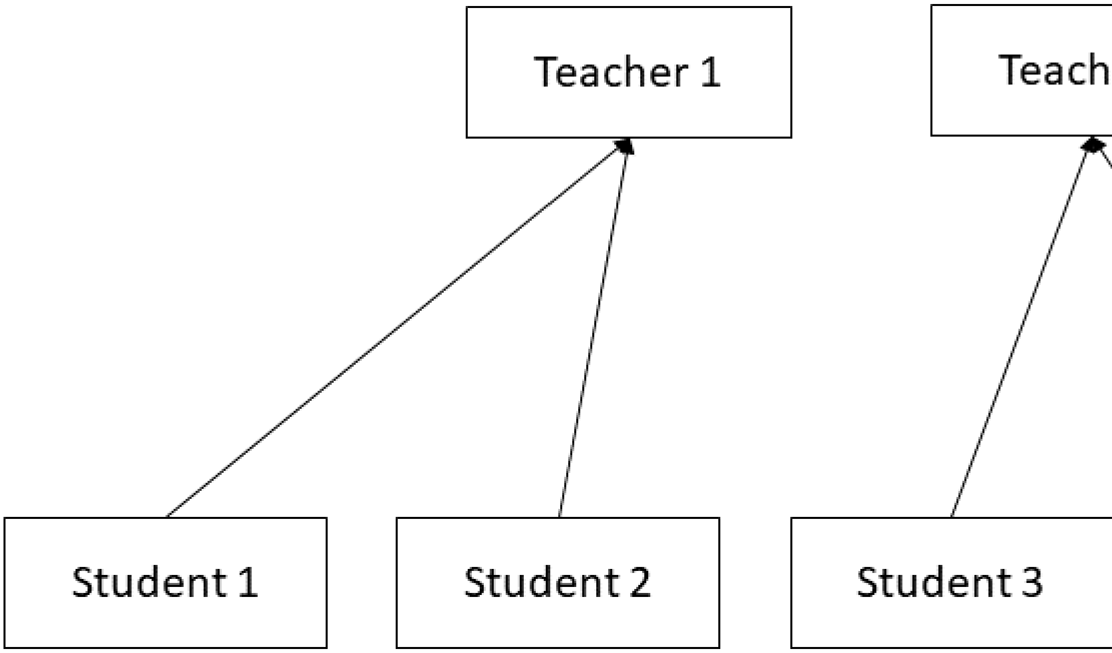


Figure 2.1: Figure 1: A pure hierarchical model

(#fig:fig 1)

With these equations, y_{ij} is outcome and β_{0j} represents intercepts. In the intercept-only model presented, there are no slope components, but if we were to add in predictors, we could examine how slopes vary. r_{ij} is the random variation in level one, while u_{0j} is random variation in level 2 [Snijders and Bosker, 2012].

2.1 Non-hierarchical models in brief

While data can be purely hierarchical as described above, situations often arise when level 1 units do not fit neatly into one and only one level 2 cluster. One example of this type of non-hierarchical model is a cross-classified model. In this instance, level 1 units are members of more than one level two cluster, but there is not a pure hierarchy. For example, students may be nested within both biology classes and chemistry classes, with the outcome measurement being science knowledge. It would seem wise to take into account both the impact of biology classes as well as chemistry classes. However, students who were in the same biology class are not all in the same chemistry class. This situation would require the use of a cross-classified model. Cross-classified models, like pure hierarchical models, can also be represented in table or visual form. In table form, the cross classified structure becomes evident when there are level one units in more than one row and column. In the simple example below, there are multiple cells filled in both rows (Physics Teacher 1 and 2) and in columns (Math Teacher 2) [Beretvas, 2011].

Another type of non-hierarchical model is a multiple membership model. In this instance, there is only one level 2 cluster being considered, but the level 1 units do not all belong to one and only one cluster. These types of models are frequently employed in education to model what are termed mobile students: students who attend more than one school over a given time frame. An example of mobile students might be those who attend more than one elementary school prior to a state-wide 5th grade test. Multiple membership data are also evident in table form, as well as in a unit diagram. As with the other diagrams, however, these can quickly become visually confusing with large data sets [Leckie, 2013a, Beretvas [2011]].

Moving forward, we will focus exclusively on multiple membership models, going more in-depth into their intricacies as well as walking through a tutorial running a model and interpreting the output.

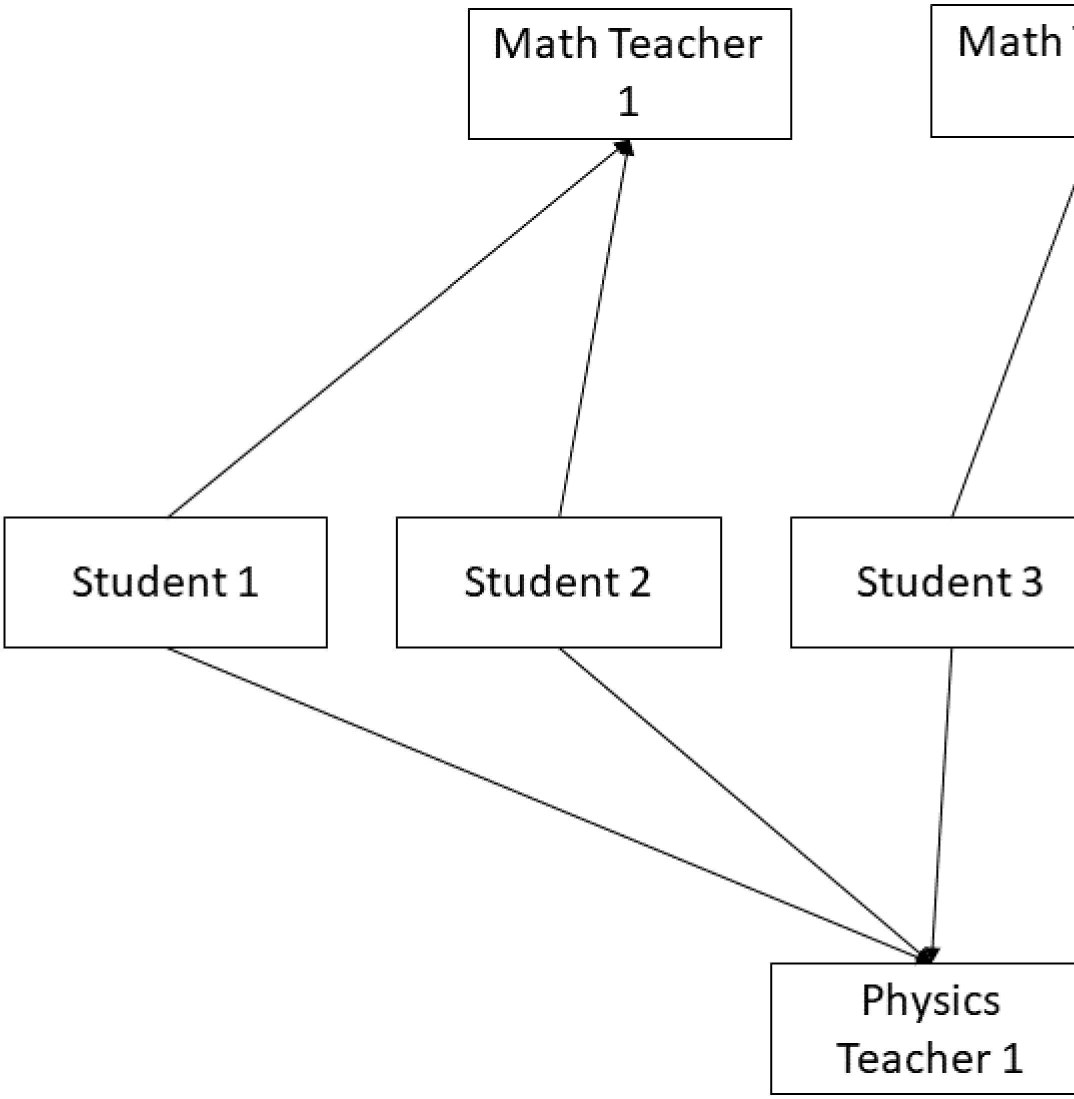


Figure 2.2: Figure 2: A Cross-classified model

(#fig:fig 2)

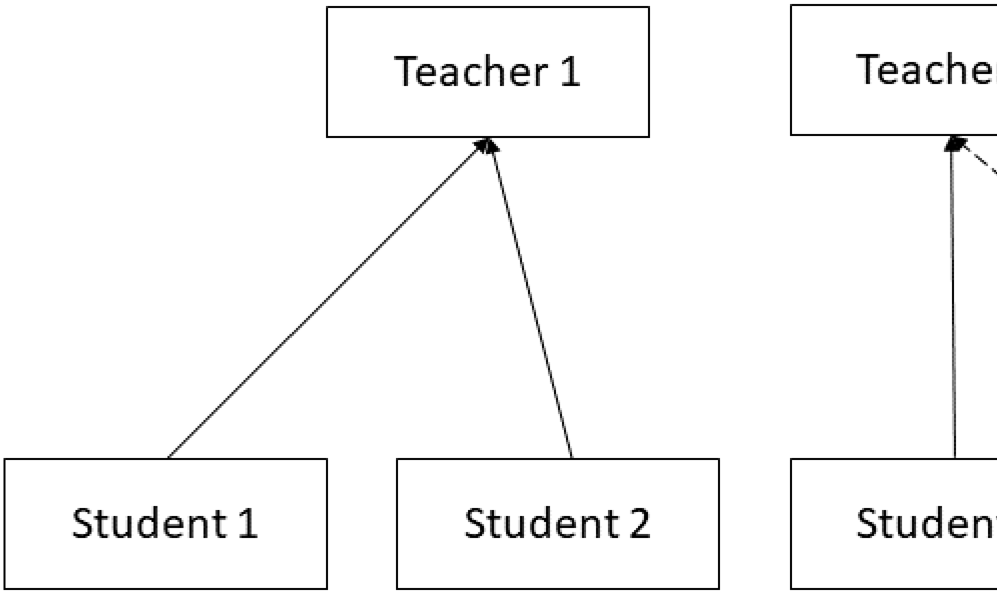


Figure 2.3: Figure 3: A Multiple Membership model
(#fig:fig 3)

Chapter 3

Data

The data used for this tutorial were simulated in R to loosely represent a cohort of JMU students. As accurate counts and frequencies of the predictors were not available outside of a request to PAIR, "best guesses" were employed where necessary. In a fictional scenario, JMU has decided to add a Math Achievement Test as a graduation requirement: students must take this test the semester they anticipate graduating as a measure of their Math Achievement at JMU, with the point of comparison being a cut-score. The stakes are similar to Assessment Day and there are no penalties for low scores. JMU, recognizing that many factors may play into how well a student does on this test, wants to consider the role different math teachers play as well as some student-level predictors. Not wanting this to hold any penalty towards teachers either, JMU has assigned each teacher an anonymous ID and associated it with the years of experience. JMU is additionally considering what teacher qualities help students be more successful, with the idea of offering more professional development in those areas. Currently, the only teacher-level predictor available is number of years of experience of the instructor.

