R for Multilevel Models

Sunthud Pornprasertmanit

University of Kansas

Table of Contents

Updated: 4/4/2013

R Basics

Install Packages

In this section, the basic R commands that are useful for understanding a multilevel model in R are covered. First, we will need two main packages for multilevel models: `lme4` (Bates, Maechler, & Bolker, 2012) and `nlme` (Pinheiro et al., 2013). If those packages have not been installed, the packages can be installed:

```
install.packages(c("lme4", "nlme"))
```

The `install.packages` function is used to install a package into a hard drive. Normally, the name of a single package is listed in the function, such as `install.packages("lme4")`. If more than one package is needed, users can use the `c` function to concatenate two packages together. The `lme4` package is mainly used. The `nlme` will be used for some techniques that require modeling error structures. If the packages are installed, the packages are still not available in the R program. The `install.packages` function is similar to keep the packages in a library. To use it, one should bring a desired package on the table by using the `library` function:

```
library(lme4)
```

Interaction with R

In R, there are two ways to run a program. First, users may type a command to R console (the windows where the typing line has >). This approach is good for a quick command that users are not interested to save it for later use. Second, users can type commands in blank text file with `.R` extension. In the original R program, users can go to `File`… → `New Script` to open a blank page. Users can type the command and use `Ctrl + R` for Windows or `Apple + Return` for Mac to execute the command. If users write a script, users can comment commands by using a pound sign, `#`. The script can be saved in a file for later use. In running a multilevel model, using R script is highly recommended.

Furthermore, R is case-sensitive. The function or object names in lowercase and uppercase can have different meanings. For example, there are functions called `anova` and `Anova` that means different things. Therefore, please be aware of the big and small cases.

Reading Data Files

Next, let's import a data set into R program. The target data set is `mlbook2_r.dat` from Snijders and Bosker (2011). If the file is opened by basic text editor file, such as `Notepad` or `TextEdit`, the file contains with the first line of variable names and the other lines of data for each case. Each observation is separated by white spaces (i.e., tabs or blanks). Then, the `read.table` function can be used to import data:

```
dat <- read.table("C:/Users/student/Desktop/mlbook2_r.dat", header = TRUE)
```

In this line, `dat` is the object name that users wish to save the data to and the arrow, `<-`, means "is assigned as". Thus, this line means to read a data file and save into the object named `dat`. In the `read.table` function, the first argument is the data file directory. Note that backslash is not allowed in writing a directory in R. Users need to use forward slash, `/`, or double backslash instead, `\\`. The second argument, `header`, is a logical whether the first line contains variable names. If so, the argument is specified as `TRUE`. Otherwise, specify as `FALSE`. Users may use `T` and `F` as acronyms for `TRUE` and `FALSE`. By default, this function will separate observations by white spaces.

Instead of specifying a data file directory, users may run the following commands:

```
dat <- read.table(file.choose(), header = TRUE)
```

R will provide a pop-up window to choose the directory of a data file.

Sometimes, observations in a data set are separated by commas. Users may run a following line to read data file with comma separated values.

```
dat2 <- read.table("C:/Users/student/Desktop/mlbook2_r.csv", header = TRUE, sep = ",")
```

The `sep` argument represents the character used in separating observations. Users may check the `read.csv` function for reading the comma-separated-values data.

Once the data is saved in R workspace, the data can be viewed by just typing its name.

```
dat
```

Rather than viewing the whole data set, the `head` or `tail` functions can be used to view a several first or a several last cases:

```
head(dat)

tail(dat)
```

Descriptive Statistics

To investigate a summary of descriptive statistics from a data set, the `summary` function can be used on the data object:

```
summary(dat)
```

```
    schoolnr       pupilNR_new      langPOST          ses                 IQ_verb              sex
 Min.   :  1.0   Min.   :   3   Min.   : 8.00   Min.   :-17.73000   Min.   :-7.87000   Min.   :0.0000
 1st Qu.: 69.0   1st Qu.:1137   1st Qu.:36.00   1st Qu.: -7.73000   1st Qu.:-0.87000   1st Qu.:0.0000
 Median :136.0   Median :2210   Median :42.00   Median : -1.73000   Median : 0.13000   Median :0.0000
 Mean   :132.3   Mean   :2174   Mean   :41.41   Mean   :  0.04834   Mean   : 0.04418   Mean   :0.4872
 3rd Qu.:189.0   3rd Qu.:3214   3rd Qu.:48.00   3rd Qu.:  9.27000   3rd Qu.: 1.13000   3rd Qu.:1.0000
 Max.   :259.0   Max.   :4214   Max.   :58.00   Max.   : 22.27000   Max.   : 6.63000   Max.   :1.0000
    Minority         denomina        sch_ses              sch_iqv            sch_min
 Min.   :0.0000   Min.   :1.000   Min.   :-17.72700   Min.   :-4.81130   Min.   :0.00000
 1st Qu.:0.0000   1st Qu.:1.000   1st Qu.: -4.38100   1st Qu.:-0.36680   1st Qu.:0.00000
 Median :0.0000   Median :2.000   Median :  0.13000   Median : 0.09320   Median :0.00000
 Mean   :0.0471   Mean   :2.254   Mean   :  0.02444   Mean   : 0.01574   Mean   :0.05169
 3rd Qu.:0.0000   3rd Qu.:3.000   3rd Qu.:  4.43900   3rd Qu.: 0.46650   3rd Qu.:0.05900
 Max.   :1.0000   Max.   :5.000   Max.   : 15.54500   Max.   : 2.47690   Max.   :0.92000
```

From the data set, there are two ways to select a column from a data set. First, the dollar sign, `$`, can be used:

```
dat$ses
```

The second approach is to use a square bracket to select a desired element of a data set:

```
dat[ , "ses"]
```

The data object has two dimensions: rows and columns. In the square bracket, two elements mean index of rows and index of columns. Because the first index is blank, all rows are selected. Because the second index is `"ses"`, the column with the name of `ses` is selected.

The mean, standard deviation, minimum, maximum, or variance can be computed from the target variable by the `mean`, `sd`, `min`, `max`, and `var` functions, respectively:

```
mean(dat$ses)
```

```
[1] 0.04833954
```

```
sd(dat$ses)
```

```
[1] 10.89977
```

```
min(dat$ses)
```

```
[1] -17.73
```

```
max(dat$ses)
```

```
[1] 22.27
```

```
var(dat$ses)
```

```
[1] 118.805
```

Sometimes, a descriptive statistic of all variables in a data set is needed. The `apply` function can be used to apply a descriptive statistic to all row vectors or all column vectors.

```
apply(dat, 2, mean)
```

```
    schoolnr  pupilNR_new     langPOST          ses      IQ_verb          sex     Minority     denomina 
1.322866e+02 2.174353e+03 4.141299e+01 4.833954e-02 4.418308e-02 4.872272e-01 4.709952e-02 2.254125e+00 
     sch_ses      sch_iqv      sch_min 
2.444279e-02 1.573949e-02 5.168999e-02 
```

The first argument is the target data set. The second argument is the dimension to separate data into vectors: 1 = separate data by rows and 2 = separate data by columns. The third argument is the function to be applied on the separated vectors. Thus, this code means to separate data into different vectors based on columns and apply the `mean` function into each vector.

Note that the output provides the result in scientific formula. Scientific formula *A*e+*B* is equivalent to $A \times 10^B$. For example, 1.322866e+02 $= 1.322866 \times 10^2 = 132.3866$.

As another example, the standard deviation of each column (variable) can be computed:

```
apply(dat, 2, sd)
```

```
    schoolnr  pupilNR_new     langPOST          ses      IQ_verb          sex     Minority     denomina 
  70.4121415 1198.8182350    8.8930451   10.8997703    2.0407464    0.4999033    0.2118799    1.1072764 
     sch_ses      sch_iqv      sch_min 
   6.1485393    0.8177883    0.1243000 
```

Sometimes, the descriptive statistics of each group are of interest. The `aggregate` function can be used:

```
aggregate(ses ~ sex, dat, mean)
```

```
  sex        ses
1   0 -0.2764453
2   1  0.3901529
```

For the `sex` variable, the female group is coded as 1 and male group is coded as 0. The first argument of the `aggregate` function is a formula. The dependent variable is listed on the left hand side of the tilde, ~, and the independent variable is listed on the right hand side of the tilde. In this case, the descriptive statistics of `ses` are separated by `sex`. The second argument is the target data set. The third argument is the target function, which is `mean`.

To find a correlation, the `cor` function is used. However, not all variables are appropriate in finding correlation, such as `schoolnr`, which is school ID, or `pupulNR-new`, which is student ID. Thus, only three variables are selected to find correlation, `langPOST`, `IQ_verb`, and `ses`, then the `cor` function is applied:

```
subsetdat <- dat[,c("langPOST", "IQ_verb", "ses")]

cor(subsetdat)
```

```
          langPOST   IQ_verb       ses
langPOST 1.0000000 0.6084031 0.3674754
IQ_verb  0.6084031 1.0000000 0.3258190
ses      0.3674754 0.3258190 1.0000000
```

The `subsetdat` saves the subset of the target data set. Because three variables are selected, the `c` function is used to concatenate variable names.

Multiple Regression

Let's run a multiple regression on the target data set. For example, language scores (`langPOST`) are predicted by socioeconomic status (`ses`) and IQ-verbal score (`IQ_verb`). The `lm` function, which stands for linear model, can be used:

```
out <- lm(langPOST ~ ses + IQ_verb, data = dat)

summary(out)
```

```
Call:
lm(formula = langPOST ~ ses + IQ_verb, data = dat)

Residuals:
     Min       1Q   Median       3Q      Max 
-30.0784  -4.3549   0.5034   4.8496  25.2241 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 41.30026    0.11222  368.03   <2e-16 ***
ses          0.15449    0.01089   14.19   <2e-16 ***
IQ_verb      2.38242    0.05816   40.97   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.878 on 3755 degrees of freedom
Multiple R-squared: 0.4022,     Adjusted R-squared: 0.4019 
F-statistic:  1263 on 2 and 3755 DF,  p-value: < 2.2e-16
```

For the `lm` function, the first argument is a formula. Similar to the `aggregate` function, the left hand side of the tilde is a dependent variable and the right hand side is the list of independent variables. Different independent variables in a formula are separated by a plus sign, +. This formula can be interpreted as "`longPOST` is predicted by `ses` and `IQ_verb`". The second argument, `data`, is the target data set that contains the variables listed in the formula.

Unlike other statistical packages, the analysis result is saved in an object. The `summary` function can be used to provide the output. The output is separated into four parts:

1. `Call`. The code that users used to build the output
2. `Residuals`. The residual statistics including minimum, the first quartile, median, the third quartile, and maximum.
3. `Coefficients`. The table of regression coefficient. The intercept is listed as the first line. The following lines are the regression coefficients of `ses` and `IQ_verb`. All rows provide the regression coefficient values, standard errors, *t*-statistics, and *p*-values.
4. `Others`. The first row provides the standard error of the estimate and the degree of freedom. The second row provides the coefficient of determination (R-squared) and its adjusted value. The third row provides the *F* statistic and its *p* value for testing whether the coefficient of determination is significantly different from 0.

Note that the assumption of uncorrelated errors is violated here because the data come from intact groups (schools). Multilevel model will be shown later.

Not only the `summary` function can be run on the output, other functions can be applied on the output.

1. `coef`. Request the regression coefficient in a model

```
coef(out)
```

```
(Intercept)         ses     IQ_verb 
 41.3002551   0.1544867   2.3824221
```

2. `confint`. Request the confidence interval for each regression coefficient

```
confint(out)
```

```
                 2.5 %    97.5 %
(Intercept) 41.0802377 41.520272
ses          0.1331384  0.175835
IQ_verb      2.2683992  2.496445
```

3. `vcov`. Request the (asymptotic) variance-covariance matrix of a parameter estimate. This matrix represents the expected variance and covariance of statistics assuming that multiple random sampling can be drawn. The `vcov` matrix is useful for probing interaction. Note that the square roots of the diagonal elements are standard errors of regression coefficients.

```
vcov(out)
```

```
              (Intercept)           ses       IQ_verb
(Intercept)  1.259325e-02  3.384850e-06 -0.0001394652
ses          3.384850e-06  1.185635e-04 -0.0002063269
IQ_verb     -1.394652e-04 -2.063269e-04  0.0033822660
```

4. `residuals`. Request the residual values of each case.

```
residuals(out)
```

```
         1          2          3          4          5          6 
-2.0265141  0.1730243 -0.6872987  7.5031743 -9.3412320 -2.9148651 …
```

5. `predict`. Request the predicted scores of each case.

```
predict(out)
```

```
       1        2        3        4        5        6 
48.02651 44.82698 33.68730 38.49683 29.34123 32.91487 …
```

If categorical variable is used as a predictor in a regression model, the categorical variable needs to be transformed as dummy-coded variables (or other types of coding, such as effect coding or contrast coding). For example, `sex` is a categorical variable. Users need to make sure that `sex` is in a valid format. In this case, females are coded as 1 and males are coded as 0 already. Thus, this variable is good.