JMU hopes to answer the following research questions: RQ1: Do teacher characteristics influence student math scores? RQ2: What student characteristics predict math scores? RQ3: Does more teacher experience lead to better student outcomes?

The predictors are listed in the table below, along with a brief description.

3.1 Compact vs. wide forms

The data are in a .cvs file in "compact" form, as opposed to "wide" form. Compact form contains two different sets of variables: one set for the first through twelfth teacher to instruct each student and another set represent the multiple

Table 3.1: Simulated dataset that will be used for analysis

Predictor	Description
MathAch	Mach Achievement; outcome variable, score on a (fictional) math test given to stud
S_ID	Student ID; a single number representation of students, from 1 to 5200
SAT_M	SAT math score; ranges from 200-800 and represents the SAT math score of studen
S_gend	Student gender; a non-binary gender indicator with 0 = female, 1 = male, and 2 =
female, NB	Dummy coded student gender, since gender is categorical. Male is the reference
S_SES	Student SES; values range from 2-29
num_tchrs	Number of teachers; the number of math teachers a student had at JMU
phys	Physics; if a student took physics or not, with 0 = no and 1 = yes
phys_tchr	Physics teacher; if a student took physics, the ID of the teacher they had (ranging
tchr1-tchr12	Teachers 1 through 12; teacher IDs (ranging from 1-56) of teachers students had. I
w1-w12	Weights; these represent the amount of time spent with each teacher. Values can ra
t1_exp-t12_exp	Teacher experience; Given the compact nature of this dataset, the experience of th

membership weight variables. Wide form would have the same information, but in only one set of variables representing the individual teacher IDs and the proportion of time each teacher spent instructing each student in the cells (see tables below).

3.1.1 Compact

In compact form, there is a set of variables for the max number of level 2 units encountered, and another set of variables for the weights. Adding in a level-2 predictor here would only necessitate adding in the appropriate number of columns for the number of level 2 units encountered for that predictor. Using our data as an example, we have teacher experience as a level 2 predictor, and 12 possible teacher encounters. To add in teacher experience, we would add in 12 columns, t_exp1-t_exp12, which would populate with the appropriate experience for the teacher the student had. The weight would be calculated via the weight columns.

3.1.2 Wide

While perhaps less evident in this small example, if there were more teacher IDs (56, as in our data for example), but a small number of level 2 units encountered and the associated weights (12 of each in our data), the 'wide' descriptor would become more evident. The wide descriptor becomes even more evident when considering level two predictors in the model. In the wide format, every level-2 predictor we add will be calculated across all the level-2 ids. Using our data

Table 3.2: (#tab:compact ex)Data in compact form

Student	Teacher.1	Teacher.2	Teacher.3	w1	w2	w3
1	2	4	1	0.33	0.33	0.33
2	2	3	1	0.33	0.33	0.33
3	3	1	4	0.33	0.33	0.33
4	0	3	0	0.00	1.00	0.00
5	1	0	0	1.00	0.00	0.00
6	0	1	3	0.00	0.50	0.50
7	0	1	2	0.00	0.50	0.50
8	0	0	1	0.00	0.00	1.00
9	1	2	0	0.50	0.50	0.00
10	4	0	2	0.50	0.00	0.50

Table 3.3: (#tab:wide ex)Data in wide form

Student	T1	T2	T3	T4
1	0.33	0.33	0.00	0.33
2	0.33	0.33	0.33	0.00
3	0.33	0.00	0.33	0.33
4	0.00	0.00	1.00	0.00
5	1.00	0.00	0.00	0.00
6	0.50	0.00	0.50	0.00
7	0.50	0.50	0.00	0.00
8	1.00	0.00	0.00	0.00
9	0.50	0.50	0.00	0.00
10	0.00	0.50	0.00	0.50

as an example, we have teacher experience as a level-2 predictor. Adding it in using a wide format would mean including 56 columns for the teacher IDs, with weights in the cell to represent the time each student spent with that teacher. Then, there would be another 56 columns for the teacher experience for each of the teachers, with each of their respective experiences.

While the wide form is less efficient, some programs require the data to be in one form or the other. In this instance, MLwiN using Bayesian routines (what we will be using) requires compact form [Leckie, 2013a].

3.2 Examining the Data

The next step is to determine what type of predictors we have and what they look like, as well as if we have a true hierarchical model, a "nesting as nuisance" model, or a non-hierarchical model such as cross-classified or multiple membership models. We will be looking for if students (in this case) belong to one and only one cluster (teachers), which would indicate a hierarchical approach or perhaps a "nesting as nuisance" approach. However, if, as is the case, students belong to multiple clusters, we will be taking a non-hierarchical approach. For this example, we are only considering one cluster, so we will be using a multiple membership design. We can see in the `num_tchrs` column there are students ranging from having had 1 teacher to 12 teachers, indicating we have a multiple membership column. We could also look at the `tchr1` through `tchr12` columns, and see there are values in more than one of those - indicating that students have encountered more than one math teacher.

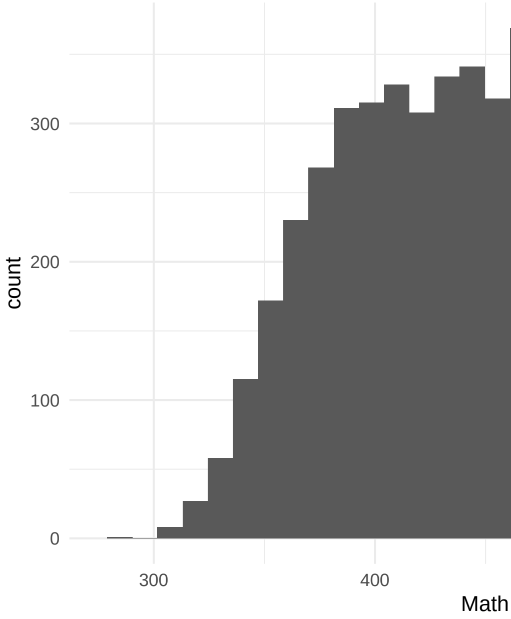
```
#Read in the data file
StudData <- read.csv("exampledata2.csv")

#See what it looks like
head(StudData)
```

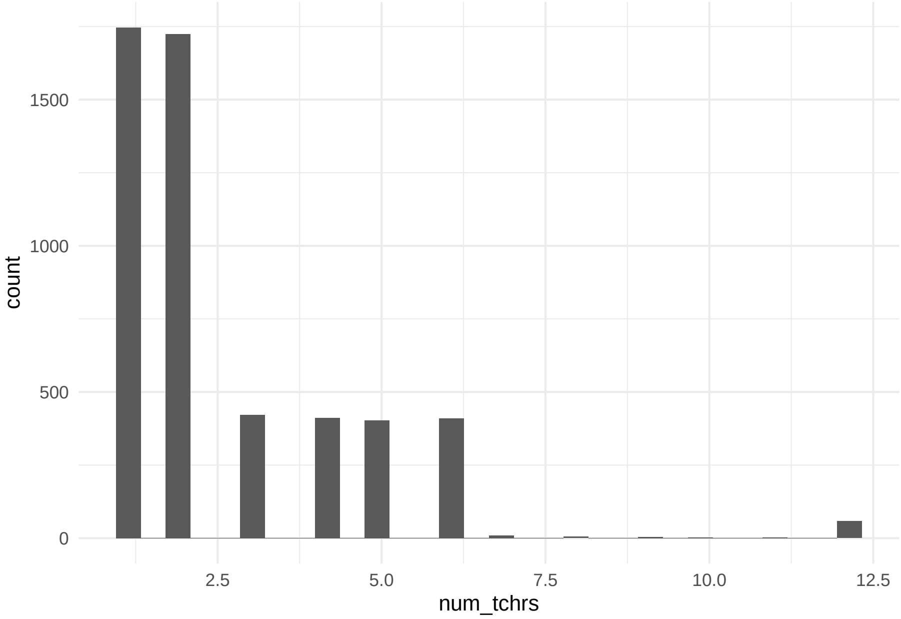
```
##   X.2 X.1 X S_ID S_gend     S_SES num_tchrs phys tchr1 tchr2 tchr3 tchr4 tchr5 tchr6
## 1   1   1 1    1      0 14.880674         3    0    24    32    46     0     0     0
## 2   2   2 2    2      0 14.575710         4    0    30    52    54    18     0     0
## 3   3   3 3    3      1 13.173884         1    0    12     0     0     0     0     0
## 4   4   4 4    4      1 13.065602         2    0    32    30     0     0     0     0
## 5   5   5 5    5      0 16.105715         2    0     3    48     0     0     0     0
## 6   6   6 6    6      1  8.190244         2    1     3    44     0     0     0     0
##       Math t10_exp    t1_exp    t2_exp   t3_exp   t4_exp t5_exp t6_exp t7_exp t8_exp
## 1 530.3671       0  9.512710  9.603359 11.08068 0.000000      0      0      0      0
## 2 523.2353       0 10.138500 12.664019 11.33673 4.434515      0      0      0      0
## 3 457.6296       0 12.682655  0.000000  0.00000 0.000000      0      0      0      0
## 4 504.4141       0  9.603359 10.138500  0.00000 0.000000      0      0      0      0
## 5 443.2032       0  9.541097 12.631969  0.00000 0.000000      0      0      0      0
## 6 431.5336       0  9.541097 11.309230  0.00000 0.000000      0      0      0      0
```

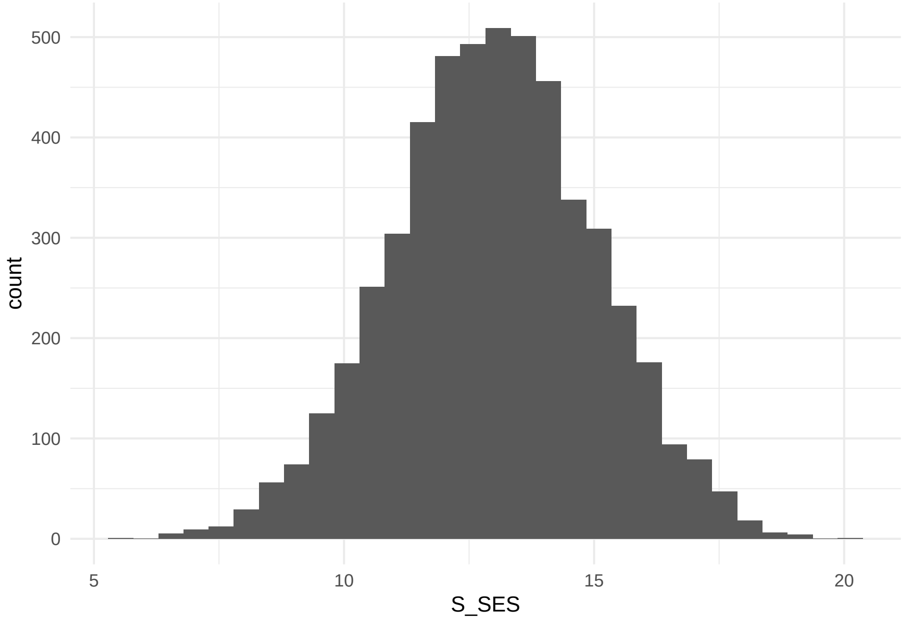
```
summary(StudData)
```

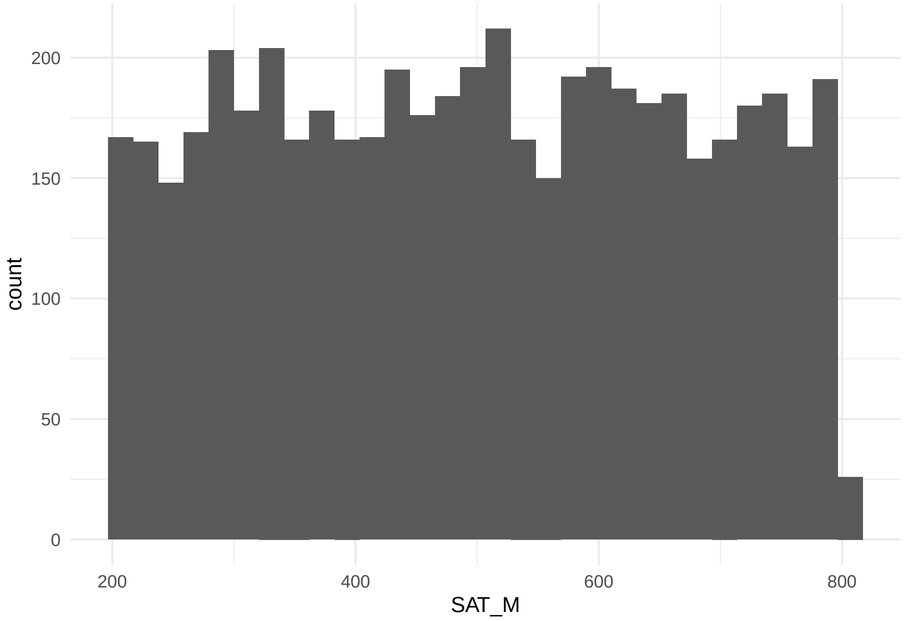
```
##       X.2            X.1             X             S_ID          S_gend          S_
##  Min.   :   1   Min.   :   1   Min.   :   1   Min.   :   1   Min.   :0.0000   Min.
##  1st Qu.:1301   1st Qu.:1301   1st Qu.:1301   1st Qu.:1301   1st Qu.:0.0000   1st Qu
##  Median :2600   Median :2600   Median :2600   Median :2600   Median :0.0000   Median
##  Mean   :2600   Mean   :2600   Mean   :2600   Mean   :2600   Mean   :0.4708   Mean
## 3rd Qu.:3900   3rd Qu.:3900   3rd Qu.:3900   3rd Qu.:3900   3rd Qu.:1.0000   3rd Qu.:14.353
## Max.   :5200   Max.   :5200   Max.   :5200   Max.   :5200   Max.   :2.0000   Max.   :20.249
##     tchr4            tchr5           tchr6            tchr7             tchr8             tch
## Min.   : 0.000   Min.   : 0.00   Min.   : 0.000   Min.   : 0.0000   Min.   : 0.0000   Min.
## 1st Qu.: 0.000   1st Qu.: 0.00   1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.
## Median : 0.000   Median : 0.00   Median : 0.000   Median : 0.0000   Median : 0.0000   Median
## Mean   : 7.238   Mean   : 5.06   Mean   : 2.625   Mean   : 0.4515   Mean   : 0.4065   Mean
## 3rd Qu.: 1.000   3rd Qu.: 0.00   3rd Qu.: 0.000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.
## Max.   :57.000   Max.   :56.00   Max.   :57.000   Max.   :55.0000   Max.   :56.0000   Max.
##       w2               w3                w4                w5                w6
## Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.
## 1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st
## Median :0.2000   Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000   Medi
## Mean   :0.2424   Mean   :0.07666   Mean   :0.04988   Mean   :0.03008   Mean   :0.01458   Mean
## 3rd Qu.:0.5000   3rd Qu.:0.16600   3rd Qu.:0.08300   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd
## Max.   :0.5000   Max.   :0.33000   Max.   :0.25000   Max.   :0.20000   Max.   :0.16600   Max.
##      w11                 w12                 Math          t10_exp            t1_exp
## Min.   :0.0000000   Min.   :0.0000000   Min.   :283.9   Min.   : 0.0000   Min.   : 4.435   Mi
## 1st Qu.:0.0000000   1st Qu.:0.0000000   1st Qu.:396.6   1st Qu.: 0.0000   1st Qu.: 9.300   1s
## Median :0.0000000   Median :0.0000000   Median :442.4   Median : 0.0000   Median :10.725   Me
## Mean   :0.0009777   Mean   :0.0009258   Mean   :442.0   Mean   : 0.1229   Mean   :10.453   Me
## 3rd Qu.:0.0000000   3rd Qu.:0.0000000   3rd Qu.:486.1   3rd Qu.: 0.0000   3rd Qu.:11.512   3r
## Max.   :0.0900000   Max.   :0.0830000   Max.   :614.1   Max.   :14.0426   Max.   :14.728   Ma
##     t7_exp            t8_exp            t9_exp           t11_exp           t12_exp          p
## Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min
## 1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st
## Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000   Med
## Mean   : 0.1677   Mean   : 0.1475   Mean   : 0.1378   Mean   : 0.1203   Mean   : 0.1151   Mea
## 3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd
## Max.   :14.7281   Max.   :13.9805   Max.   :14.7281   Max.   :14.7281   Max.   :14.7281   Max
```



We can also see distributions of the outcome and predictors.




500
400
300
200
100
0
count
5
10
15
20
S_SES


200
150
100
50
0
count
200
400
600
800
SAT_M

3.3 Software Considerations

MLwiN[Charlton et al.] is one of the most commonly found software programs for running multiple membership models, as it can natively handle such complex models. I have found that SAS and Stata can be 'tricked' into running simple multiple membership models by fitting them as constrained hierarchical models, but with non-continuous outcomes or increased complexity they quickly become computationally inefficient. More worryingly is that the parameter estimates generated by these methods are found to be biased. The same sources say that R can also be tricked, but I have not found that to be the case [Leckie, 2013b]. That said, both R and Stata have functions or packages that allow you to use their interface and run MLwiN in the background, both to allow use of a familiar interface but also to allow for accurate estimation methods with full model recognition. MLwiN is not a free software and is produced by the University of Bristol. A 30-day free trial with full functionality is available to anyone in the world, and for researchers in the US, a single-user license is £400 or a PhD license is available for £225, though it expires after 3 years. In order to perform the analyses demonstrated below, I obtained a 30-day free license and used the R2MLwiN [Zhang et al., 2023] package to allow me to use an R interface. While the R code is provided, it will not work unless you also have a valid MLwiN license.

Heck et al. [2022] used OpenBUGS for some more complex multiple membership cross classified modeling of longitudinal data, but at this time OpenBUGS website no longer exists and has been ported to MultiBUGS [Mul, Goudie et al. [2020]]. One potential drawback of this program is the mis-classification of its software framework by virus detection programs as a virus, leading it to be uninstallable or to throw errors. MultiBUGS also reports that an R interface (R2MultiBUGS) is under development.

Chapter 4

Multiple Membership models: the details

The key feature of a multiple membership model is that level 1 units do not belong to one and only one level 2 cluster. There may be some units that exhibit true hierarchy, but there are others that are 'multiple members': they belong to 2 or more clusters. A classic example of this is mobile students in educational settings (eg. Chung and Beretvas [2012], Heck et al. [2022], Leroux [2019]). Another example in the health care setting is patient outcomes, but each patient frequently sees more than one nurse [Leckie, 2013a]. Using a multiple membership model allows the influence of multiple level 2 clusters to be taken into account. Additionally, just like ignoring hierarchical data violates the independent observations assumption and cause an inflated type I error rate, ignoring a multiple membership structure of your data will also cause an inflated type I error rate.

Historically, when researchers encountered a multiple membership data structure (ie, mobile students) and were unable to model it correctly, there were a few options to handle this data that they would choose from, all of which resulted in mis-specified models. In one option, researchers would only consider the unit to be a member of their most recent cluster. As an example, only consider the school a child was in at the time of testing, rather than any other previous schools. Another option was to simply delete the mobile students from the data set, leaving only a pure hierarchy of students nested in schools. A third option was to acknowledge the mobility of students by including a dichotomous or ordinal "mobility" predictor to include in a regular hierarchical model. This would control for the influence of mobility on the outcome, but did not allow for the modeling of the effects of different schools on the student. However, none of these methods are an appropriate treatment of multiple membership data. The results obtained from the 'delete mobile students' approach, for example, can only be generalized to other non-mobile students as well as decreasing the power.

In the study by Chung and Beretvas [2012], they reported negative bias in the level 2 predictor coefficients and in the level 2 variance components when employing the ‘only count the last school’ technique. In the same study, they also found that the level 1 variance components were positively biased. It is for these reasons that researchers investigating multiple membership data use a multiple membership model, in order to allow for generalization of results, proper capturing of mobile student effects, and validity of inferences about school effects [Smith and Beretvas, 2017].