Let's do some practice on dummy coding on the `airquality` data, which has been provided in R already:

```
head(airquality)
```

```
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
```

In this example, the `Ozone` variable is predicted by `Temp` and `Month`. `Month` is a categorical variable with five categories. Thus, four dummy variables are needed. Let's use `Month = 5` as the reference group so four dummy variables can be made:

```
m6 <- airquality$Month == 6

m7 <- airquality$Month == 7

m8 <- airquality$Month == 8

m9 <- airquality$Month == 9

airquality2 <- data.frame(airquality, m6, m7, m8, m9)

tail(airquality2)
```

```
    Ozone Solar.R Wind Temp Month Day    m6    m7    m8   m9
148    14      20 16.6   63     9  25 FALSE FALSE FALSE TRUE
149    30     193  6.9   70     9  26 FALSE FALSE FALSE TRUE
150    NA     145 13.2   77     9  27 FALSE FALSE FALSE TRUE
151    14     191 14.3   75     9  28 FALSE FALSE FALSE TRUE
152    18     131  8.0   76     9  29 FALSE FALSE FALSE TRUE
153    20     223 11.5   68     9  30 FALSE FALSE FALSE TRUE
```

The double equal signs, ==, are used to evaluate whether the `Month` variable has a specific value. The results are provided as `TRUE` and `FALSE`, which R also understands as `1` and `0`, respectively. The `data.frame` function is used to combine data and vectors into a single data set.

Then, the `lm` function can be applied:

```
out2 <- lm(Ozone ~ Temp + m6 + m7 + m8 + m9, data = airquality2)

summary(out2)
```

```
Call:
lm(formula = Ozone ~ Temp + m6 + m7 + m8 + m9, data = airquality2)

Residuals:
   Min     1Q Median     3Q    Max 
-42.95 -13.86  -2.05  12.13 116.05 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -156.8309    21.7054  -7.225 6.94e-11 ***
```

```
Temp            2.7041     0.3182   8.498 1.05e-13 ***
m6TRUE        -25.2449     9.5883  -2.633  0.00968 **
m7TRUE        -10.8856     8.3786  -1.299  0.19659
m8TRUE        -10.2475     8.3946  -1.221  0.22480
m9TRUE        -19.6563     6.9842  -2.814  0.00579 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22.92 on 110 degrees of freedom
  (37 observations deleted due to missingness)
Multiple R-squared: 0.5383,     Adjusted R-squared: 0.5173
F-statistic: 25.65 on 5 and 110 DF,  p-value: < 2.2e-16
```

The regression coefficients of the dummy variables are shown in the output. Alternatively, the `Month` variable can be transformed into the factor format, which R understands as categorical variable and R will transform to dummy variables automatically.

```
airquality$Month <- factor(airquality$Month, labels=c("May", "Jun", "Jul", "Aug", "Sep"))

out3 <- lm(Ozone ~ Temp + Month, data = airquality)

summary(out3)
```

```
Call:
lm(formula = Ozone ~ Temp + Month, data = airquality)

Residuals:
   Min     1Q Median     3Q    Max
-42.95 -13.86  -2.05  12.13 116.05

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -156.8309    21.7054  -7.225 6.94e-11 ***
Temp           2.7041     0.3182   8.498 1.05e-13 ***
MonthJun     -25.2449     9.5883  -2.633  0.00968 **
MonthJul     -10.8856     8.3786  -1.299  0.19659
MonthAug     -10.2475     8.3946  -1.221  0.22480
MonthSep     -19.6563     6.9842  -2.814  0.00579 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22.92 on 110 degrees of freedom
  (37 observations deleted due to missingness)
Multiple R-squared: 0.5383,     Adjusted R-squared: 0.5173
F-statistic: 25.65 on 5 and 110 DF,  p-value: < 2.2e-16
```

The `factor` function is used to change the variable format. The `labels` argument is used to put the label for each group. Then the `Month` variable can be put into the `lm` formula directly. The reference group is always the first group listed in the `labels` argument.

Multilevel Regression Basics

The script for running multilevel regression is similar to the script in multiple regression. Users just need to aware of the different roles of their variables. In multiple regression, there are two types of variables: independent variables and dependent variable. In multilevel regression, however, there are at least four types of variables: dependent variable, level-1 (L1) independent variable, level-2 (L2) independent variable, and L2 ID (the variable classifying each case into different groups).

Different software packages require different types of data. For example, HLM can use two data sets: one for L1 and another for L2. In the `lme4` and `nlme` packages in R, the data set must be in a long format. That is, all L1 and L2 variables are in the same data set where rows represent L1 units. One variable is used as L2 ID. The L2 independent variables must have the same values for the same L2 units.

Student ID	School ID	DV	L1 IV	L2 IV
1	**1**	5	4	4
2	**1**	6	1	4
3	**1**	2	2	4
4	**1**	3	6	4
5	**2**	8	8	8
6	**2**	9	9	8
7	**2**	5	4	8
8	**2**	4	2	8
9	**3**	1	3	6
10	**3**	7	4	6
11	**3**	5	5	6
12	**3**	3	7	6

Notice that the L2 IV values in the same schools are the same. If users have data in the different format, they need to transform it into the appropriate format.

Model 0: Null Model

Let's run a very basic model. The language scores (`langPOST`) is used as a dependent variable. In this data set, students are nested in different schools by school ID (`schoolnr`). The null model would be

L1	$Y_{ij} = \beta_{0j} + e_{ij}$	$e_{ij} \sim N(0, \sigma^2)$
L2	$\beta_{0j} = \gamma_{00} + u_{0j}$	$u_{0j} \sim N(0, \tau_{00})$

These notations represent

- Y_{ij} = The language score of Student *i* in School *j*
- β_{0j} = The average language score within School *j*
- γ_{00} = The average language score across all schools
- e_{ij} = The deviation of the language score of Student *i* from the School *j* mean
- u_{0j} = The deviation of the language score of School *j* mean from the grand mean
- σ^2 = The language score variance within schools (L1 variance)
- τ_{00} = The language score variance across schools (L2 variance)

From this model, the `lme4` package can be used to run the model by the `lmer` function:

```
library(lme4)

m0 <- lmer(langPOST ~ 1 + (1|schoolnr), data = dat, REML=FALSE)

summary(m0)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + (1 | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 26601 26620 -13298    26595   26596
Random effects:
 Groups   Name        Variance Std.Dev.
 schoolnr (Intercept) 18.125   4.2574
 Residual             62.851   7.9278
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
            Estimate Std. Error t value
```

```
(Intercept)  41.0046     0.3249   126.2
```

In the `lmer` function, the first argument is formula. The dependent variable is separated from the independent variable by the tilde. The number `1` after the tilde is used to represent intercept, which can be viewed as an independent variable with a constant of 1. The addition notation is the random effect, `(1|schoolnr)`. This random effect means the intercept (`1`) is random across school (`schoolnr`). Users may find the mapping from the formula and the reduced-form equation easily.

```
langPOST ~ 1 + (1|schoolnr)
```

$$Y_{ij} = \underbrace{\gamma_{00}(1)}_{\text{Fixed Effect}} + \underbrace{u_{0j}(1)}_{\text{Random Effect}} + e_{ij}$$

The second argument, `data`, is the target data set. The third argument, `REML`, is to choose whether Residual Maximum Likelihood (REML) is used. If `FALSE`, Full-Information Maximum Likelihood (FIML) is used. The advantages and disadvantages of both methods are not discussed here (see Snijders & Bosker, 2011 for further details).

Similar to multiple regression, the output is saved into an object and the `summary` function is used to get the output. The output is separated into three parts:

1. `Model Description and Model Fit Statistics`. The model formula and the target data are described. Further, AIC, BIC, log-likelihood ratio (`logLik`), deviance, and deviance from REML (`REMLdev`) are provided as model fit statistics.
2. `Random Effects`. The variance and standard deviation of random effects. In this model, the variance at school level $\left(Var(u_{0j})\text{ or }\tau_{00}\right)$ and the variance at student level $\left(Var(e_{ij})\text{ or }\sigma^2\right)$ are listed, as well as their standard deviations.
3. `Fixed Effects`. The regression coefficients, standard errors, and *t*-statistics are provided. In this case, the grand mean, γ_{00}, is provided.

There are no two important pieces of information here: *p*-value and intraclass correlation. Users may need a little program practice here. The *p*-value can be approximated by normal distribution—assuming that sample size in L2 is large (says > 30; see Snijders and Bosker, 2011, for a better approximation by *t* distribution). There are several steps to calculate the *p*-value[1]:

1. The summary of the multilevel output can be saved as an object

```
out0 <- summary(m0)
```

2. Use the `coef` function on the summary object to extract the fixed-effect table. Save the fixed-effect table:

```
coef0 <- coef(out0)

coef0
```

```
            Estimate Std. Error  t value
```

[1] Check http://finzi.psych.upenn.edu/R/Rhelp02a/archive/76742.html for the reasons why Douglas Bates did not include the *p* value in the `lme4` package.

```
(Intercept)  41.0046   0.3248646 126.2206
```

3. The *t* values are saved as a vector

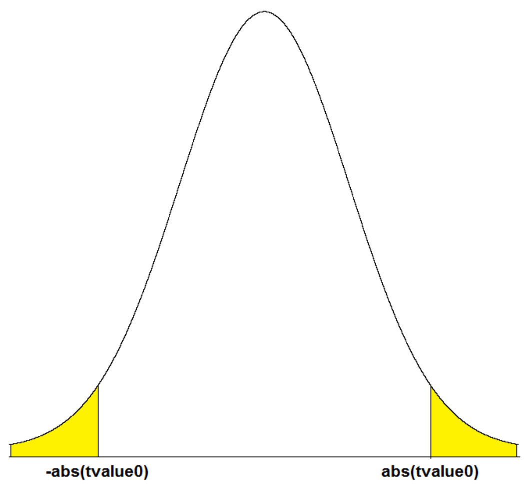
```
tvalue0 <- coef0[,"t value"]
```

4. Use the `pnorm` function to approximate the area under the normal distribution using the *t* value:

```
pnorm(abs(tvalue0), lower.tail=FALSE) * 2
```

```
[1] 0
```

The `abs` function is the absolute value function to get rid of the negative sign (if any). Then, the area over the absolute of the *t*-value under the normal distribution is calculated—the `lower.tail` argument is `FALSE` so the calculation is based on the upper tail. The resulting area is multiplied by 2 to take into account both the left and right extremes. The figure below shows how the `pnorm` function works:



The resulting *p* value is approximately 0 (report it as $p < .001$), which is congruent with very high *t* value.

The intraclass correlation can be computed by the following steps:

1. Save the summary of the multilevel output.
2. Put `@REmat` after the summary output to get the random effect matrix

```
ranef0 <- out0@REmat

ranef0
```

```
 Groups     Name          Variance Std.Dev.
 "schoolnr" "(Intercept)" "18.125" "4.2574"
 "Residual" ""            "62.851" "7.9278"
```

3. Extract appropriate values for τ_{00} and σ^2. Use the `as.numeric` function to change the string format to number:

```
tau00 <- as.numeric(ranef0[1, 3])

sigma2 <- as.numeric(ranef0[2, 3])
```

4. Compute intraclass correlation, $\rho = \tau_{00}/(\tau_{00} + \sigma^2)$:

```
icc <- tau00/(tau00 + sigma2)

icc
```

```
[1] 0.2238318
```

You may calculate the intraclass correlation by using R as a fancy calculator:

```
18.125 / (18.125 + 62.851)
```

```
[1] 0.2238318
```

Model 1: Analysis of Covariance Model

In this model, a L1 predictor is put in the model with fixed slope. For example, the language scores (`langPOST`) is used as a dependent variable. The L1 predictor is verbal IQ score (`IQ_verb`). The analysis of covariance (ANCOVA) model would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + u_{0j} \qquad u_{0j} \sim N(0, \tau_{00})$$
$$\beta_{1j} = \gamma_{10}$$

These notations should represent (the blue lines indicate that the meanings changed from Model 0)

- Y_{ij} = The language score of Student *i* in School *j*
- X_{ij} = The verbal IQ score of Student *i* in School *j*
- β_{0j} = The expected average of language score within School *j* when the verbal IQ score is 0, which is also referred to as *adjusted mean*
- β_{1j} = The expected change in language score when the verbal IQ score of students in School *j* increases by 1. In this case, the expected changes across schools are the same.
- γ_{00} = The expected average language score across all schools when the verbal IQ score is 0.
- γ_{10} = The schools' average expected change in language score when the verbal IQ score increase by 1.
- e_{ij} = The difference between the actual language score and the predicted language score of Student *i* in School *j*
- u_{0j} = The deviation of the adjusted language score average of School *j* (when the verbal IQ score is 0) from the grand mean of adjusted average across schools
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the verbal IQ score
- τ_{00} = The language score residual variance across schools (L2 residual variance) controlling for the verbal IQ score