4.1 Notation

There are three possibilities for how to depict these models, each with its own set of advantages and disadvantages. Standard hierarchical notation, which we are familiar with from truly hierarchical models, uses subscripts to identify individual and cluster level predictors and effects. The disadvantage of this notation is that, due to its familiarity, individuals may incorrectly assume that unit i is strictly nested within cluster j, which is not the case (Snijders and Bosker [2012]; Leckie [2013a]). Another method is to use multiple subscript notation (Beretvas [2011]). This notation makes the non-hierarchical structure more apparent, however, it can quickly become challenging to read.

The notation we will use moving forward is classification notation [Leckie, 2013a]:

$$y_i = \beta_0 + \sum_{j \in school(i)} w^{(2)}_{j,i} u^{(2)}_j + e_i$$

$$u^{(2)}_j \sim N(0, \sigma^2_{u(2)})$$

$$e_i \sim N(0, \sigma^2_e)$$

With this notation, the multiple membership component is represented by $\sum_{j \in school(i)} w^{(2)}_{j,i} u^{(2)}_j$, which represents the weighted ($w^{(2)}_{j,i}$) cluster effects ($u^{(2)}_j$). The weight component represents the amount of time unit i spent in cluster j, with the superscript (2) representing this is associated with the second level. All levels in the model are indicated, with level 1 (1) being implicit. The superscripts/subscripts for each level will identify to which level the random effects, variance parameters, and covariance parameters are associated with.

4.2 Weights

Weights are assigned to the level two clusters to reflect the different influences each one has on the level 1 units, and sum to one for any given level 1 unit. These are decided by the researcher, who may take one of several approaches.

One approach is to assign equal weights, regardless of time. This is the approach taken in the example data set for this tutorial: it was assumed that each teacher seen by a student had equal influence, and the data set was designed in such a way that students did not see any given teacher more than once. In a mobile student situation, it may be that if a student attended a school for any length of time, that school would be given an equal weight to all the other schools attended by the student, even if they attended other schools for longer. Another approach is to account for length of time. In the previous example, if a student spent 1 year at school 1 but 2 years at school two, a researcher may weight school one as 0.33 and school 2 as 0.67. This will take into account the fact that since the student spent more time at school 2, it likely had a larger impact. A third approach seen is weighting later educational settings more heavily than earlier ones, to reflect the fact that the more recent education may have a larger impact on test scores (for example) than earlier education [Chung and Beretvas, 2012].

If we look at the data (**??**data), we can pull two students to see the earlier equation written out for each of them, student 1 and student 3. For student 1, we see that they had three teachers (24, 32, and 46) while student 3 only had one teacher (12). The equations for students 3 and 1 are shown below, indicating the role cluster weights play in these models.

4.2.1 Student 3

$$y_3 = \beta_0 + \sum_{j \in \{12\}} w^{(2)}_{j,3} u^{(2)}_j + e_3$$

$$= \beta_0 + w^{(2)}_{12,3} u^{(2)}_{12} + e_3$$

$$= \beta_0 + u^{(2)}_{12} + e_3$$

4.2.2 Student 1

$$y_1 = \beta_0 + \sum_{j \in \{24,32,46\}} w^{(2)}_{j,1} u^{(2)}_j + e_1$$

$$= \beta_0 + w^{(2)}_{24,1} u^{(2)}_{24} + w^{(2)}_{32,1} u^{(2)}_{32} + w^{(2)}_{46,1} u^{(2)}_{46} + e_1$$

$$= \beta_0 + 0.33 u^{(2)}_{24} + 0.33 u^{(2)}_{32} + 0.33 u^{(2)}_{46} + e_1$$

These weights are used both to identify amount of belonging to a particular level 2 cluster, but also to appropriately weight any level 2 predictors. For example, if we were to add in `t_exp` as teacher-level predictor, the resulting equations would look like this:

$$y_i = \beta_0 + \beta_1 \sum_{j \in school(i)} w_{j,i}^{(2)} t\_exp_{2j}^{(2)} + \sum_{j \in school(i)} w_{j,i}^{(2)} u_j^{(2)} + e_i$$

$$u_j^{(2)} \sim N(0, \sigma_{u(2)}^2)$$

$$e_i \sim N(0, \sigma_e^2)$$

The term $\sum_{j \in school(i)} w_{j,i}^{(2)} t\_exp_{2j}^{(2)}$ is teacher experience (`t_exp`), taking into account that since students saw different teachers, the experience of those teachers will be different. This is called the weighted sum of this predictor, with β_1 being the slope coefficient. We are able to keep the term as-is in our equation, or to simplify the equation, we could calculate it in our data set to be $\bar{x}_{2i}$. The resulting equation would then be

$$y_i = \beta_0 + \beta_1 \overline{(t\_exp)}_{2i} + \sum_{j \in school(i)} w_{j,i}^{(2)} u_j^{(2)} + e_i$$

4.3 Centering

In pure hierarchical models, predictors (typically at level 1) can either be group mean centered (CWC) or grand mean centered (CGM), with CWC being the generally preferred option due to the resulting interpretability of the coefficients [Enders, 2013]. However, in a multiple membership model, level 1 units belong to one or more level 2 clusters, meaning group mean centering can get really complicated really fast and indeed may not be possible. If centering is mentioned in the literature surrounding multiple membership models, it is only grand mean centering that is referenced. This may be due to ease, or due to lack of exploration using group mean centering. It may also be due to requiring a large sample size to have enough units in each unique 'cluster' combination. For example, students who just had teacher 1 would be a cluster, while students who had teacher 1 and teacher 2 would be another cluster, etc. In our example with 56 teachers, and a max draw of 12 teachers, there are slightly more combinations than our sample size by several orders of magnitude. Much of the literature using multiple membership models (eg. [Leckie, 2013a, Heck et al. [2022], Ma et al. [2021], Gero et al. [2020]]) did not mention any type of centering; an example where it was mentioned is Smith and Beretvas [2017], where grand mean centering is specifically addressed in their model descriptions.

4.4 VPC and ICC

4.4.1 ICC

Again comparing to pure hierarchical models, one of the first things that is calculated after running an intercept-only model is the ICC, or intraclass correlation. When calculated as $\frac{\tau_{00}}{\tau_{00}+\sigma^2}$, the ICC represents the amount of variance in the outcome variable that can be attributed to the fact that level 1 units are nested in level 2 clusters [Snijders and Bosker, 2012]. As with most things thus far, calculating an ICC for a multiple membership model is not so straightforward. The equation for a multiple membership model is more complex, and takes into account the cluster 'profile' of two units. If we are looking at the cluster level ICC for two different units, *i* and *i'*, we would calculate it as a correlation like so:

$$corr(y_i, y_{i'}) = \frac{\sigma^2_{u(2)} \sum_{j \in cluster(i) \cup cluster(i')} w^{(2)}_{j,i} w^{(2)}_{j,i'}}{\sqrt{\sigma^2_{u(2)} \sum_{j \in cluster(i)} w^{(2)}_{j,i} + \sigma^2_e}\sqrt{\sigma^2_{u(2)} \sum_{j \in cluster(i')} w^{(2)}_{j,i'} + \sigma^2_e}}$$

Interpreting the ICC as the correlation between the outcomes of two students who had the same teacher, and only that teacher, it would be calculated as

$$ICC = \frac{\sigma^2_{u(2)}}{\sigma^2_{u(2)} + \sigma^2_e}$$

However, if these students had two teachers each, and only shared one of them, the calculation for the ICC becomes

$$ICC = \frac{0.25\sigma^2_{u(2)}}{0.5\sigma^2_{u(2)} + \sigma^2_e}$$

This is because the weights for each teacher are incorporated to reflect the amount of time two students had similar teachers and the amount of time they had different teachers [Leckie, 2013a].

4.4.2 VPC

Another way to determine how variance is partitioned in a model is the variance partition coefficient. This reports the proportion of observed variation in the outcome that is at each level. The VPC is different from the ICC in that it is not the model implied correlation within a cluster, but the observed variation. As with the ICC, the multiple membership model is more complex in this calculation as well. The variation in outcomes that can be attributed to clusters depends on how much the level 1 units are spread across clusters: are they members of one cluster or multiple clusters? The cluster-level VPC equation representing the amount of variation in outcomes that is attributed to clusters in a multiple membership model is [Leckie, 2013a]

$$VPC_{u(2)} = \frac{\sigma^2_{u(2)} \sum_{j \in cluster(i)} (w^{(2)}_{j,i})^2}{\sigma^2_{u(2)} \sum_{j \in cluster(i)} (w^{(2)}_{j,i})^2 + \sigma^2_e}$$

If we consider the level 1 units that were members of only one cluster, the VPC is equivalent to the ICC for two individuals that had the same, single, teacher above:

$$VPC_{u(2)} = \frac{\sigma^2_{u(2)}}{\sigma^2_{u(2)} + \sigma^2_e}$$

As above, however, it gets more complex when level 1 units are members of more than one cluster, and it no longer equals the ICC. Following a similar pattern to above, if we consider the population of level 1 units that were members of two different clusters (which one is not important here; we are looking only at cluster-level variance), the VPC calculation becomes

$$VPC_{u(2)} = \frac{0.5\sigma^2_{u(2)}}{0.5\sigma^2_{u(2)} + \sigma^2_e}$$

What this illustrates is that as level 1 units are members of more clusters, the amount of variance in outcomes that is attributed to clusters decreases. Additionally, with a multiple membership model, this is perhaps the easier of the two to both compute and calculate, as we are not considering how the cluster pattern of two individuals compares but rather how many clusters an individual is a member of.

Chapter 5

Running a Multiple Membership Model

As noted in 3.3, we will be using MLwiN to evaluate the multiple membership models. There is an R package, R2MLwiN [Zhang et al., 2023] that allows for an R-interface which runs MLwiN in the background. This allows users to work with a program they are already familiar with while accessing the capabilities of MLwiN. However, as also previously noted, you must have an active MLwiN license, and have the program installed on your computer for your R code to work.

5.1 Intercept-only Model

We will first run an intercept-only model, looking at Math as an outcome and only including the intercept and the multiple membership effect of teachers. This model will inform us as to how the variance in Math scores is divided between teacher variance components (level 2) and student variance components (level 1).

$$y_i = \beta_0 + \sum_{j \in school(i)} w_{j,i}^{(2)} u_j^{(2)} + e_i$$

$$u_j^{(2)} \sim N(0, \sigma_{u(2)}^2)$$

$$e_i \sim N(0, \sigma_e^2)$$

5.1.1 First Steps

Setting this up to run in R, we first need to make sure we have MLwiN installed, as well as call the `R2MLwiN` package. Additionally, R2MLwiN defaults the path to MLwiN as "C:/Program Files/MLwiN v3.06/". If yours, like mine, is installed elsewhere, you need to include `options(MLwiN_pah = "path/to/MLwiN")`. An easy way to copy the path is to find it in your computer, hold 'shift' and right click the program. The option "copy as path" comes up as an option, and you can just paste it in.