The ANCOVA model can be run by the `lmer` function:

```
m1 <- lmer(langPOST ~ 1 + IQ_verb + (1|schoolnr), data = dat, REML=FALSE)

summary(m1)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + (1 | schoolnr)
```

```
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24920 24945 -12456    24912   24917
Random effects:
 Groups   Name        Variance Std.Dev.
 schoolnr (Intercept)  9.8451  3.1377
 Residual             40.4689  6.3615
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
            Estimate Std. Error t value
(Intercept) 41.05490    0.24336  168.70
IQ_verb      2.50745    0.05438   46.11

Correlation of Fixed Effects:
        (Intr)
IQ_verb 0.003
```

The addition notation in the formula is the fixed effect of verbal IQ. The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + IQ_verb + (1|schoolnr)
```

$$Y_{ij} = \underbrace{\gamma_{00}(1) + \gamma_{10}X_{ij}}_{\text{Fixed Effect}} + \underbrace{u_{0j}(1) + e_{ij}}_{\text{Random Effect}}$$

The output is similar to the null model. The model fit statistics are different from the null model. You may notice that the AIC and BIC are lower for the ANCOVA model, which means that the ANCOVA model fit better. Deviances between two models can be compared together by the deviance test, which will be described later.

The variances of the random effects are different from those in the null model because the meanings of the random effects between two models are different. τ_{00} and σ^2 in this model represents residual variances (rather than total variances in the null model).

The output of fixed effect has two rows, which represent intercept (γ_{00}) and the effect of verbal IQ (γ_{10}). The later part of the output is the correlation of the fixed effects. The correlation is actually the asymptotic variance-covariance matrix of regression coefficient (`vcov`) that is transformed into a correlation matrix. In my opinion, the correlation is not really useful unless users wish to investigate for multicollinearity problem (which I think this is not an optimal option).

To find the *p*-values of the fixed effect, the similar codes from the null model can be applied:[2]

```
out1 <- summary(m1)

coef1 <- coef(out1)

tvalue1 <- coef1[,"t value"]

pnorm(abs(tvalue1), lower.tail=FALSE) * 2
```

```
(Intercept)     IQ_verb
          0           0
```

[2] Because of multiple *p*-values, researchers may have a problem of inflated familywise error rate. Users may use the `p.adjust` function to correct the *p*-values. For example, the Holm's method is used by

```
p.adjust(pnorm(abs(tvalue1), lower.tail=FALSE) * 2, method = "holm")
```

To find the residual intraclass correlation, the codes used to find intraclass correlation in the null model can be applied:

```
ranef1 <- out1@REmat

tau00_1 <- as.numeric(ranef1[1, 3])

sigma2_1 <- as.numeric(ranef1[2, 3])

icc_1 <- tau00_1 / (tau00_1 + sigma2_1)

icc_1
```

```
[1] 0.1956732
```

Model 2: Means-as-Outcomes Model

In this model, L1 predictor is not included and L2 predictor is included in the model. For this example, the language scores (`langPOST`) is used as a dependent variable and predicted by the five types of schools (`denomina`). Because the type of school is a categorical variable, the variable must be transformed into the factor format:

```
dat$denomina <- factor(dat$denomina)
```

The frequency of each group can be examined by the `table` function:

```
table(dat$denomina)
```

```
   1    2    3    4    5 
1047 1369  922  180  240 
```

The Means-as-Outcomes model would be

L1 $$Y_{ij} = \beta_{0j} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} + u_{0j} \qquad u_{0j} \sim N(0, \tau_{00})$$

These notations should represent (the blue lines indicate that the meanings changed from Model 0)

- Y_{ij} = The language score of Student *i* in School *j*
- W_{1j} = A dummy variable whether School *j* is classified as Type 2
- W_{2j} = A dummy variable whether School *j* is classified as Type 3
- W_{3j} = A dummy variable whether School *j* is classified as Type 4
- W_{4j} = A dummy variable whether School *j* is classified as Type 5
- β_{0j} = The average language score within School *j*
- γ_{00} = The average language score across all schools in Type 1
- γ_{10} = The difference in language score between all schools in Type 2 and all schools in Type 1
- γ_{20} = The difference in language score between all schools in Type 3 and all schools in Type 1
- γ_{30} = The difference in language score between all schools in Type 4 and all schools in Type 1
- γ_{40} = The difference in language score between all schools in Type 5 and all schools in Type 1
- e_{ij} = The deviation of the language score of Student *i* from the School *j* mean
- u_{0j} = The deviation of the language score of School *j* mean from the mean across schools in the same Type that School *j* is in

- σ^2 = The language score variance within schools (L1 variance)
- τ_{00} = The language score residual variance across schools (L2 variance) controlling for the type of schools

The Means-as-Outcomes model can be run by the `lmer` function:

```
m2 <- lmer(langPOST ~ 1 + denomina + (1|schoolnr), data = dat, REML=FALSE)

summary(m2)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + denomina + (1 | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 26588 26632 -13287    26574   26567
Random effects:
 Groups   Name        Variance Std.Dev.
 schoolnr (Intercept) 15.833   3.9790
 Residual             62.880   7.9297
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
            Estimate Std. Error t value
(Intercept)  39.2940     0.5747   68.38
denomina2     3.1839     0.7766    4.10
denomina3     0.9820     0.8350    1.18
denomina4     4.4124     1.5269    2.89
denomina5     2.6042     1.3629    1.91

Correlation of Fixed Effects:
          (Intr) denmn2 denmn3 denmn4
denomina2 -0.740
denomina3 -0.688  0.509
denomina4 -0.376  0.279  0.259
denomina5 -0.422  0.312  0.290  0.159
```

The additional notation in this formula is the fixed effect of the types of schools. The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + denomina + (1|schoolnr)
langPOST ~ 1 + d2 + d3 + d4 + d5 + (1|schoolnr)
```

$$\underbrace{Y_{ij} = \gamma_{00}(1) + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j}}_{\text{Fixed Effect}} + \underbrace{u_{0j}(1) + e_{ij}}_{\text{Random Effect}}$$

Fixed Effect + Random Effect

The output is similar to Model 1. In the random-effect section, τ_{00} should be different from the null model because of different meanings. σ^2, however, should have a similar value to Model 0. The output of the fixed effect has five rows, which represent intercept (γ_{00}) and the differences of a specific type of schools from the reference type of school (γ_{01}, γ_{02}, γ_{03}, and γ_{04}).

To find *p*-values of the fixed effect, the similar codes from the null model can be applied:

```
out2 <- summary(m2)

coef2 <- coef(out2)

tvalue2 <- coef2[,"t value"]

pnorm(abs(tvalue2), lower.tail=FALSE) * 2
```

```
 (Intercept)    denomina2    denomina3    denomina4    denomina5
0.0000000000 0.0000413438 0.2396008125 0.0038539161 0.0560301936
```

To find residual intraclass correlation, the codes used to find intraclass correlation in the null model can be applied:

```
ranef2 <- out2@REmat

tau00_2 <- as.numeric(ranef2[1, 3])

sigma2_2 <- as.numeric(ranef2[2, 3])

icc_2 <- tau00_2 / (tau00_2 + sigma2_2)

icc_2
```

```
[1] 0.2011485
```

Model 3: Adjusted-Means-as-Outcomes Model

In this model, Both L1 and L2 predictors are included in the model. The regression coefficient of the L1 predictor, however, is not random across schools. For this example, the language scores (`langPOST`) is predicted by the verbal IQ scores (`IQ_verb`) and the five types of schools (`denomina`). The Adjusted-Means-as-Outcomes model would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} + u_{0j} \qquad u_{0j} \sim N(0, \tau_{00})$$

$$\beta_{1j} = \gamma_{10}$$

These notations should represent (the blue lines indicate that the meanings changed from Model 1)

- Y_{ij} = The language score of Student *i* in School *j*
- X_{ij} = The verbal IQ score of Student *i* in School *j*
- W_{1j} = A dummy variable whether School *j* is classified as Type 2
- W_{2j} = A dummy variable whether School *j* is classified as Type 3
- W_{3j} = A dummy variable whether School *j* is classified as Type 4
- W_{4j} = A dummy variable whether School *j* is classified as Type 5
- β_{0j} = The expected average of language score within School *j* when the verbal IQ score is 0, which is also referred to as *adjusted mean*
- β_{1j} = The expected change in language score when the verbal IQ score of students in School *j* increases by 1. In this case, the expected changes across schools are the same.
- γ_{00} = The expected average language score across all schools when the verbal IQ score is 0 and the type of school is 1.
- γ_{01} = The difference in adjusted language score (when the verbal IQ score is 0) between all schools in Type 2 and all schools in Type 1
- γ_{02} = The difference in adjusted language score between all schools in Type 3 and all schools in Type 1
- γ_{03} = The difference in adjusted language score between all schools in Type 4 and all schools in Type 1
- γ_{04} = The difference in adjusted language score between all schools in Type 5 and all schools in Type 1

- γ_{10} = The schools' average expected change in language score when the verbal IQ score increase by 1.
- e_{ij} = The difference between the actual language score and the predicted language score of Student *i* in School *j*
- u_{0j} = The deviation of the adjusted language score average of School *j* (when the verbal IQ score is 0) from the mean across schools in the same Type that School *j* is in
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the verbal IQ score
- τ_{00} = The language score residual variance across schools (L2 residual variance) controlling for the verbal IQ score and type of schools

The Adjusted-Means-as-Outcomes model can be run by the `lmer` function:

```
m3 <- lmer(langPOST ~ 1 + IQ_verb + denomina + (1|schoolnr), data = dat, REML=FALSE)

summary(m3)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + denomina + (1 | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24910 24960 -12447    24894   24894
Random effects:
 Groups   Name        Variance Std.Dev.
 schoolnr (Intercept)  8.7919  2.9651
 Residual             40.4741  6.3619
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
            Estimate Std. Error t value
(Intercept) 40.12190    0.43551   92.13
IQ_verb      2.50392    0.05438   46.05
denomina2    2.19581    0.58807    3.73
denomina3    0.12757    0.63261    0.20
denomina4    2.02632    1.15697    1.75
denomina5    0.81287    1.03187    0.79

Correlation of Fixed Effects:
          (Intr) IQ_vrb denmn2 denmn3 denmn4
IQ_verb    0.041
denomina2 -0.741 -0.037
denomina3 -0.689 -0.030  0.510
denomina4 -0.378 -0.045  0.280  0.260
denomina5 -0.423 -0.038  0.313  0.291  0.160
```

The formula includes the fixed effects of both verbal IQ and the types of schools. The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + IQ_verb + denomina + (1|schoolnr)
langPOST ~ 1 + IQ_verb + d2 + d3 + d4 + d5 + (1|schoolnr)
```

$$Y_{ij} = \underbrace{\gamma_{00}(1) + \gamma_{10}X_{ij} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j}}_{\text{Fixed Effect}} + \underbrace{u_{0j}(1)}_{\text{Random Effect}} + e_{ij}$$

Fixed Effect + Random Effect

To find *p*-values of the fixed effect, the similar codes from the null model can be applied:

```
out3 <- summary(m3)

coef3 <- coef(out3)

tvalue3 <- coef3[,"t value"]
```

```
pnorm(abs(tvalue3), lower.tail=FALSE) * 2
```

```
(Intercept)       IQ_verb     denomina2     denomina3     denomina4     denomina5
0.0000000000 0.0000000000 0.0001885398 0.8401800753 0.0798770541 0.4308395114
```

To find residual intraclass correlation, the codes used to find intraclass correlation in the null model can be applied:

```
ranef3 <- out3@REmat

tau00_3 <- as.numeric(ranef3[1, 3])

sigma2_3 <- as.numeric(ranef3[2, 3])

icc_3 <- tau00_3 / (tau00_3 + sigma2_3)

icc_3
```

```
[1] 0.1784578
```

Model 4: Random Coefficients Regression

Similar to Model 1, a L1 predictor is put in the model but the slope is random across schools. For example, the language scores (`langPOST`) is predicted by verbal IQ score (`IQ_verb`) but the effect of verbal IQ allows to be varied across schools. The random-coefficients regression model would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + u_{1j} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ \tau_{10} & \tau_{11} \end{bmatrix}\right)$$

These notations should represent (the blue lines indicate that the meanings changed from Model 1)

- Y_{ij} = The language score of Student *i* in School *j*
- X_{ij} = The verbal IQ score of Student *i* in School *j*
- β_{0j} = The expected average of language score within School *j* when the verbal IQ score is 0, which is also referred to as *adjusted mean*
- β_{1j} = The expected change in language score when the verbal IQ score of students in School *j* increases by 1.
- γ_{00} = The expected average language score across all schools when the verbal IQ score is 0.
- γ_{10} = The schools' average expected change in language score when the verbal IQ score increase by 1.
- e_{ij} = The difference between the actual language score and the predicted language score of Student *i* in School *j*
- u_{0j} = The deviation of the adjusted language score average of School *j* (when the verbal IQ score is 0) from the grand mean of adjusted average across schools
- u_{1j} = The deviation of the slope of verbal IQ score of School *j* from the average across schools
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the verbal IQ score
- τ_{00} = The language score residual variance across schools (L2 residual variance) controlling for the verbal IQ score
- τ_{11} = The variance of the slope of verbal IQ score across schools

- τ_{10} = The covariance between the expected value of language score when the verbal IQ score equals 0 and the slope of verbal IQ scores
- $\rho_{10} = \tau_{10}/\sqrt{\tau_{00}\tau_{11}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)

The random-coefficients regression model can be run by the `lmer` function:

```
m4 <- lmer(langPOST ~ 1 + IQ_verb + (1 + IQ_verb|schoolnr), data = dat, REML=FALSE)

summary(m4)
```

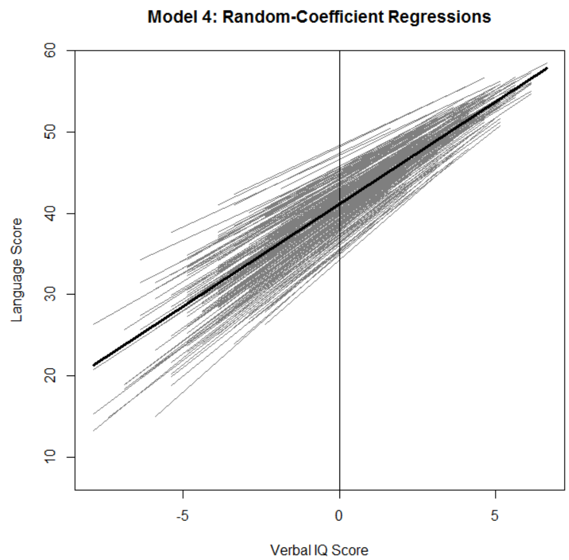
```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + (1 + IQ_verb | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24891 24928 -12439    24879   24884
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 schoolnr (Intercept)  9.77498 3.12650
          IQ_verb      0.20244 0.44994  -0.768
 Residual             39.75002 6.30476
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
            Estimate Std. Error t value
(Intercept)  41.1281     0.2425   169.6
IQ_verb       2.5194     0.0633    39.8

Correlation of Fixed Effects:
        (Intr)
IQ_verb -0.353
```

The additional notation in this formula is the random effect of verbal IQ. The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + IQ_verb + (1 + IQ_verb|schoolnr)
```

$$Y_{ij} = \underbrace{\gamma_{00}(1) + \gamma_{10}X_{ij}}_{\text{Fixed Effect}} + \underbrace{u_{0j}(1) + u_{1j}X_{ij}}_{\text{Random Effect}} + e_{ij}$$

The output is similar to the previous models. The random effects has an additional line of random slope of the effect of verbal IQ, which is τ_{11}. The covariance, τ_{10}, between random intercept and random slope is not listed here. Rather, the correlation, ρ_{10}, between random intercept and random slope is listed. Before interpreting the correlation, the meaning of verbal IQ equal 0 must be investigated. The average of the verbal IQ across all students is close to 0; therefore, β_{0j} (or u_{0j}) can be viewed as the expected average of school language score when verbal IQ equals to its mean. Because the correlation is strongly negative, the school that has a low expected value of language score (when verbal IQ equals 0) will be more likely to have a stronger positive slope than the school with a high expected value. See the figure:



The grey lines are the regression lines of each school. The black line is the regression line from the average intercept and average slope across schools. See the vertical line representing the verbal IQ of 0. When the language score is low, the slope is steeper. As another insight from this figure, when the verbal IQ score is low (the left part of the *X* axis), the language score is lower on average and has more variance. When the verbal IQ score is high, the language score is higher on average and has less variance.

To find *p*-values of the fixed effect, the similar codes from the null model can be applied:

```
out4 <- summary(m4)

coef4 <- coef(out4)

tvalue4 <- coef4[,"t value"]

pnorm(abs(tvalue4), lower.tail=FALSE) * 2
```

```
(Intercept)      IQ_verb
          0            0
```

To find residual intraclass correlation, the codes used to find intraclass correlation in the null model can be applied (be careful on the position of τ_{00} and σ^2):

```
ranef4 <- out4@REmat

tau00_4 <- as.numeric(ranef4[1, 3])

sigma2_4 <- as.numeric(ranef4[3, 3])

icc_4 <- tau00_4 / (tau00_4 + sigma2_4)

icc_4
```

```
[1] 0.1973747
```

Model 5: Random Coefficients with Fixed Intercept Regression

Similar to Model 4, a L1 predictor is put in the model but the slope is random across schools. However, the intercept is fixed across groups. For example, the language scores (`langPOST`) is predicted verbal IQ score (`IQ_verb`). The effect of verbal IQ varies across schools. The expected language score when the verbal IQ is 0 is the same across schools. This model is rarely used in practice except some longitudinal data or repeated-measures design. The random-coefficients with fixed intercept regression model would be

L1	$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij}$	$e_{ij} \sim N(0, \sigma^2)$
L2	$\beta_{0j} = \gamma_{00}$	$u_{1j} \sim N(0, \tau_{11})$
	$\beta_{1j} = \gamma_{10} + u_{1j}$	

These notations should represent (the blue lines indicate that the meanings changed from Model 4)

- Y_{ij} = The language score of Student *i* in School *j*
- X_{ij} = The verbal IQ score of Student *i* in School *j*
- β_{0j} = The expected average of language score within School *j* when the verbal IQ score is 0, which is also referred to as *adjusted mean*. In this case, the adjusted mean is constant across schools.
- β_{1j} = The expected change in language score when the verbal IQ score of students in School *j* increases by 1.
- γ_{00} = The expected average language score across all schools when the verbal IQ score is 0.
- γ_{10} = The schools' average expected change in language score when the verbal IQ score increase by 1.
- e_{ij} = The difference between the actual language score and the predicted language score of Student *i* in School *j*
- u_{1j} = The deviation of the slope of verbal IQ score of School *j* from the average across schools
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the verbal IQ score
- τ_{11} = The variance of the slope of verbal IQ score across schools

The random-coefficients with fixed intercept regression model can be run by the `lmer` function:

```
m5 <- lmer(langPOST ~ 1 + IQ_verb + (0 + IQ_verb|schoolnr), data = dat, REML=FALSE)

summary(m5)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + (0 + IQ_verb | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 25342 25367 -12667    25334   25340
Random effects:
 Groups   Name    Variance Std.Dev.
 schoolnr IQ_verb  0.28423 0.53313
 Residual         48.61650 6.97255
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
            Estimate Std. Error t value
(Intercept)  41.3671     0.1162   356.0
IQ_verb       2.6806     0.0691    38.8

Correlation of Fixed Effects:
        (Intr)
IQ_verb -0.023
```

Note that 0 is used instead of 1 in the parenthesis, which means that the random intercept is not estimated.[3] If `(IQ_verb|schoolnr)` is specified, the function will still estimate the random intercept as a default. The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + IQ_verb + (0 + IQ_verb|schoolnr)
```

$$Y_{ij} = \underbrace{\gamma_{00}(1) + \gamma_{10}X_{ij}}_{\text{Fixed Effect}} + \underbrace{u_{1j}X_{ij}}_{\text{Random Effect}} + e_{ij}$$

In the output, the `Random effects` section has the variance of the random slope, which is τ_{11}, but not the random intercept variance. Because there is only one random effect in L2, the correlation between random effects does not exist as well.

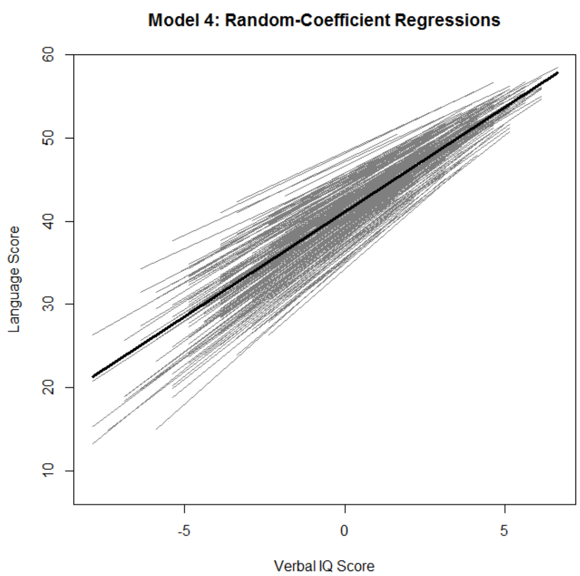
To find *p*-values of the fixed effect, the similar codes from the null model can be applied:

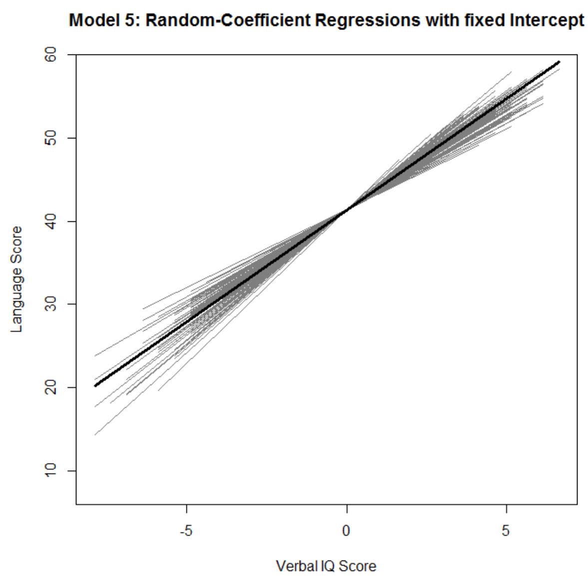
```
out5 <- summary(m5)

coef5 <- coef(out5)

tvalue5 <- coef5[,"t value"]

pnorm(abs(tvalue5), lower.tail=FALSE) * 2
```

```
(Intercept)     IQ_verb
          0           0
```

Because the random intercept does not exist, the intraclass correlation is not defined.

Users may be wondering about the difference between Model 4 and Model 5. Let's draw the regression lines of each group to see the difference:





[3] In `lme4` version 0.999999-0, the code with fixed intercept and random slope has a problem when L1 predictor is in a factor format (e.g., sex is transformed into a factor format). Users should transform any L1 categorical variables into dummy variables by hand for a model with fixed intercept and random slope (similar to Model 5)

The grey lines are the regression lines of each school. The black line is the regression line from the average intercept and average slope across schools. Notice that in Model 5, the expected language score at verbal IQ of 0 is not varied (a point).

Model 6: Intercepts- and Slopes-as-Outcomes Model

From Model 4, a L2 predictor is used to predict both random intercepts and random slopes. Looking in a different view, from Model 3, a random slope is specified. For example, the language scores (`langPOST`) is predicted verbal IQ score (`IQ_verb`) and the type of schools (`denomina`). The slope of verbal IQ score on the language score is varying across schools. The random slope is also predicted by the type of schools. The intercepts- and slopes-as-outcomes model would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + \gamma_{11}W_{1j} + \gamma_{12}W_{2j} + \gamma_{13}W_{3j} + \gamma_{14}W_{4j} + u_{1j} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ \tau_{10} & \tau_{11} \end{bmatrix}\right)$$

These notations should represent (the blue lines indicate that the meanings changed from Model 4)

- Y_{ij} = The language score of Student *i* in School *j*
- X_{ij} = The verbal IQ score of Student *i* in School *j*
- W_{1j} = A dummy variable whether School *j* is classified as Type 2
- W_{2j} = A dummy variable whether School *j* is classified as Type 3
- W_{3j} = A dummy variable whether School *j* is classified as Type 4
- W_{4j} = A dummy variable whether School *j* is classified as Type 5
- β_{0j} = The expected average of language score within School *j* when the verbal IQ score is 0, which is also referred to as *adjusted mean*.
- β_{1j} = The expected change in language score when the verbal IQ score of students in School *j* increases by 1.
- γ_{00} = The expected average language score across all schools when the verbal IQ score is 0 and the type of school is 1.
- γ_{01} = The difference in adjusted language score (when the verbal IQ score is 0) between all schools in Type 2 and all schools in Type 1
- γ_{02} = The difference in adjusted language score between all schools in Type 3 and all schools in Type 1
- γ_{03} = The difference in adjusted language score between all schools in Type 4 and all schools in Type 1
- γ_{04} = The difference in adjusted language score between all schools in Type 5 and all schools in Type 1
- γ_{10} = The schools' average expected change in language score when the verbal IQ score increase by 1 given that the type of school is 1. That is the average slope of the verbal IQ among school with the type of 1.
- γ_{11} = The difference in the slope of verbal IQ between all schools in Type 2 and all schools in Type 1
- γ_{12} = The difference in the slope of verbal IQ between all schools in Type 3 and all schools in Type 1

- γ_{13} = The difference in the slope of verbal IQ between all schools in Type 4 and all schools in Type 1
- γ_{14} = The difference in the slope of verbal IQ between all schools in Type 5 and all schools in Type 1
- e_{ij} = The difference between the actual language score and the predicted language score of Student *i* in School *j*
- u_{0j} = The deviation of the adjusted language score average of School *j* (when the verbal IQ score is 0) from the mean across schools in the same Type that School *j* is in
- u_{1j} = The deviation of the slope of verbal IQ score of School *j* from the expected slope across schools in the same type that School *j* is in
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the verbal IQ score
- τ_{00} = The language score residual variance across schools (L2 residual variance) controlling for the verbal IQ score and the type of school
- τ_{11} = The residual variance of the slope of verbal IQ score across schools controlling for the type of school
- τ_{10} = The covariance between the residual of the random intercept and the residual of the random slope
- $\rho_{10} = \tau_{10}/\sqrt{\tau_{00}\tau_{11}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)

The intercepts- and slopes-as-outcomes model can be run by the `lmer` function:

```
m6 <- lmer(langPOST ~ 1 + IQ_verb + denomina + IQ_verb*denomina + (1 + IQ_verb|schoolnr), data =
dat, REML=FALSE)

summary(m6)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + denomina + IQ_verb * denomina + (1 +      IQ_verb | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24881 24968 -12426    24853   24859
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 schoolnr (Intercept)  8.69164 2.94816
          IQ_verb      0.16195 0.40243  -0.826
 Residual             39.77899 6.30706
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
                  Estimate Std. Error t value
(Intercept)       40.22489    0.43295   92.91
IQ_verb            2.68758    0.11224   23.95
denomina2          2.15422    0.58451    3.69
denomina3          0.09217    0.62872    0.15
denomina4          2.21263    1.16046    1.91
denomina5          0.65672    1.03226    0.64
IQ_verb:denomina2 -0.19222    0.15510   -1.24
IQ_verb:denomina3 -0.31980    0.16486   -1.94
IQ_verb:denomina4 -0.62469    0.30638   -2.04
IQ_verb:denomina5  0.01977    0.26655    0.07

Correlation of Fixed Effects:
            (Intr) IQ_vrb denmn2 denmn3 denmn4 denmn5 IQ_v:2 IQ_v:3 IQ_v:4
IQ_verb     -0.290
denomina2   -0.741  0.215
denomina3   -0.689  0.200  0.510
denomina4   -0.373  0.108  0.276  0.257
denomina5   -0.419  0.122  0.311  0.289  0.156
IQ_vrb:dnm2  0.210 -0.724 -0.323 -0.145 -0.078 -0.088
```

```
IQ_vrb:dnm3  0.198 -0.681 -0.146 -0.324 -0.074 -0.083  0.493
IQ_vrb:dnm4  0.106 -0.366 -0.079 -0.073 -0.437 -0.045  0.265  0.249
IQ_vrb:dnm5  0.122 -0.421 -0.091 -0.084 -0.046 -0.407  0.305  0.287  0.154
```

In the formula, the asterisk is used to specify any types of interaction. In this case, the interaction of verbal IQ score and type of schools is specified by `IQ_verb*denomina`.[4] The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + IQ_verb + denomina
           + IQ_verb*denomina
           + (1 + IQ_verb|schoolnr)
langPOST ~ 1 + IQ_verb + d2 + d3 + d4 + d5
           + IQ_verb*d2 + IQ_verb*d3 + IQ_verb*d4 + IQ_verb*d5
           + (1 + IQ_verb|schoolnr)
```

$$\begin{aligned}Y_{ij} = \gamma_{00}(1) + \gamma_{10}X_{ij} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} \\ +\gamma_{11}X_{ij}W_{1j} + \gamma_{12}X_{ij}W_{2j} + \gamma_{13}X_{ij}W_{3j} + \gamma_{14}X_{ij}W_{4j}\end{aligned}$$ Fixed Effect

$$+u_{0j}(1) + u_{1j}X_{ij} + e_{ij}$$ Random Effect

The output is similar to the previous models. In the fixed effects, the colon, `:`, means the interaction effect. For example, `IQ_verb:denomina2` is the interaction effect between verbal IQ score, X_{ij}, and the dummy variable representing school type 2, W_{1j}, which represents γ_{11}.

To find *p*-values of the fixed effect, the similar codes from the null model can be applied:

```
out6 <- summary(m6)

coef6 <- coef(out6)

tvalue6 <- coef6[,"t value"]

pnorm(abs(tvalue6), lower.tail=FALSE) * 2
```

```
      (Intercept)           IQ_verb         denomina2         denomina3         denomina4
     0.000000e+00      1.022534e-126      2.282213e-04      8.834484e-01      5.656111e-02
        denomina5 IQ_verb:denomina2 IQ_verb:denomina3 IQ_verb:denomina4 IQ_verb:denomina5
     5.246490e-01      2.152141e-01      5.240791e-02      4.145315e-02      9.408734e-01
```

If the resulting *p* values are not easy to see which effects are significant, a little trick can be used by comparing the resulting *p* values with a priori alpha level (e.g., .05):

```
(pnorm(abs(tvalue6), lower.tail=FALSE) * 2) < .05
```

```
      (Intercept)           IQ_verb         denomina2         denomina3         denomina4
             TRUE              TRUE              TRUE             FALSE             FALSE
        denomina5 IQ_verb:denomina2 IQ_verb:denomina3 IQ_verb:denomina4 IQ_verb:denomina5
            FALSE             FALSE             FALSE              TRUE             FALSE
```

Note that these *p*-values are not adjusted for familywise error rate.

To find the residual intraclass correlation, the codes used to find intraclass correlation in the null model can be applied (be careful on the position of τ_{00} and σ^2):

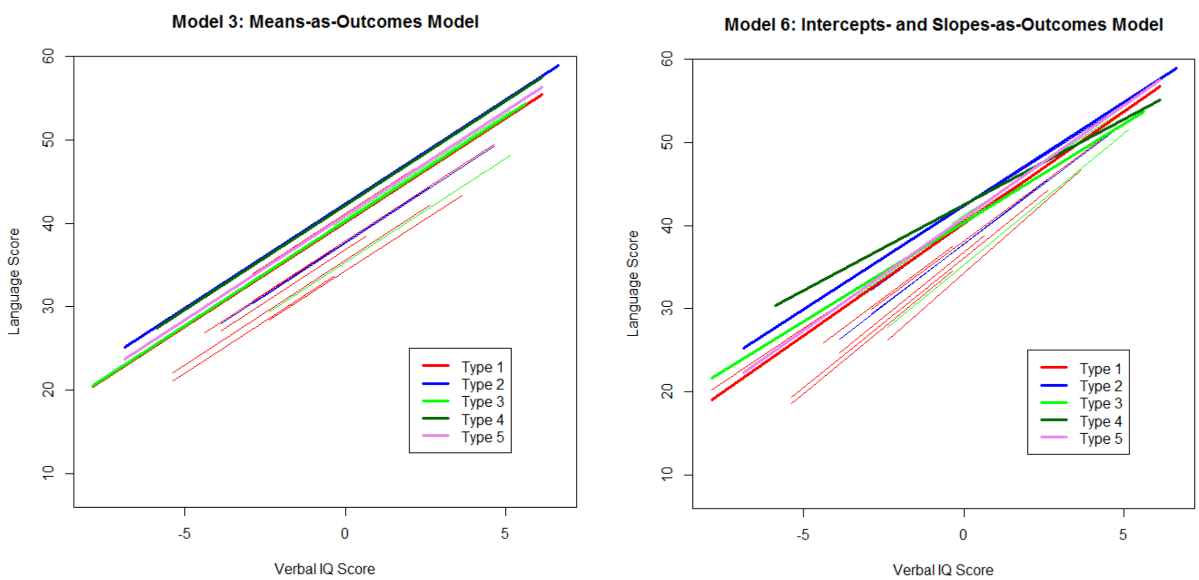
```
ranef6 <- out6@REmat

tau00_6 <- as.numeric(ranef6[1, 3])

sigma2_6 <- as.numeric(ranef6[3, 3])
```

[4] Users can specify interactions by colon, `:` , or asterisk, `*`. Check the difference between two methods by going to the help page of the formula function by typing `?formula`

```
icc_6 <- tau00_6 / (tau00_6 + sigma2_6)

icc_6
```

```
[1] 0.1793177
```

Users may be wondering about the difference between Model 3 and Model 6. Model 3 does not have the random slope whereas Model 6 has random slope. Let's draw the regression lines of each group to see the difference:



In the figures, different colors represent different school types. The solid line is the regression line based on the average intercept and average slope across schools with the same type. The thin line is the regression line of each school. There are many more thin lines in the graphs but most of them are overlapped with the solid line. Notice that Model 3 has a constant slope across schools and the slopes of each color are the same. On the other hand, Model 6 has different slopes across schools and the slopes of each color are different.

I will show you four more models. These models will be used as reference models only in order to make a deviance test, which is shown later. The parameter estimates in this model could not be trusted because the constrained parameters in these models are very important. The constraint can make a serious model misspecification that makes biased estimates of the parameters in the model.

Model 7: Random Coefficients Regression without Covariances between Random Effects (Reference Model)

This model is similar to Model 4 but the covariance between random intercepts are random slopes is fixed to 0. The random-coefficients regression model without covariances between random effects would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + u_{0j} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ 0 & \tau_{11} \end{bmatrix}\right)$$

$$\beta_{1j} = \gamma_{10} + u_{1j}$$

The interpretations of those notations are similar with Model 4 except τ_{10} and ρ_{10} are all 0.

The random-coefficients regression model without covariances between random effects can be run by the `lmer` function:

```
m7 <- lmer(langPOST ~ 1 + IQ_verb + (1|schoolnr) + (0 + IQ_verb|schoolnr), data = dat,
REML=FALSE)

summary(m7)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + (1 | schoolnr) + (0 + IQ_verb | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24913 24945 -12452    24903   24908
Random effects:
 Groups   Name        Variance Std.Dev.
 schoolnr (Intercept)  9.81365 3.13267
 schoolnr IQ_verb      0.17618 0.41974
 Residual             39.80108 6.30881
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
            Estimate Std. Error t value
(Intercept) 41.09121    0.24370  168.61
IQ_verb      2.53511    0.06294   40.28

Correlation of Fixed Effects:
        (Intr)
IQ_verb 0.001
```

The formula in this model is interesting. The random intercept is listed in the first parenthesis to be random across schools. The random slope is listed in the second parenthesis with 0 to tell the program that the covariance between random intercept and random slope is not estimated. The mapping from the formula and reduced-form equation would be

`langPOST ~ 1 + IQ_verb + (1|schoolnr) + (0 + IQ_verb|schoolnr)`
$Y_{ij} = \underbrace{\gamma_{00}(1) + \gamma_{10}X_{ij}}_{\text{Fixed Effect}} + \underbrace{u_{0j}(1) + u_{1j}X_{ij}}_{\text{Random Effect}} + e_{ij}$

The output is similar to Model 4 but the correlation between the random intercepts and random slopes in the `Random effects` section was not shown. That is, the covariance/correlation was fixed to 0. Note that this model should not be used as a final model. Even though the covariance/correlation is small, the covariance/correlation should be estimated. Otherwise, the fixed correlation can lead to biases in the parameters in the model. Therefore Model 4 is preferred to Model 7. The reason why this model is listed here because this model can be used to compare with Model 4 by deviance test listed below.