5.1.2 Define the model

After we have loaded in the appropriate packages, we take a few steps to actually run the model. First, we define the intercept-only model, including both the teacher columns and the student IDs. For our dataset, our model definition will be `intonly <- Math ~ 1 + (1|tchr1) + (1|S_ID)`. Important to note is that when defining the teacher columns (`(1|tchr1)`), we use the first column name, not `tchr` as you might be inclined to do. The random part of the model, `(1|tchr1)` and `(1|S_ID)`, is written in descending order of hierarchy. With this model specification, we are allowing intercepts to vary at level 2 (`(1|tchr1)`) and level 1 (`(1|S_ID)`).

5.1.3 Set up and send to MLwiN

The next step is to define the multiple membership variables (teacher columns for us) and the associated weights. To do this, we make an object that is a list of list of lists. I named it `MultiMemb` in the code chunk below:

```
MultiMemb <- list(list(
  mmvar = list("tchr1", "tchr2", "tchr3", "tchr4", "tchr5", "tchr6", "tchr7", "tchr8", 
  weights = list("w1", "w2", "w3", "w4", "w5", "w6", "w7", "w8", "w9", "w10", "w11", "
```

Within this, `mmvar` specifies classification units (unit of multiple membership). `weights` is where the associated weights go, and is the student-level weighting (in our case) given to each teacher they had. The final `NA` indicates that level 1 has no multiple membership classification. This set-up is why our data needed to be in compact rather than long form (3.1).

After we have defined `MultiMemb`, we can actually run the model. This line is putting everything together:

```
(MMembModel1 <- runMLwiN(Formula = intonly, data = StudData, estoptions = list(EstM = 
```

Looking at it, we are calling MLwiN with `runMLwiN`, defining what the model should be with `Formula`, and defining our data set with `data`. `estoptions` is our estimation options. When `EstM` is equal to 1, we are using MCMC estimation. And lastly, within that same statement, `mm` is calling our `MultiMemb` object to match everything up and weight appropriately [Zhang et al., 2023].

5.1.4 Put it all together

Taking everything from above and putting it all together, we get the following code chunk, and then output.

```
library(R2MLwiN)
options(MLwiN_path="C:\\Program Files (x86)\\MLwiN trial\\i386\\mlwin.exe")

intonly <- Math ~ 1 + (1|tchr1) + (1|S_ID)

MultiMemb <- list(list(
  mmvar = list("tchr1", "tchr2", "tchr3", "tchr4", "tchr5", "tchr6", "tchr7", "tchr8", "tchr9", "
  weights = list("w1", "w2", "w3", "w4", "w5", "w6", "w7", "w8", "w9", "w10", "w11", "w12")), NA)

(MMembModel1 <- runMLwiN(Formula = intonly, data = StudData, estoptions = list(EstM = 1, drop.dat
```

```
## MLwiN is running, please wait......
```

```
##
## -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
## MLwiN (version: 2.36)  multilevel model (Normal)
##        N min     mean max N_complete min_complete mean_complete max_complete
## tchr1 56  75 92.85714 122         56           75      92.85714          122
## Estimation algorithm:  MCMC      Cross-classified              Elapsed time : 31.43s
## Number of obs:  5200 (from total 5200)          Number of iter.: 5000  Chains: 1  Burn-in: 500
## Bayesian Deviance Information Criterion (DIC)
## Dbar      D(thetabar)    pD      DIC
## 56520.996  56473.520  47.479     56568.477
## ----------------------------------------------------------------------------------------------
## The model formula:
## Math ~ 1 + (1 | tchr1) + (1 | S_ID)
## Level 2: tchr1     Level 1: S_ID
## ----------------------------------------------------------------------------------------------
## The fixed part estimates:
##                 Coef.   Std. Err.       z   Pr(>|z|)       [95% Cred.   Interval]   ESS
## Intercept   441.78207     2.38716  185.07          0  ***   436.82835   446.43202   346
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ----------------------------------------------------------------------------------------------
## The random part estimates at the tchr1 level:
##                      Coef.   Std. Err.   [95% Cred.   Interval]    ESS
## var_Intercept   281.90093    66.21423    176.86386   432.97710   2258
## -----------------------------------------------------------------------------------------------------
## The random part estimates at the S_ID level:
##                       Coef.   Std. Err.   [95% Cred.    Interval]    ESS
## var_Intercept   3077.47965    59.92732   2960.59277   3194.83199   4679
## -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
```

5.2 The Output

Looking at the output from our intercept only model, we have a lot of information. First, R and MLwiN are reminding us what we put in:

```
MLwiN (version: 2.36)  multilevel model (Normal)
       N min     mean max N_complete min_complete mean_complete max_complete
tchr1 56  75 92.85714 122         56           75      92.85714          122
Estimation algorithm:  MCMC      Cross-classified              Elapsed time : 7.78s
Number of obs:  5200 (from total 5200)          Number of iter.: 5000  Chains: 1  Burn-
```

Since this is using Baysian MCMC estimation, we also get the number of iterations and burn-in count. This is adjustable if you felt your model needed different parameters, but for this tutorial, I stuck with the default setting as shown.

We then get the DIC:

```
Bayesian Deviance Information Criterion (DIC)
Dbar      D(thetabar)    pD      DIC
56520.996  56473.520  47.479     56568.477
```

Again, since this is MCMC estimation, we do not get any likelihood ratios, nor can we perform any likelihood ration tests. This DIC is our baseline, to which other models including predictors will be compared to. If a predictor is useful at explaining variance, we expect the DIC to decrease. We can also use the DIC to determine if the multiple membership model is preferred to a single level model. The single model would be specified and run as follows:

```
intonlyred <- Math ~ 1 + (1|S_ID)
(MMembModelRed <- runMLwiN(Formula = intonlyred, data = StudData, estoptions = list(Es
```

```
## MLwiN is running, please wait......
```

```
##
## -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
## MLwiN (version: 2.36)  multilevel model (Normal)
## Estimation algorithm:  MCMC      Elapsed time : 3.13s
## Number of obs:  5200 (from total 5200)          Number of iter.: 5000  Chains: 1  Burn-in: 500
## Bayesian Deviance Information Criterion (DIC)
## Dbar      D(thetabar)    pD      DIC
## 56780.219  56778.250  1.968      56782.184
## -----------------------------------------------------------------------------------------------
## The model formula:
## Math ~ 1 + (1 | S_ID)
## Level 1: S_ID
## -----------------------------------------------------------------------------------------------
## The fixed part estimates:
##                 Coef.   Std. Err.       z   Pr(>|z|)       [95% Cred.   Interval]    ESS
## Intercept   441.95409     0.78136  565.62          0  ***   440.41947   443.46758   5000
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## -----------------------------------------------------------------------------------------------
## The random part estimates at the S_ID level:
##                      Coef.   Std. Err.   [95% Cred.    Interval]    ESS
## var_Intercept   3235.94230    63.15501   3116.37379   3364.41530   5000
## -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

From the output, we see that the DIC for the single level model is 56721.641, which is 200.645 higher than the multiple membership model. This indicates that the multiple membership model is statistically preferred over the single level model.

We next get reminded of our model formula, and what the grouping predictors were:

```
The model formula:
Math ~ 1 + (1 | tchr1) + (1 | S_ID)
Level 2: tchr1     Level 1: S_ID
```

Skipping down to "The fixed part estimates:"

```
The fixed part estimates:
                Coef.   Std. Err.       z   Pr(>|z|)       [95% Cred.   Interval]   ESS
Intercept   441.78207     2.38716  185.07          0  ***   436.82835   446.43202   346
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

we see that the intercept is 441.78. This is the overall mean of math achievement across all the students and all the teachers. This output also provides us with a 95% credibility interval of (436.83, 446.43). Keeping in mind that Bayesian techniques were used, we interpret this as there is a 95% probability that the true estimate of the intercept would lie within this range, given our observed data.

Going further down, we come to the random part estimates at level 2 and level 1, or the variance components:

```
The random part estimates at the tchr1 level:
                    Coef.   Std. Err.   [95% Cred.   Interval]    ESS
var_Intercept   281.90093    66.21423    176.86386   432.97710   2258
--------------------------------------------------------------------------------------
The random part estimates at the S_ID level:
                     Coef.   Std. Err.   [95% Cred.    Interval]    ESS
var_Intercept   3077.47965    59.92732   2960.59277   3194.83199   4679
```

Using these, we can calculate the variance partition coefficient (VPC) for this model. However, as previously mentioned in 4.4.2, using only the values in the table will only tell us the variance partition coefficient for students who only had a single teacher.

$$VPC_{u(2)} = \frac{\sigma^2_{u(2)}}{\sigma^2_{u(2)} + \sigma^2_e} = \frac{281.90}{3077.48 + 281.90} = 8.4\%$$

From this, we can say that for students that had a single teacher, 8.4% of the variation in their math scores is between teachers while the remaining 91.6% is between students. [1] This is also going to be the maximum amount of variation in student math scores that will be between teachers. As students have more teachers, less of the variation in math scores will be between teachers and more will be between students. Knowing this, predictors at level 1 will have the most impact, while level 2 predictors in this instance will have less of an impact.

5.3 Adding in level 1 predictors

We can add in level 1 predictors as predictors of student level variance in Math scores. Our data set has SES, gender, and SAT math scores as student-level predictors. Gender was dummy coded since there were three levels, with males being the reference for both NB and female. We use the same set up as for the intercept only model, only this time we add in our level 1 predictors. We do not need to adjust anything for the multiple membership part of the code, as that is staying the same.

Prior to adding in SES and SAT math, we will grand mean center them.