Model 8: Intercepts- and Slopes-as-Outcomes Model without Cross-level Interaction (Reference Model)

This model is similar to Model 6 but the cross-level interactions are not specified. That is, γ_{11}, γ_{12}, γ_{13}, and γ_{14} are constrained to 0. The intercepts- and slopes-as-outcomes model without cross-level interaction would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + u_{1j} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ \tau_{10} & \tau_{11} \end{bmatrix}\right)$$

The interpretations of these notations are similar to Model 6 except the followings:

- γ_{10} = The schools' average expected change in language score when the verbal IQ score increase by 1.
- u_{1j} = The deviation of the slope of verbal IQ score of School *j* from the average slope across schools (γ_{10})
- τ_{11} = The variance of the slope of verbal IQ score across schools
- τ_{10} = The covariance between the residual of the random intercept and the random slope
- $\rho_{10} = \tau_{10}/\sqrt{\tau_{00}\tau_{11}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)

The intercepts- and slopes-as-outcomes model without cross-level interaction can be run by the `lmer` function:

```
m8 <- lmer(langPOST ~ 1 + IQ_verb + denomina + (1 + IQ_verb|schoolnr), data = dat, REML=FALSE)

summary(m8)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + denomina + (1 + IQ_verb | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24880 24942 -12430    24860   24860
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 schoolnr (Intercept)  8.78392 2.96377
          IQ_verb      0.19834 0.44536  -0.792
 Residual             39.75412 6.30509
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
            Estimate Std. Error t value
(Intercept) 40.42641    0.42065   96.10
IQ_verb      2.51732    0.06308   39.91
denomina2    1.93266    0.55312    3.49
denomina3   -0.29465    0.59491   -0.50
denomina4    1.23024    1.04196    1.18
denomina5    0.77965    0.94095    0.83

Correlation of Fixed Effects:
          (Intr) IQ_vrb denmn2 denmn3 denmn4
IQ_verb   -0.171
denomina2 -0.734 -0.022
denomina3 -0.683 -0.021  0.522
denomina4 -0.386 -0.036  0.299  0.278
denomina5 -0.428 -0.036  0.331  0.308  0.176
```

Note that, in the formula, the product term is not listed. The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + IQ_verb + denomina
           + (1 + IQ_verb|schoolnr)
langPOST ~ 1 + IQ_verb + d2 + d3 + d4 + d5
           + (1 + IQ_verb|schoolnr)
```

$Y_{ij} = \gamma_{00}(1) + \gamma_{10}X_{ij} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j}$	Fixed Effect
$+u_{0j}(1) + u_{1j}X_{ij} + e_{ij}$	Random Effect

The parameter estimates from this model could be wrong because the cross-level interactions were not listed. The effect of not listing the cross-level interaction could be a serious misspecification and the parameter estimates in the current model is biased (Raudenbush & Byrk, 2002). Therefore, Model 6 is preferred to Model 8. This model, however, is useful in comparing with Model 6 to check the significance of the cross-level interactions using deviance test.

Model 9: Intercepts- and Slopes-as-Outcomes Model without Random Slopes (Reference Model)

This model is similar to Model 6. However, the random slopes and the covariance between random intercepts and random slopes are dropped. That is, τ_{11} and τ_{10} are fixed to 0. The random slope is also predicted by the type of schools. The intercepts- and slopes-as-outcomes model without random slopes would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} + u_{0j} \qquad u_{0j} \sim N(0, \tau_{00})$$

$$\beta_{1j} = \gamma_{10} + \gamma_{11}W_{1j} + \gamma_{12}W_{2j} + \gamma_{13}W_{3j} + \gamma_{14}W_{4j}$$

The interpretations of these notations are similar to Model 6.

The intercepts- and slopes-as-outcomes model without random slopes can be run by the `lmer` function:

```
m9 <- lmer(langPOST ~ 1 + IQ_verb + denomina + IQ_verb*denomina + (1|schoolnr), data = dat,
REML=FALSE)

summary(m9)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + denomina + IQ_verb * denomina + (1 |      schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24906 24981 -12441    24882   24889
Random effects:
 Groups   Name        Variance Std.Dev.
 schoolnr (Intercept)  8.7363  2.9557
 Residual             40.3543  6.3525
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
                  Estimate Std. Error t value
(Intercept)       40.18310    0.43512   92.35
IQ_verb            2.69086    0.09812   27.42
denomina2          2.13444    0.58695    3.64
denomina3          0.07139    0.63139    0.11
denomina4          2.24679    1.16412    1.93
denomina5          0.66261    1.03209    0.64
IQ_verb:denomina2 -0.18142    0.13644   -1.33
IQ_verb:denomina3 -0.39670    0.14539   -2.73
IQ_verb:denomina4 -0.63913    0.27524   -2.32
IQ_verb:denomina5  0.04242    0.23424    0.18

Correlation of Fixed Effects:
            (Intr) IQ_vrb denmn2 denmn3 denmn4 denmn5 IQ_v:2 IQ_v:3 IQ_v:4
IQ_verb      0.074
denomina2   -0.741 -0.055
denomina3   -0.689 -0.051  0.511
denomina4   -0.374 -0.028  0.277  0.258
denomina5   -0.422 -0.031  0.313  0.291  0.158
IQ_vrb:dnm2 -0.053 -0.719  0.031  0.037  0.020  0.022
IQ_vrb:dnm3 -0.050 -0.675  0.037  0.032  0.019  0.021  0.485
IQ_vrb:dnm4 -0.026 -0.356  0.019  0.018 -0.119  0.011  0.256  0.241
IQ_vrb:dnm5 -0.031 -0.419  0.023  0.021  0.012 -0.060  0.301  0.283  0.149
```

The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + IQ_verb + denomina
             + IQ_verb*denomina
             + (1|schoolnr)
langPOST ~ 1 + IQ_verb + d2 + d3 + d4 + d5
             + IQ_verb*d2 + IQ_verb*d3 + IQ_verb*d4 + IQ_verb*d5
             + (1|schoolnr)
```

$$Y_{ij} = \gamma_{00}(1) + \gamma_{10}X_{ij} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j}$$

Fixed Effect

Random Effect

$$+\gamma_{11}X_{ij}W_{1j} + \gamma_{12}X_{ij}W_{2j} + \gamma_{13}X_{ij}W_{3j} + \gamma_{14}X_{ij}W_{4j}$$
$$+u_{0j}(1) + e_{ij}$$

The output is similar to Model 6 but the variance of random slope and the correlation between the random intercepts and random slopes in the `Random effects` section were not listed. If this model was true, the type of schools would be the only reason why the effects of verbal IQ were different across schools. This situation is rarely true, however. This model is usually compared with Model 6 to check whether the predictors on the slope equation are the only explanation why the effects of L1 predictors are varied across L2 units.

Model 10: Intercepts- and Slopes-as-Outcomes Model without Residual Covariances between Random Effects (Reference Model)

This model is similar to Model 6. However, the covariance between random intercepts and random slopes are dropped. That is, τ_{10} and ρ_{10} are fixed to 0. The intercepts- and slopes-as-outcomes model without random slopes would be

$$\text{L1} \qquad Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$
$$\text{L2} \qquad \begin{aligned} \beta_{0j} &= \gamma_{00} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} + u_{0j} \\ \beta_{1j} &= \gamma_{10} + \gamma_{11}W_{1j} + \gamma_{12}W_{2j} + \gamma_{13}W_{3j} + \gamma_{14}W_{4j} + u_{1j} \end{aligned} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ 0 & \tau_{11} \end{bmatrix}\right)$$

The interpretations of these notations are similar to Model 6.

The intercepts- and slopes-as-outcomes model without residual covariances between random effects can be run by the `lmer` function:

```
m10 <- lmer(langPOST ~ 1 + IQ_verb + denomina + IQ_verb*denomina + (1|schoolnr) + (0 +
IQ_verb|schoolnr), data = dat, REML=FALSE)

summary(m10)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + denomina + IQ_verb * denomina + (1 |      schoolnr) + (0 + IQ_verb |
schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24903 24984 -12439    24877   24882
Random effects:
 Groups   Name        Variance Std.Dev.
 schoolnr (Intercept)  8.72342 2.9535
 schoolnr IQ_verb      0.13734 0.3706
 Residual             39.83406 6.3114
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
                  Estimate Std. Error t value
(Intercept)       40.20594    0.43579   92.26
IQ_verb            2.71494    0.11150   24.35
denomina2          2.14390    0.58776    3.65
denomina3          0.06481    0.63266    0.10
denomina4          2.22625    1.16414    1.91
denomina5          0.67794    1.03387    0.66
IQ_verb:denomina2 -0.19289    0.15407   -1.25
IQ_verb:denomina3 -0.37552    0.16440   -2.28
IQ_verb:denomina4 -0.63523    0.30517   -2.08
IQ_verb:denomina5  0.01697    0.26258    0.06

Correlation of Fixed Effects:
            (Intr) IQ_vrb denmn2 denmn3 denmn4 denmn5 IQ_v:2 IQ_v:3 IQ_v:4
IQ_verb      0.066
denomina2   -0.741 -0.049
denomina3   -0.689 -0.045  0.511
denomina4   -0.374 -0.025  0.278  0.258
denomina5   -0.422 -0.028  0.313  0.290  0.158
IQ_vrb:dnm2 -0.048 -0.724  0.027  0.033  0.018  0.020
```

```
IQ_vrb:dnm3 -0.045 -0.678  0.033  0.029  0.017  0.019  0.491
IQ_vrb:dnm4 -0.024 -0.365  0.018  0.017 -0.105  0.010  0.264  0.248
IQ_vrb:dnm5 -0.028 -0.425  0.021  0.019  0.010 -0.058  0.307  0.288  0.155
```

The formula in this model is a combination between Model 6 and Model 7. The random intercept is listed in the first parenthesis to be random across schools. The random slope is listed in the second parenthesis with 0 to tell the program that the covariance between random intercept and random slope is not estimated. The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + IQ_verb + denomina
           + IQ_verb*denomina
           + (1|schoolnr) + (0 + IQ_verb|schoolnr)
langPOST ~ 1 + IQ_verb + d2 + d3 + d4 + d5
           + IQ_verb*d2 + IQ_verb*d3 + IQ_verb*d4 + IQ_verb*d5
           + (1|schoolnr) + (0 + IQ_verb|schoolnr)
```

$Y_{ij} = \gamma_{00}(1) + \gamma_{10}X_{ij} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j}$ $+\gamma_{11}X_{ij}W_{1j} + \gamma_{12}X_{ij}W_{2j} + \gamma_{13}X_{ij}W_{3j} + \gamma_{14}X_{ij}W_{4j}$	Fixed Effect
$+u_{0j}(1) + u_{1j}X_{ij} + e_{ij}$	Random Effect

The output is similar to Model 6 but the partial correlation between the random intercepts and random slopes in the `Random effects` section was not listed. That is, the partial covariance/correlation is fixed to 0. Note that this model should not be used as a final model. Even though the partial covariance/correlation is small, the partial covariance/correlation should be estimated. Otherwise, the fixed correlation can lead to biases in the parameters in the model. Therefore, Model 6 is preferred to Model 10. The reason why this model is listed here because this model can be used to compare with Model 6 by deviance test.

Comparisons between Models

Deviance Test

In the previous section, I illustrated how to run different multilevel models. You may notice that the test statistics for the variance of random effects do not exist. The deviance test (or likelihood ratio test) can be used to test for null hypotheses of any elements in the covariance matrix of random effects, such as τ_{00}, τ_{10}, or τ_{11}. Furthermore, the *z* test (Wald test) mentioned above is not the optimal method for significance testing even for fixed effects. Theoretically, the *z* test is based on the first-order Taylor series approximation of the standard errors of parameter estimates. The deviance test is based on the second-order Taylor series approximation so, in principle, the deviance test should provide more accurate significance testing (Cheung, 2009). Furthermore, the deviance test can be used to compare more than one parameter at once, such as testing the contribution of W_j by testing both γ_{01} and γ_{11} simultaneously.

In R, the deviance test is relatively easy. Users can compare two models by simply using the `anova` function:

```
anova(m1, m4)
```

```
Data: dat
Models:
m1: langPOST ~ 1 + IQ_verb + (1 | schoolnr)
m4: langPOST ~ 1 + IQ_verb + (1 + IQ_verb | schoolnr)
   Df   AIC   BIC logLik  Chisq Chi Df Pr(>Chisq)
m1  4 24920 24945 -12456
```

```
m4   6 24891 24928 -12439 33.382      2  5.639e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The arguments of the `anova` function are only the results from the `lmer` function. The order of objects does not matter—`m4` can be listed before `m1`. In the output, Model 1 was compared with Model 4. The chi-square value is 33.382, which can be computed by the difference between log-likelihood values $\left(-12439-(-12456)\right)$ or the difference between deviances $(24912-24879)$. The degree of freedom is calculated by the difference in the number of estimated parameters. Because τ_{11} and τ_{10} are not estimated in Model 1, the degree of freedom is 2. If the chi-square value and degree of freedom are calculated manually, the *p* value can be calculated by the `pchisq` function:

```
pchisq(33.382, 2, lower.tail=FALSE)
```

```
[1] 5.638853e-08
```

The results from the `anova` function and the `pchisq` function match each other. Note that the most importance things in using the deviance test are 1) to make sure that two models are nested and 2) to realize what we are testing. In comparing between Model 1 and Model 4, the random slopes are compared. Because the effect is significant, the model with random slope is preferred. The following table shows the examples of model comparison using deviance test and its interpretation. The highlighted rows represent the popular uses of deviance test.

Models	Null hypothesis	Result	Interpretation
M0 **vs.** M1	$\gamma_{10}=0$	$\chi^2(1) = 1683.1, p < .001$	The effect of verbal IQ was significant.
M0 **vs.** M2	$\gamma_{01}=0$ $\gamma_{02}=0$ $\gamma_{03}=0$ $\gamma_{04}=0$	$\chi^2(4) = 21.22, p < .001$	The means of language scores across different schools were significantly different.
M1 **vs.** M3	$\gamma_{01}=0$ $\gamma_{02}=0$ $\gamma_{03}=0$ $\gamma_{04}=0$	$\chi^2(4) = 18.12, p = .001$	The adjusted means of language scores (controlling for verbal IQ) across different schools were significantly different.
M1 **vs.** M4	$\tau_{11}=0$ $\tau_{10}=0$	$\chi^2(2) = 33.38, p < .001$	The effect of verbal IQ was random across school and the random slope was related to random intercept.
M1 **vs.** M7	$\tau_{11}=0$	$\chi^2(1) = 8.75, p = .003$	The effect of verbal IQ was significantly random across school.
M2 **vs.** M3	$\gamma_{10}=0$	$\chi^2(1) = 1680, p < .001$	The effect of verbal IQ controlling for the type of schools was significant.
M3 **vs.** M8	$\tau_{11}=0$ $\tau_{10}=0$	$\chi^2(2) = 34.34, p < .001$	The effect of verbal IQ controlling for the type of schools was random across school and the random slope was related to random intercept.
M3 **vs.** M9	$\gamma_{11}=0$ $\gamma_{12}=0$ $\gamma_{13}=0$ $\gamma_{14}=0$	$\chi^2(4) = 11.69, p = .020$	The cross-level interactions between type of schools and verbal IQ were significant in predicting language scores.
M4 **vs.** M5	$\tau_{00}=0$ $\tau_{10}=0$	$\chi^2(2) = 454.81, p < .001$	The expected value of language score when verbal IQ is 0 was varied across schools. The random intercept was related to random slope

Models	Null hypothesis	Result	Interpretation
M4 vs. M6	$\gamma_{01} = 0$ $\gamma_{02} = 0$ $\gamma_{03} = 0$ $\gamma_{04} = 0$ $\gamma_{11} = 0$ $\gamma_{12} = 0$ $\gamma_{13} = 0$ $\gamma_{14} = 0$	$\chi^2(8) = 25.87, p = .001$	The effect of type of schools on random intercept or random slope was significant. That is, type of schools should be added in the model.
M4 vs. M7	$\tau_{10} = 0$	$\chi^2(1) = 24.63, p < .001$	The covariance between random intercept and random slope was significant.
M4 vs. M8	$\gamma_{01} = 0$ $\gamma_{02} = 0$ $\gamma_{03} = 0$ $\gamma_{04} = 0$	$\chi^2(4) = 19.08, p < .001$	The effect of type of schools on random intercept was significant.
M5 vs. M7	$\tau_{00} = 0$	$\chi^2(1) = 430.18, p < .001$	The expected value of language score when verbal IQ is 0 was varied across schools.
M6 vs. M8	$\gamma_{11} = 0$ $\gamma_{12} = 0$ $\gamma_{13} = 0$ $\gamma_{14} = 0$	$\chi^2(4) = 6.79, p = .15$	The cross-level interactions between type of schools and verbal IQ were not significant in predicting language scores.
M6 vs. M9	$\tau_{11} = 0$ $\tau_{10} = 0$	$\chi^2(2) = 29.44, p < .001$	The type of schools did not fully explain the variation between slopes across schools. The random slope was also correlated with random intercepts.
M6 vs. M10	$\tau_{10} = 0$	$\chi^2(1) = 24.11, p < .001$	The type of schools did not fully explain the covariance between random intercepts and random slopes.
M9 vs. M10	$\tau_{11} = 0$	$\chi^2(1) = 5.33, p = .021$	The type of schools did not fully explain the variation between slopes across schools.

Note that some pairs or comparison provided the same test. For example, the test of cross-level interactions can be tested by (Model 3 vs. Model 9) or (Model 6 vs. Model 8). You may notice that the first comparison was significant but the second comparison was not significant. This difference is the reason why, sometimes, build-up strategy and tear-down strategy (which are both exploratory) winds up with different models. Note that, even worse, different experts have different opinions on how to implement those exploratory strategies. Therefore, building a model based on theory is strongly encouraged.

If, at the end of the day, the cross-level interactions must be evaluated, in my opinion, the comparison between Model 6 and Model 8 was more trustworthy. The accuracy of the deviance test decreases when the degree of misspecification of the nested (restricted) model is higher. Model 3 was misspecified because the random slope was not included. The misspecification in Model 3 was much stronger than in Model 8. Thus, I trust the comparison between Model 6 and Model 8 more and I tend to conclude that the cross-level interactions were not significant.

Let's run examples of using build-up strategy and using tear-down strategy and examine how deviance test can be used in these situations. Remember that if two nested models are significantly different from each other, the more complex model (parent model) is preferred. If two nested models are not significantly different from each other, parsimonious model (nested model) is preferred.

Build-up Strategy

Here are the steps of build-up strategy when language score is predicted by IQ score and type of schools.

1. Null model (Model 0)
2. Random intercept, L1 predictor with a fixed slope (Model 1). Model 0 vs. Model 1: $\chi^2(1) = 1683.1$, $p < .001$. Therefore, Model 1 is preferred.
3. Random intercept, L1 and L2 predictors with fixed slope (Model 3). Model 1 vs. Model 3: $\chi^2(4) = 18.12$, $p = .001$. Therefore, Model 3 is preferred.
4. Look for random effects on a variable-by-variable basis (Model 8). Model 3 vs. Model 8: $\chi^2(2) = 34.34$, $p < .001$. Therefore, Model 8 is preferred.
5. Look for cross-level interactions (Model 6). Model 8 vs. Model 6: $\chi^2(4) = 6.79$, $p = .15$. Therefore, Model 8 is preferred.

Tear-down Strategy

Here are the steps of tear-down strategy when language score is predicted by IQ score and type of schools.

1. Full model with cross-level interactions (Model 6).
2. Drop cross-level interactions (Model 8). Model 8 vs. Model 6: $\chi^2(4) = 6.79$, $p = .15$. Therefore, Model 8 is preferred because Model 8 provided equivalent fit to Model 6 with a fewer number of parameters.
3. Drop random effects (Model 3). Model 8 vs. Model 3: $\chi^2(2) = 34.34$, $p < .001$. Model 8 and Model 3 did not have the same amount of model fit. Therefore, Model 8 is preferred because the drop of parameters provided significantly reduction in model fit.

In this case, the build-up and tear-down strategies led to the same conclusions. However, in reality, the numbers of L1 and L2 predictors are much higher. The chance of two strategies winding up to a different model is high.

Other Model Fit Statistics

Deviance test is not the only option for a model comparison. Users may notice that the first section of the output of the `lmer` function provides AIC and BIC for a model comparison. In general, a model with lower AIC or BIC is preferred. Users can compare it manually, which may be very tedious. The following code can be used to aggregate AIC/BIC information into the same table.

First, the model outputs are aggregated into a single object by the `list` function.

```
models <- list(m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10)
```

The `models` object will be saved as a list of outputs. Because we have a list of similar objects, a function can be applied to all elements in the list simultaneously. In this case, the `summary` function is applied to each element. The function that helps us to apply the `summary` function elementwise is the `lapply` function:

```
models.summary <- lapply(models, summary)
```

The first argument is a list of similar objects. The second argument is the function to be applied elementwise. The `models.summary` object will be a list of the summaries of all 11 objects. Next, we would like to extract the model fit information from the summaries of all objects. The model fit information can be extracted from the `"AICtab"` slot of the summaries. For a single output, the `slot` function can be used:

```
m0sum <- summary(m0)

slot(m0sum, name = "AICtab")
```

```
      AIC      BIC    logLik deviance  REMLdev 
 26601.28 26619.98 -13297.64 26595.28 26595.69
```

However, we wish to get model fit from all outputs in a list at the same time. The `lapply` function could be used to get the model fit results simultaneously.

```
lapply(models.summary, slot, name = "AICtab")
```

```
[[1]]
      AIC      BIC    logLik deviance  REMLdev 
 26601.28 26619.98 -13297.64 26595.28 26595.69 

[[2]]
      AIC      BIC    logLik deviance  REMLdev 
 24920.17 24945.09 -12456.08 24912.17 24917.14
…
```

The first argument is the list of similar object. The second argument is the function. Then, additional arguments that users wish to pass to the specified elementwise function (`slot` in this case) can be listed in the third or the following arguments.

The output is quite inconvenient to deal with. The results would be easier to handle if they were displayed in a table format. Instead of the `lapply` function, the `sapply` function can be used to reduce the result into a table format:

```
sapply(models.summary, slot, "AICtab")
```

```
         [,1]      [,2]      [,3]      [,4]      [,5]      [,6]     [,7]      [,8]      [,9]      [,10]     [,11]    
AIC      26601.28  24920.17  26588.06  24910.05  24890.79  25341.59 24880.92  24913.42  24879.71  24906.36  24903.03 
BIC      26619.98  24945.09  26631.68  24959.9   24928.18  25366.52 24968.16  24944.58  24942.03  24981.14  24984.04 
logLik   -13297.64 -12456.08 -13287.03 -12447.03 -12439.39 -12666.8 -12426.46 -12451.71 -12429.85 -12441.18 -12438.51
deviance 26595.28  24912.17  26574.06  24894.05  24878.79  25333.59 24852.92  24903.42  24859.71  24882.36  24877.03 
REMLdev  26595.69  24917.14  26567.16  24893.92  24883.6   25339.57 24858.85  24908.1   24860.11  24888.71  24882.21
```

To make a nicer table, column names can be added to the table:

```
modelfit <- sapply(models.summary, slot, "AICtab")

colnames(modelfit) <- c("m0", "m1", "m2", "m3", "m4", "m5", "m6", "m7", "m8", "m9", "m10")

modelfit
```

```
         m0        m1        m2        m3        m4        m5       m6        m7        m8        m9        m10      
AIC      26601.28  24920.17  26588.06  24910.05  24890.79  25341.59 24880.92  24913.42  24879.71  24906.36  24903.03 
BIC      26619.98  24945.09  26631.68  24959.9   24928.18  25366.52 24968.16  24944.58  24942.03  24981.14  24984.04 
logLik   -13297.64 -12456.08 -13287.03 -12447.03 -12439.39 -12666.8 -12426.46 -12451.71 -12429.85 -12441.18 -12438.51
deviance 26595.28  24912.17  26574.06  24894.05  24878.79  25333.59 24852.92  24903.42  24859.71  24882.36  24877.03 
REMLdev  26595.69  24917.14  26567.16  24893.92  24883.6   25339.57 24858.85  24908.1   24860.11  24888.71  24882.21
```

You will notice that Model 8 provided the lowest AIC whereas Model 4 provided the lowest BIC. The decision of using AIC or BIC is not a consensus. Hox (2010) proposed that BIC has a slightly better performance whereas Vrieze (2012) argued that BIC, in most case, is inappropriate.

Proportion of Variance Explained

Proportion of Dependent Variable's Score Variance Explained

The proportion of total variance explained by predictors can be computed from the amount of residual variance reduced when predictors are included in a model. For example, from the null model (Model 0), verbal IQ was included to create Model 1. Users can calculate the amount of language score variances explained by verbal IQ. The proportion of variance explained (R^2) can be computed by the following formulas:

$$R_1^2 = \frac{\sigma_e^2 - \sigma_{(e|X,W)}^2}{\sigma_e^2}$$

$$R_{\beta_{0j}}^2 = \frac{\tau_{00} - \tau_{(00|X,W)}}{\tau_{00}}$$

where "$|X, W$" means given L1 predictors (X) and L2 predictors (W) are included in the model. R_1^2 is the proportion of DV variance explained at the lower level. $R_{\beta_{0j}}^2$ is the proportion of DV variance explained at the upper level. Note that R_1^2 and $R_{\beta_{0j}}^2$ can be greater than 1 or less than 0, which are not a good property. Snijders and Bosker (2011) proposed the corrected formula for R_1^2 and $R_{\beta_{0j}}^2$:

$$\tilde{R}_1^2 = \frac{\sigma_e^2 + \tau_{00} - \sigma_{(e|X,W)}^2 - \tau_{(00|X,W)}}{\sigma_e^2 + \tau_{00}}$$

$$\tilde{R}_{\beta_{0j}}^2 = \frac{\tau_{00} + \frac{\sigma_e^2}{\tilde{n}} - \tau_{(00|X,W)} - \frac{\sigma_{(e|X,W)}^2}{\tilde{n}}}{\tau_{00} + \frac{\sigma_e^2}{\tilde{n}}}$$

where

$$\tilde{n} = \frac{J}{\frac{1}{n_1} + \frac{1}{n_2} + \cdots + \frac{1}{n_J}}$$

where $\tilde{n}$ is the harmonic mean of group sizes, J is the number of groups, and n_j is the size of Group j. Let's run an example on finding the proportion of total variances of language scores explained by both verbal IQ and type of schools. That is, Model 0 is compared with Model 3.