[1] This is why I believe the data could have been simulated better.

```
library(misty)
StudData$SESc <- center(StudData$S_SES, type = "CGM")
StudData$SATc <- center(StudData$SAT_M, type = "CGM")
```

This results in a final model, which can be written as

$$y_i = \beta_0 + \beta_1(SESc) + \beta_2(female) + \beta_3(NB) + \beta_4(SATc) + \sum_{j \in school(i)} w_{j,i}^{(2)} u_j^{(2)} + e_i$$

$$u_j^{(2)} \sim N(0, \sigma^2_{u(2)})$$

$$e_i \sim N(0, \sigma^2_e)$$

Putting the centered predictors into our model, we get the following input: (NOTE: I tried adding in random effects, but the model was unable to converge, so we are staying with fixed effects of SES and SAT only)

```
lev1 <- Math ~ 1 + SESc + female + NB + SATc + (1|tchr1) + (1|S_ID)

MultiMemb <- list(list(
  mmvar = list("tchr1", "tchr2", "tchr3", "tchr4", "tchr5", "tchr6", "tchr7", "tchr8", "tchr9", "
  weights = list("w1", "w2", "w3", "w4", "w5", "w6", "w7", "w8", "w9", "w10", "w11", "w12")), NA)

(MMembModel2 <- runMLwiN(Formula = lev1, data = StudData, estoptions = list(EstM = 1, drop.data =
```

```
## MLwiN is running, please wait......
```

```
##
## -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
## MLwiN (version: 2.36)  multilevel model (Normal)
##        N min     mean max N_complete min_complete mean_complete max_complete
## tchr1 56  75 92.85714 122         56           75      92.85714          122
## Estimation algorithm:  MCMC      Cross-classified              Elapsed time : 11.41s
## Number of obs:  5200 (from total 5200)          Number of iter.: 5000  Chains: 1  Burn-in: 500
## Bayesian Deviance Information Criterion (DIC)
## Dbar      D(thetabar)    pD      DIC
## 45953.879  45894.098  59.781     46013.660
## ----------------------------------------------------------------------------------------------
## The model formula:
## Math ~ 1 + SESc + female + NB + SATc + (1 | tchr1) + (1 | S_ID)
## Level 2: tchr1     Level 1: S_ID
## ----------------------------------------------------------------------------------------------
## The fixed part estimates:
##                 Coef.   Std. Err.       z    Pr(>|z|)       [95% Cred.   Interval]    ESS
## Intercept   442.34444     2.08061  212.60          0  ***   438.28115   446.29587
## SESc          0.30067     0.13997    2.15    0.03171  *       0.02828     0.58208
## female       -0.54715     0.58243   -0.94     0.3475          -1.67917     0.58191
## NB           -0.19573     1.44259   -0.14     0.8921          -2.99360     2.58687
## SATc          0.30025     0.00164  183.56          0  ***     0.29711     0.30344
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## -------------------------------------------------------------------------------------
## The random part estimates at the tchr1 level:
##                     Coef.   Std. Err.   [95% Cred.   Interval]    ESS
## var_Intercept   301.45458    60.80637    204.20305   440.78723   4371
## -------------------------------------------------------------------------------------
## The random part estimates at the S_ID level:
##                     Coef.   Std. Err.   [95% Cred.   Interval]    ESS
## var_Intercept   403.48601     7.96562    388.56632   419.39902   5000
## -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

Looking at the output for this model, we can first look at the fixed part, to see which predictors are significant and which are not:

```
The fixed part estimates:
                Coef.   Std. Err.        z     Pr(>|z|)        [95% Cred.   Interval]    E
Intercept   432.14999     2.13888   202.04            0  ***   428.14288   436.35112
SESc          0.19067     0.13910     1.37       0.1705         -0.08307     0.46388   46
female        0.07923     0.58179     0.14       0.8917         -1.05906     1.23982   50
NB           -5.54371     1.43551    -3.86    0.0001125  ***    -8.44713    -2.76427   50
SATc          0.29936     0.00164   182.35            0  ***     0.29614     0.30264   46
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From this, we see that SAT scores are significant, as is the NB gender code. The intercept represents that the average student math score across all teachers is 432.15. The slope was not allowed to randomly vary across clusters (see above), so all clusters have the same slope. SAT has a positive effect: for every point increase in SAT math scores, Math outcome scores are predicted to be 0.3 points higher, holding all else constant. Being nonbinary, however, has a negative effect when compared to males: they are predicted to score 5.5 points less than males, holding everything else constant. This is likely not due to math ability, but a number of other situational, structural, and social factors outside of the direct control of these students.

We can also see how well our model did overall by examining the DIC.

```
Bayesian Deviance Information Criterion (DIC)
Dbar      D(thetabar)    pD      DIC
45975.020  45915.676  59.346     46034.367
```

There was a decrease of over 10000 in the DIC between the intercept only model and this model, indicating that it is highly significant.

We can also examine the random effects, just for intercepts, and use this to calculate the percent reduction in variance at each level.

```
The random part estimates at the tchr1 level:
                    Coef.   Std. Err.   [95% Cred.   Interval]    ESS
var_Intercept   258.75691    52.28955    177.05400   377.15343   4283
------------------------------------------------------------------------------------------------
The random part estimates at the S_ID level:
                    Coef.   Std. Err.   [95% Cred.   Interval]    ESS
var_Intercept   404.72528     7.93413    389.22146   420.44359   5000
```

Here, the teacher-level variance is now interpreted as how much variance between teachers remains after accounting for the predictor variables. Similarly, the student-level variance is interpreted as how much variance between students remains after accounting for the predictor variables. As we only added in student-level predictors, we would most anticipate that the student-level variance would decrease. Calculating the percent reduction in student-level variance, we get

$$\frac{\sigma^2(intonly) - \sigma^2(level1)}{\sigma^2(intonly)} = \frac{3077.48 - 404.73}{3077.48} = 86.8\%$$

86.8% reduction in level 1 variance, which by most definitions is quite a lot!

We can also look at the level 2 variance reduction, though we are guessing it will not be much reduced at all.

$$\frac{\sigma^2(intonly) - \sigma^2(level1)}{\sigma^2(intonly)} = \frac{281.9 - 258.76}{281.9} = 8.2\%$$

We get that there was an 8.2% reduction in the level 2 variance. Keeping in mind that there wasn't much variance at level 2 to begin with, this is not very practically significant.

5.4 Adding in level 2 predictors

It is also possible to add in level 2 predictors to our model, to try and explain the (small) amount of teacher level variance. However, as this is a multiple membership model, and we are accounting for the fact that students may have encountered one or more teachers, we will incorporate weights into these predictors. In our data set, we have teacher experience as a teacher level predictor. Since both the values on the predictor and the weight are known, we can calculate a new predictor incorporating these items.

$$\overline{t\_exp_{2i}} = \sum_{j \in teacher(i)} w^{(2)}_{j,i} (t\_exp)^{(2)}_{2j}$$

```
StudData$tot_exp <- (StudData$t1_exp*StudData$w1 + StudData$t2_exp*StudData$w2 + StudDa
```

After calculating the weighted average of teacher experience, we can put it in our model, which will now be represented as the equation:

$$y_i = \beta_0+\beta_1(SESc)+\beta_2(female)+\beta_3(NB)+\beta_4(SATc)+\beta_5(\overline{t\_exp_{2i}})+\sum_{j\in school(i)} w^{(2)}_{j,i}u^{(2)}_j+e_i$$

```
lev2 <- Math ~ 1 + SESc + female + NB + SATc + tot_exp + (1|tchr1) + (1|S_ID)

MultiMemb <- list(list(
  mmvar = list("tchr1", "tchr2", "tchr3", "tchr4", "tchr5", "tchr6", "tchr7", "tchr8",
  weights = list("w1", "w2", "w3", "w4", "w5", "w6", "w7", "w8", "w9", "w10", "w11", "

(MMembModel2 <- runMLwiN(Formula = lev2, data = StudData, estoptions = list(EstM = 1,
```

```
## MLwiN is running, please wait......


##
## -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
## MLwiN (version: 2.36)  multilevel model (Normal)
##        N min     mean max N_complete min_complete mean_complete max_complete
## tchr1 56  75 92.85714 122         56           75      92.85714          122
## Estimation algorithm:  MCMC      Cross-classified              Elapsed time : 7s
## Number of obs:  5200 (from total 5200)          Number of iter.: 5000  Chains: 1  Bu
## Bayesian Deviance Information Criterion (DIC)
## Dbar      D(thetabar)    pD      DIC
## 45955.359  45895.484  59.875     46015.234
## -----------------------------------------------------------------------------------
## The model formula:
## Math ~ 1 + SESc + female + NB + SATc + tot_exp + (1 | tchr1) +
##     (1 | S_ID)
## Level 2: tchr1     Level 1: S_ID
## -----------------------------------------------------------------------------------
## The fixed part estimates:
##                 Coef.   Std. Err.       z     Pr(>|z|)       [95% Cred.   Interval]
## Intercept   416.34406    11.86173   35.10   6.796e-270  ***   392.63960   438.36022
## SESc          0.29967     0.13931    2.15      0.03147  *       0.02538     0.56744
## female       -0.53750     0.57763   -0.93       0.3521         -1.67365     0.58375
## NB           -0.26779     1.41534   -0.19       0.8499         -3.01926     2.50454
## SATc          0.30029     0.00161  186.27            0  ***     0.29717     0.30337
## tot_exp       2.48257     1.09498    2.27      0.02338  *       0.47155     4.77992
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## -----------------------------------------------------------------------------------
## The random part estimates at the tchr1 level:
##                     Coef.    Std. Err.    [95% Cred.    Interval]    ESS
## var_Intercept   276.54452    55.33169    185.96627   405.82281   3217
## ---------------------------------------------------------------------------------------------
## The random part estimates at the S_ID level:
##                     Coef.    Std. Err.    [95% Cred.    Interval]    ESS
## var_Intercept   403.45930     8.05699    388.21652   419.58182   5110
## -*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

Again looking at the output, we can start with the fixed effects, where we see that teacher experience is a significant predictor of math outcomes. The more experience a teacher has, the better their student math outcomes.

```
The fixed part estimates:
                Coef.    Std. Err.       z      Pr(>|z|)       [95% Cred.   Interval]    ESS
Intercept   416.34406    11.86173    35.10   6.796e-270  ***   392.63960   438.36022     74
SESc          0.29967     0.13931     2.15      0.03147  *       0.02538     0.56744   4690
female       -0.53750     0.57763    -0.93       0.3521         -1.67365     0.58375   5000
NB           -0.26779     1.41534    -0.19       0.8499         -3.01926     2.50454   5000
SATc          0.30029     0.00161   186.27            0  ***     0.29717     0.30337   5000
tot_exp       2.48257     1.09498     2.27      0.02338  *       0.47155     4.77992     78
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interestingly, once including teacher experience as a predictor, NB is no longer significant, but student SES is. However, looking at the ESS, or Effective Sample Size, we can see that it is very low for `tot_exp`, meaning that while 5000 draws were taken, they had as little information as might be expected in a sample of only 78 independent values, indicating a high level of autocorrelation [Leckie, 2013a].

Examining the model as a whole, we can compare the DIC of this model to the DIC of the model only containing level 1 predictors:

```
Bayesian Deviance Information Criterion (DIC)
Dbar      D(thetabar)    pD      DIC
45955.359  45895.484  59.875     46015.234
```

While not as an extreme lowering of the DIC as we saw in the level 1 model, the DIC is still 19.133 less. While there are no "hard" guidelines, it is generally accepted that a difference larger than 10 indicate that the model with the lower DIC should be used. So, in this case, the model incorporating the level 2 predictor is a better fit than the model without.

Lastly, we can look at the reduction in variance components by looking at the random parts (again, we held all slopes constant).

```
The random part estimates at the tchr1 level:
                    Coef.   Std. Err.   [95% Cred.   Interval]    ESS
var_Intercept   276.54452    55.33169    185.96627   405.82281   3217
--------------------------------------------------------------------------------------
The random part estimates at the S_ID level:
                    Coef.   Std. Err.   [95% Cred.   Interval]    ESS
var_Intercept   403.45930     8.05699    388.21652   419.58182   5110
```

Performing the same calculations as above, we see that student level variance decreased by an additional 0.3%. This would make sense, since no additional level 1 predictors were added to the model, so we were not anticipating a large decrease in student level variance. However, teacher level variance INCREASED by 6.87%, indicating that perhaps this is not a good predictor to be using.

5.5 Circling back to the research questions

At the outset of this fictional study, JMU said they had three research questions they would like to answer with these data and models.

5.5.1 RQ1: Do teacher characteristics influence student math scores? (ie, do we need to factor in nesting and multiple membership?)

We can answer this question with the intercept-only model. We saw that while there wasn't much variation at the teacher level, there was more than zero, indicating that including the nesting and the multiple membership structure is important. This was further verified by comparing the DIC of the intercept-only model to the single level model and finding that incorporating nesting caused a significant reduction of DIC (200).

5.5.2 RQ2: What student characteristics predict math scores?

This was answered when we added in level 1 predictors. We saw that adding in SES, SAT math, and gender dummy codes produced a significantly better model than the intercept only model. Of these predictors, SAT math and the nonbinary gender code were significant predictors of math scores. The model including these predictors resulted in an 86.8% reduction in student level variance, and a model that fit significantly better than the intercept only model.

5.5.3 RQ3: Does more teacher experience lead to better student outcomes?

Answering this is a bit more complicated. Looking at model fit and significance of teacher experience, we would conclude that as teacher experience increases, so too does student math outcomes. Additionally, this is a significant predictor, and leads to a better fitting model. However, inclusion of this predictor does not reduce teacher level variance and in fact increases it. There may be other, better, teacher level predictors that should be added to the model to create a better fit.

Chapter 6

Other uses of Multiple Membership Models

Multiple membership models can be combined with a number of other model types, and they are not restricted to only one level. It is possible to have multiple membership at level 2, but a pure hierarchy at level 3. Multiple membership models can be combined with a three level model to incorporate longitudinal data as well [for example Leroux, 2019]. They can also be combined with cross-classified models [eg. Browne et al., 2001, Beretvas [2011]] to handle even more complex data structures.

A use I found particularly interesting for multiple membership models was to handle missing data. Hill and Goldstein [1998] proposed using a multiple membership structure to equally weight all possible schools for a student when actual school enrollment was unknown. For students who had a known school membership, they were simply members of one school, with a weight of 1. I found this an interesting approach to missing data, as you are still able to use the student information, though with less school information than those students who have a known membership.

Appendix 1

```r
setwd("path\\goes\\here")
library(haven)
library(tidyverse)
library(R2MLwiN)
options(MLwiN_path="C:\\Program Files (x86)\\MLwiN trial\\i386\\mlwin.exe")

################################################################################
####Data Simulation and Saving out the .csv file for later analysis#############
################################################################################

#Simulate data for use in project; just need one data file - not a simulation study
set.seed(1)

#Number of level 1 units
N = 5200

#Student ID
S_ID <- 1:N

#SAT Math - as a "pre"
#Scores range from 200-800
sat <- c(200:800)
StudData$SAT_M <- sample(sat, N, replace = TRUE)

#Student gender
gender <- c(0, 1, 2)
S_gend <- sample(x = gender, size = N, replace = TRUE,
                 prob = c(0.56, 0.40, 0.04))

#Since have 3 codes, will need dummy coding
#Use 'male' as reference
StudData$female <- ifelse(StudData$S_gend == 0, 1, 0)
StudData$NB <- ifelse(StudData$S_gend == 2, 1, 0)
```

```
#Student SES
#Random generation - between 2 and 29; using the Kuppuswamy SES scale
S_SES <- rnorm(n = N, mean = 13, sd = 2)

#Number of different math teachers had by each student
#Treat as count data
tchrs <- c(1:12)
num_tchrs <- sample(x = tchrs, size = N, replace = TRUE,
                    prob = c(0.3325, 0.3325, 0.08, 0.08, 0.08, 0.08,
                             0.001, 0.001, 0.001, 0.001, 0.001, 0.01))

#Physics teachers (as an example of cross-classified data)
#Should work to make this better - match physics class with likelihood of STEM major
#based on math classes.
p <- c(1, 0)
phys <- sample(x = p, size = N, replace = TRUE,
               prob = c(0.75, 0.25))

#Join what we have so far in a dataframe
StudData <- data.frame(S_ID, S_gend, S_SES, num_tchrs, phys)

#22 physics teachers at JMU
StudData$phys_tchr <- ifelse(StudData$phys == 1, (StudData$phys_tchr = (sample(22, siz

#Teacher experience
#This gets added to the df later; kept as a variable for right now.
T_exp <- rnorm(n = 56, mean = 10, sd = 2)
T_ID <- c(1:56)

TData <- data.frame(T_ID, T_exp)

#Teachers - 56 of them to choose from
#Need to reference num_tchrs column.  Everyone has tchr1.
StudData$tchr1 = (sample(56, size = N, replace = TRUE))
StudData$tchr2 <- ifelse(num_tchrs >= 2, (StudData$tchr2 = (sample(56, size = N, repla
StudData$tchr3 <- ifelse(num_tchrs >= 3, (StudData$tchr3 = (sample(56, size = N, repla
StudData$tchr4 <- ifelse(num_tchrs >= 4, (StudData$tchr4 = (sample(56, size = N, repla
StudData$tchr5 <- ifelse(num_tchrs >= 5, (StudData$tchr5 = (sample(56, size = N, repla
StudData$tchr6 <- ifelse(num_tchrs >= 6, (StudData$tchr6 = (sample(56, size = N, repla
StudData$tchr7 <- ifelse(num_tchrs >= 7, (StudData$tchr7 = (sample(56, size = N, repla
StudData$tchr8 <- ifelse(num_tchrs >= 8, (StudData$tchr8 = (sample(56, size = N, repla
StudData$tchr9 <- ifelse(num_tchrs >= 9, (StudData$tchr9 = (sample(56, size = N, repla
StudData$tchr10 <- ifelse(num_tchrs >= 10, (StudData$tchr10 = (sample(56, size = N, re
StudData$tchr11 <- ifelse(num_tchrs >= 11, (StudData$tchr11 = (sample(56, size = N, re
StudData$tchr12 <- ifelse(num_tchrs >= 12, (StudData$tchr12 = (sample(56, size = N, re
#Need to clean out same numbers in one row - go through and check if values exist in any prior co
#do nothing.  If so, add one.  Not the most elegant, but will do for now.  Should fix to be bette
c <- select(StudData, c("tchr1", "tchr2", "tchr3", "tchr4", "tchr5", "tchr6",
                           "tchr7", "tchr8", "tchr9", "tchr10", "tchr11", "tchr12"))
c$tchr2 <- ifelse(c$tchr2 == c$tchr1, (c$tchr2 + 1), c$tchr2)
c$tchr3 <- ifelse((c$tchr3 != 0),
                  (ifelse((c$tchr3 == c$tchr2|c$tchr3 == c$tchr1), (c$tchr3 + 1), c$tchr3)), 0)
c$tchr4 <- ifelse((c$tchr4 != 0),
                   (ifelse((c$tchr4 == c$tchr3|c$tchr4 == c$tchr2|c$tchr4 == c$tchr1), (c$tchr4 +
c$tchr5 <- ifelse((c$tchr5 != 0),
                  (ifelse((c$tchr5 == c$tchr4|c$tchr5 == c$tchr3|c$tchr5 == c$tchr2|c$tchr5 == c$
c$tchr6 <- ifelse((c$tchr6 != 0),
                  (ifelse((c$tchr6 == c$tchr5|c$tchr6 == c$tchr4|c$tchr6 == c$tchr3|c$tchr6 == c$
c$tchr7 <- ifelse((c$tchr7 != 0),
                  (ifelse((c$tchr7 == c$tchr6|c$tchr7 == c$tchr5|c$tchr7 == c$tchr4|c$tchr7 == c$
c$tchr8 <- ifelse((c$tchr8 != 0),
                  (ifelse((c$tchr8 == c$tchr7|c$tchr8 == c$tchr6|c$tchr8 == c$tchr5|c$tchr8 == c$
c$tchr9 <- ifelse((c$tchr9 != 0),
                  (ifelse((c$tchr9 == c$tchr8|c$tchr9 == c$tchr7|c$tchr9 == c$tchr6|c$tchr9 == c$
c$tchr10 <- ifelse((c$tchr10 != 0),
                  (ifelse((c$tchr10 == c$tchr9|c$tchr10 == c$tchr8|c$tchr10 == c$tchr7|c$tchr10 =
c$tchr11 <- ifelse((c$tchr11 != 0),
                  (ifelse((c$tchr11 == c$tchr10|c$tchr11 == c$tchr9|c$tchr11 == c$tchr8|c$tchr11
c$tchr12 <- ifelse((c$tchr12 != 0),
                  (ifelse((c$tchr12 == c$tchr11|c$tchr12 == c$tchr10|c$tchr12 == c$tchr9|c$tchr12

#Had to go through a few iterations of the above to get MLwiN to accept


#Assigning variables back
StudData$tchr2 <- c$tchr2
StudData$tchr3 <- c$tchr3
StudData$tchr4 <- c$tchr4
StudData$tchr5 <- c$tchr5
StudData$tchr6 <- c$tchr6
StudData$tchr7 <- c$tchr7
StudData$tchr8 <- c$tchr8
StudData$tchr9 <- c$tchr9
StudData$tchr10 <- c$tchr10
StudData$tchr11 <- c$tchr11
StudData$tchr12 <- c$tchr12

#Weight
#Proportion of time had each teacher - need to reference num_tchrs column for this
Weight <- data.frame(num_tchrs = 1:12,
                     w1 = c(1, 0.5, 0.33, 0.25, 0.2, 0.166, 0.1428, 0.125, 0.11, 0.1,
                     w2 = c(0, 0.5, 0.33, 0.25, 0.2, 0.166, 0.1428, 0.125, 0.11, 0.1,
                     w3 = c(0, 0, 0.33, 0.25, 0.2, 0.166, 0.1428, 0.125, 0.11, 0.1, 0.
                     w4 = c(0, 0, 0, 0.25, 0.2, 0.166, 0.1428, 0.125, 0.11, 0.1, 0.09,
                     w5 = c(0, 0, 0, 0, 0.2, 0.166, 0.1428, 0.125, 0.11, 0.1, 0.09, 0.
                     w6 = c(0, 0, 0, 0, 0, 0.166, 0.1428, 0.125, 0.11, 0.1, 0.09, 0.08
                     w7 = c(0, 0, 0, 0, 0, 0, 0.1428, 0.125, 0.11, 0.1, 0.09, 0.083),
                     w8 = c(0, 0, 0, 0, 0, 0, 0, 0.125, 0.11, 0.1, 0.09, 0.083),
                     w9 = c(0, 0, 0, 0, 0, 0, 0, 0, 0.11, 0.1, 0.09, 0.083),
                     w10 = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0.1, 0.09, 0.083),
                     w11 = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.09, 0.083),
                     w12 = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.083))

#Merge the two dataframes to add in weights
StudData <- left_join(StudData, Weight, by = "num_tchrs")

#Make teacher experience variables - need to check if there's a teacher in each column
library(bruceR)
StudData$t1_exp <- LOOKUP(StudData, "tchr1", TData, "T_ID", "T_exp", return = "new.valu
StudData$t2_exp <- LOOKUP(StudData, "tchr2", TData, "T_ID", "T_exp", return = "new.valu
StudData$t3_exp <- LOOKUP(StudData, "tchr3", TData, "T_ID", "T_exp", return = "new.valu
StudData$t4_exp <- LOOKUP(StudData, "tchr4", TData, "T_ID", "T_exp", return = "new.valu
StudData$t5_exp <- LOOKUP(StudData, "tchr5", TData, "T_ID", "T_exp", return = "new.valu
StudData$t6_exp <- LOOKUP(StudData, "tchr6", TData, "T_ID", "T_exp", return = "new.valu
StudData$t7_exp <- LOOKUP(StudData, "tchr7", TData, "T_ID", "T_exp", return = "new.valu
StudData$t8_exp <- LOOKUP(StudData, "tchr8", TData, "T_ID", "T_exp", return = "new.valu
StudData$t9_exp <- LOOKUP(StudData, "tchr9", TData, "T_ID", "T_exp", return = "new.valu
StudData$t10_exp <- LOOKUP(StudData, "tchr10", TData, "T_ID", "T_exp", return = "new.va
StudData$t11_exp <- LOOKUP(StudData, "tchr11", TData, "T_ID", "T_exp", return = "new.va
StudData$t12_exp <- LOOKUP(StudData, "tchr12", TData, "T_ID", "T_exp", return = "new.va

#Replace NA from teacher experience columns with 0
StudData[is.na(StudData)] <- 0

#Make the outcome variables - for simulation
b0 <- 250
b1 <- .25
b2 <- .3
b3 <- -0.77
b4 <- -1.3
SAT_M <- StudData$SAT_M
S_SES <- StudData$S_SES
female <- StudData$female
NB <- StudData$NB
tchr1 <- StudData$tchr1
```

```
tchr2 <- StudData$tchr2
tchr3 <- StudData$tchr3
tchr4 <- StudData$tchr4
tchr5 <- StudData$tchr5
tchr6 <- StudData$tchr6
tchr7 <- StudData$tchr7
tchr8 <- StudData$tchr8
tchr9 <- StudData$tchr9
tchr10 <- StudData$tchr10
tchr11 <- StudData$tchr11
tchr12 <- StudData$tchr12
exp1 <- StudData$t1_exp
exp2 <- StudData$t2_exp
exp3 <- StudData$t3_exp
exp4 <- StudData$t4_exp
exp5 <- StudData$t5_exp
exp6 <- StudData$t6_exp
exp7 <- StudData$t7_exp
exp8 <- StudData$t8_exp
exp9 <- StudData$t9_exp
exp10 <- StudData$t10_exp
exp11 <- StudData$t11_exp
exp12 <- StudData$t12_exp
w1 <- StudData$w1
w2 <- StudData$w2
w3 <- StudData$w3
w4 <- StudData$w4
w5 <- StudData$w5
w6 <- StudData$w6
w7 <- StudData$w7
w8 <- StudData$w8
w9 <- StudData$w9
w10 <- StudData$w10
w11 <- StudData$w11
w12 <- StudData$w12

#Probably messed this up - doubtful I modeled the teacher weights correctly
Y <- b0 + b1*S_SES  + b2*SAT_M + b3*female + b4*NB + (tchr1*w1 + tchr2*w2 + tchr3*w3 + tchr4*w4 +
  (exp1*w1 + exp2*w2 + exp3*w3 + exp4*w4 + exp5*w5 + exp6*w6 + exp7*w7 + exp8*w8 + exp9*w9 + exp1
StudData$Math <- Y

#See what outcome variable looks like
ggplot(data = StudData) +
  geom_histogram(aes(Math))
```

```
#Save out the file containing the outcome variable and others.
write.csv(StudData, "exampledata2.csv")

#Checking full model function; no level 2 predictors yet
model1 <- Math ~ 1 + S_SES + female + NB + SAT_M + (1|tchr1) + (1|S_ID)


MultiMemb <- list(list(
  mmvar = list("tchr1", "tchr2", "tchr3", "tchr4", "tchr5", "tchr6", "tchr7", "tchr8",
  weights = list("w1", "w2", "w3", "w4", "w5", "w6", "w7", "w8", "w9", "w10", "w11", "

(MMembModel <- runMLwiN(Formula = model1, data = StudData, estoptions = list(EstM = 1,
```

Bibliography

URL http://www.bristol.ac.uk/cmm/learning/online-course.

URL https://www.multibugs.org/.

JJ Allaire, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. *rmarkdown: Dynamic Documents for R*, 2022. URL https://CRAN.R-project.org/package=rmarkdown. R package version 2.14.

S. Natasha Beretvas. *Cross-classified and Multiple-membership models*, pages 313–334. Routledge Taylor and Francis Group, 2011.

William J. Browne, Harvey Goldstein, and Jon Rasbash. Multiple membership multiple classification. *Statistical Modelling*, 1:103–124, 2001.

W.J. Browne. *MCMC Estimation in MLwiN v3.06.* Centre for Multilevel Modelling, University of Bristol.

C. Charlton, J. Rasbash, W.J. Browne, M. Healy, and B. Cameron. *MLwiN Version 3.06.* Centre for Multilevel Modelling, University of Bristol.

Hyewon Chung and S. Natasha Beretvas. The impact of ignoring multiple membership data structures in multilevel models. *British Journal of Mathematical and Statistical Psychology*, 65:185–200, 2012.

Craig K. Enders. *Centering predictors and contextual effects*, pages 89–108. SAGE Publications Inc., 2013. ISBN 9780857025647. doi: 10.4135/9781446247600.n6.

Krisztina Gero, Hiroyuki Hikichi, Jun Aida, Katsunori Kondo, and Ichiro Kawachi. Associations between community social capital and preservation of functional capacity in the aftermath of a major disaster. *American Journal of Epidemiology*, 189:1369–1378, 2020.

R. J. B. Goudie, R. M. Turner, D. De Angelis, and A. Thomas. Multibugs: A parallel implementation of the bugs modelling framework for faster bayesian inference. *Journal of Statistical Software*, 95, 2020.

Ronald H. Heck, Tingting Reid, and George Leckie. Incorporating student mobility in studying academic growth in math: comparing several alternative multilevel formulations. *School Effectiveness and School Improvement*, 33: 516–543, 2022.

Peter Hill and Harvey Goldstein. Multilevel modeling of educational data with cross-classification and missing identification for units. *Journal of Educational and Behavioral Statistics*, 23:117–128, 1998.

George Leckie. *Multiple Membership Multilevel Models - Concepts.* LEMMA VLE Module 13, p. 1-57, 2013a.

George Leckie. *Cross-classified Multilevel Models - Stata Practical.* LEMMA VLE Module 12, 2013b.

Audrey J. Leroux. Student mobility in multilevel growth modeling: a multiple membership piecewise growth model. *The Journal of Experimental Education*, 87:430–448, 2019.

Xin Ma, Yin Zhang, and Xingkai Luo. Students' and teachers' critical thinking in science education: are they related to each other and with physics achievement? *Research in Science and Technological Education*, 2021. doi: 10.1080/02635143.2021.1944078.

Lindsey J. Wolff Smith and S. Natasha Beretvas. A comparison of techniques for handling and assessing the influence of mobility on student achievement. *The Journal of Experimental Education*, 85:3–23, 2017.

T.A.B. Snijders and R.J. Bosker. *The Random Intercept Model*, pages 41–73. SAGE Publications Inc, 2012.

Hadley Wickham. *tidyverse: Easily Install and Load the Tidyverse*, 2021. URL https://CRAN.R-project.org/package=tidyverse. R package version 1.3.1.

Hadley Wickham, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*, 2022. URL https://CRAN.R-project.org/package=ggplot2. R package version 3.4.0.

Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2022. URL https://yihui.org/knitr/. R package version 1.39.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2023. URL https://CRAN.R-project.org/package=bookdown. R package version 0.33.

Takuya Yanagida. *misty: Miscellaneous Functions T. Yanagida*, 2023. URL https://CRAN.R-project.org/package=misty. R package version 0.4.7.

Zhengzheng Zhang, Chris Charlton, Richard Parker, George Leckie, and William Browne. *R2MLwiN: Running MLwiN from Within R*, 2023. URL https://CRAN.R-project.org/package=R2MLwiN. R package version 0.8-8.