Remember that τ_{00} and σ^2 were extracted in the ICC calculation. The similar codes can be applied here:

```
out0 <- summary(m0)

ranef0 <- out0@REmat

tau00 <- as.numeric(ranef0[1, 3])

sigma2 <- as.numeric(ranef0[2, 3])

out3 <- summary(m3)
```

```
ranef3 <- out3@REmat
tau00_3 <- as.numeric(ranef3[1, 3])
sigma2_3 <- as.numeric(ranef3[2, 3])
```

R_1^2 can be calculated:

```
(sigma2 - sigma2_3) / sigma2
```

```
[1] 0.3560309
```

$R_{\beta_{0j}}^2$ can be calculated:

```
(tau00 - tau00_3) / tau00
```

```
[1] 0.5149297
```

To calculate $\tilde{R}_1^2$ and $\tilde{R}_{\beta_{0j}}^2$, the harmonic mean needs to be calculated:

```
groupsize <- table(dat$schoolnr)
J <- length(groupsize)
denominator <- sum(1/groupsize)
harmonic.n <- J/denominator
harmonic.n
```

```
[1] 14.38582
```

The `table` function is used to find the frequency (group size) of each school in the data set. Because the result of the `table` function is the frequency of each school, the number of schools, `J`, is calculated by the length of the frequency table by the `length` function.

$\tilde{R}_1^2$ can be calculated:

```
(sigma2 + tau00 - sigma2_3 - tau00_3) / (sigma2 + tau00)
```

```
[1] 0.3915975
```

$\tilde{R}_{\beta_{0j}}^2$ can be calculated:

```
(tau00 + (sigma2/harmonic.n) - tau00_3 - (sigma2_3/harmonic.n)) / (tau00 + (sigma2/harmonic.n))
```

```
[1] 0.4840671
```

Note that the comparing models should not have any random slopes. If the random slopes exist, all R^2 listed above are not interpretable because 1) τ_{00} depends on the centering of the predictors and 2) the strength of effect of L1 predictors are variable across groups (leading to different R^2 across groups).

Proportion of Slope Variance Explained

Because the slopes of L1 predictors are allowed to vary across L2 units in multilevel models, some predictors can be used to explain the variance of slopes. The drop in residual variance can be used to calculate the proportion of slope variance explained ($R_{\beta_{1j}}^2$) such that

$$R_{\beta_{1j}}^2 = \frac{\tau_{11} - \tau_{(11|W)}}{\tau_{11}}$$

For example, the proportion of the slope variance explained by the type of school can be calculated by comparing Model 8 and Model 6. First, the residual variance of the slope of Model 6 and Model 8 can be extracted:

```
out6 <- summary(m6)

ranef6 <- out6@REmat

tau11r <- as.numeric(ranef6[2, 3])

out8 <- summary(m8)

ranef8 <- out8@REmat

tau11 <- as.numeric(ranef8[2, 3])
```

$R^2_{\beta_{1j}}$ can be calculated:

```
(tau11 - tau11r) / tau11
```

```
[1] 0.1834728
```

Centering

Sometimes, the IV value of 0 is not meaningful. For example, the standard IQ score of 0 does not exist. Therefore, any intercepts based on the standard IQ score are not meaningful, including γ_{00}, γ_{10}, or β_{0j}. Centering the IV can make intercepts meaningful. Both L1 and L2 predictors can be centered but centering L1 predictor is more complex and crucial. More importantly, different centering approaches are appropriate for different situations. Model 1-10 analyzed above may be not appropriate in some situations.

Centering is simply to subtract an IV by a value. For L1 predictors, the centered value can be grand mean, group mean, or any meaningful values. For L2 predictors, the centered value can be grand mean or any meaningful values. I start with centering L1 predictors with grand mean and group mean. Then, centering at L2 predictors will be illustrated.

Model 1a: Grand Mean Centering / Centering for Specific Values at L1 Predictors

This model is similar to Model 1 that the language score is predicted by verbal IQ score. However, the verbal IQ is grand-mean centered by subtracting the verbal IQ score by its grand mean. The model would be

L1	$Y_{ij} = \beta_{0j} + \beta_{1j}(X_{ij} - \bar{X}_{..}) + e_{ij}$	$e_{ij} \sim N(0, \sigma^2)$
L2	$\beta_{0j} = \gamma_{00} + u_{0j}$	$u_{0j} \sim N(0, \tau_{00})$
	$\beta_{1j} = \gamma_{10}$	

These notations should represent (the blue lines indicate that the meanings changed from Model 1)

- Y_{ij} = The language score of Student *i* in School *j*
- X_{ij} = The verbal IQ score of Student *i* in School *j*
- β_{0j} = The expected mean of language scores within School *j* when the verbal IQ score is equal to its grand mean, which is also referred to as *adjusted mean*

- β_{1j} = The expected change in language score when the verbal IQ score of students in School *j* increases by 1. In this case, the expected changes across schools are the same.
- γ_{00} = The expected average language score across all schools when the verbal IQ score is equal to its grand mean.
- γ_{10} = The schools' average expected change in language score when the verbal IQ score increase by 1.
- e_{ij} = The difference between the actual language score and the predicted language score of Student *i* in School *j*
- u_{0j} = The deviation of the adjusted language score average of School *j* (when the verbal IQ score is its grand mean) from the grand mean of adjusted averages across schools
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the verbal IQ score
- τ_{00} = The language score residual variance across schools (L2 residual variance) controlling for the verbal IQ score

To implement the grand mean centering, a new variable is created by subtracting the `IQ_verb` by its grand mean.

```
dat$IQ_verb.grandMC <- dat$IQ_verb - mean(dat$IQ_verb)
```

Users may save the resulting variable into the same variable (`dat$IQ_verb`) instead of the new variable (`dat$IQ_verb.grandMC`). I recommend making a new variable so it is easier to go back if you need the original values. The model can be run:

```
m1a <- lmer(langPOST ~ 1 + IQ_verb.grandMC + (1|schoolnr), data = dat, REML=FALSE)

summary(m1a)
```

<u>Model 1</u>: No Centering	<u>Model 1a</u>: Grand Mean Centering

`Linear mixed model fit by maximum likelihood`
`Formula: langPOST ~ 1 + IQ_verb + (1 | schoolnr)`
`   Data: dat`
`   AIC   BIC logLik deviance REMLdev`
` 24920 24945 -12456    24912   24917`
`Random effects:`
` Groups   Name        Variance Std.Dev.`
` schoolnr (Intercept)  9.8451  3.1377`
` Residual             40.4689  6.3615`
`Number of obs: 3758, groups: schoolnr, 211`
 	
`Fixed effects:`
`            Estimate Std. Error t value`
`(Intercept) 41.05490    0.24336  168.70`
`IQ_verb      2.50745    0.05438   46.11`
 	
`Correlation of Fixed Effects:`
`        (Intr)`
`IQ_verb 0.003`
	`Linear mixed model fit by maximum likelihood`
	`Formula: langPOST ~ 1 + IQ_verb.grandMC + (1 | schoolnr)`
	`   Data: dat`
	`   AIC   BIC logLik deviance REMLdev`
	` 24920 24945 -12456    24912   24917`
	`Random effects:`
	` Groups   Name        Variance Std.Dev.`
	` schoolnr (Intercept)  9.8451  3.1377`
	` Residual             40.4689  6.3615`
	`Number of obs: 3758, groups: schoolnr, 211`
	
	`Fixed effects:`
	`                Estimate Std. Error t value`
	`(Intercept)     41.16569    0.24338  169.14`
	`IQ_verb.grandMC  2.50745    0.05438   46.11`
	
	`Correlation of Fixed Effects:`
	`            (Intr)`
	`IQ_vrb.grMC 0.013`

The formula is similar to <u>Model 1</u> but the centered variable (`IQ_verb.grandMC`) is included instead. I provide the results with both no centering and grand-mean centering. Only γ_{00} is different across models.[5] Note that if users include random slopes into the model, τ_{00} and τ_{01} can be different from the results without centering. I will let you run it on your own.

[5] The correlation of fixed effects can be different across models but we do not usually interpret it.

If users wish to center the verbal IQ scores at some specific value (e.g., 5), users can simply change the `mean` function to a specific value in the code

```
dat$IQ_verb.c5 <- dat$IQ_verb - 5
```

The interpretation in the model is similar to grand mean centering. The adjusted mean would be interpreted as the adjusted mean when verbal IQ score is 5.

Model 1b: Group Mean Centering at L1 Predictors

Group mean centering is not simply to subtract a constant (e.g., grand mean or a specific value) from a variable. Group mean values, obviously, have different values across groups. Therefore, the properties of the group-mean centered variable are different from the no-centered variable. Users may view group mean centering as changing L1 predictor to a within-group deviation component. The group mean can be added to the model as a L2 predictor representing between-group deviation.

$$X_{ij} = \bar{X}_{.j} + (X_{ij} - \bar{X}_{.j})$$

Total deviation = Between-group deviation + Within-group deviation

where

- X_{ij} = The verbal IQ score of Student *i* in School *j*
- $\bar{X}_{.j}$ = The average of verbal IQ score across students in School *j*

The model would be

L1	$Y_{ij} = \beta_{0j} + \beta_{1j}(X_{ij} - \bar{X}_{.j}) + e_{ij}$	$e_{ij} \sim N(0, \sigma^2)$
L2	$\beta_{0j} = \gamma_{00} + \gamma_{01}\bar{X}_{.j} + u_{0j}$	$u_{0j} \sim N(0, \tau_{00})$
	$\beta_{1j} = \gamma_{10}$	

These notations should represent (the blue lines indicate that the meanings changed from Model 1)

- Y_{ij} = The language score of Student *i* in School *j*
- β_{0j} = The average of language score within School *j*, which is also referred to as *unadjusted mean*
- β_{1j} = The expected change in language score when the verbal IQ score of students in School *j* increases by 1 controlling for schools (assuming that school membership is known), which is also referred to *within-group effect*.
- γ_{00} = The expected school average language score when the average school verbal IQ score is 0.
- γ_{01} = The expected change in school average language score when the school average of verbal IQ score increases by 1, which is also referred to *between-group effect*.
- γ_{10} = The within-group effect of the verbal IQ score, which is constant across schools
- e_{ij} = The difference between the actual language score and the predicted language score of Student *i* in School *j*
- u_{0j} = The deviation of the actual unadjusted language score average of School *j* from the predicted language school average (from the mean of verbal IQ score)
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the verbal IQ score

- τ_{00} = The residual variance of the school average language scores across schools controlling for the school average verbal IQ score

To implement the group mean centering, a new variable is created to represent a group mean and then an original variable is subtracted by the group mean.

```
dat$IQ_verb.groupMean <- ave(dat$IQ_verb, dat$schoolnr)

dat$IQ_verb.groupMC <- dat$IQ_verb - dat$IQ_verb.groupMean
```

The `ave` function is used to create a variable of group mean where the first argument is the target variable and the second argument is the grouping variable. The model can be run:

```
m1b <- lmer(langPOST ~ 1 + IQ_verb.groupMC + dat$IQ_verb.groupMean + (1|schoolnr), data = dat,
REML=FALSE)

summary(m1b)
```

Model 1: No Centering	Model 1b: Group Mean Centering

<pre>Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + (1 | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24920 24945 -12456   24912   24917
Random effects:
 Groups   Name   Variance Std.Dev.
 schoolnr (Intercept) 9.8451 3.1377
 Residual   40.4689 6.3615
Number of obs: 3758, groups: schoolnr, 211
 	
Fixed effects:
            Estimate Std. Error t value
(Intercept) 41.05490   0.24336 168.70
IQ_verb   2.50745   0.05438   46.11
 	
Correlation of Fixed Effects:
        (Intr)
IQ_verb 0.003</pre>
	<pre>Linear mixed model fit by maximum likelihood
	Formula: langPOST ~ 1 + IQ_verb.groupMC +
	dat$IQ_verb.groupMean + (1 | schoolnr)
	   Data: dat
	   AIC BIC logLik deviance REMLdev
	 24899 24930 -12445 24889 24895
	Random effects:
	 Groups Name Variance Std.Dev.
	 schoolnr (Intercept) 8.7246 2.9537
	 Residual 40.4318 6.3586
	Number of obs: 3758, groups: schoolnr, 211
	
	Fixed effects:
	                      Estimate Std. Error t value
	(Intercept) 41.07585 0.23205 177.01
	IQ_verb.groupMC 2.45412 0.05552 44.20
	dat$IQ_verb.groupMean 3.73710 0.25553 14.62
	
	Correlation of Fixed Effects:
	            (Intr) IQ_.MC
	IQ_vrb.grMC 0.000
	dt$IQ_vrb.M 0.011 0.000</pre>

Users may notice that the models are not equivalent. Model fits, fixed effects, and the variances of random effects are different. Thus, researchers should use the theory to decide the choices of centering. I recommend Enders and Tofighi (2007) for a further reading about centering. Note that users may add or not add the group mean (`IQ_verb.groupMean`) back at the upper level.

Model 11: Centering Level-2 Predictor

L2 predictors can be centered at grand mean or a specific value. L2 predictors include natural L2 predictors (e.g., school type) and the group means from L1 predictors. In this example, language score is predicted by verbal IQ scores, which is group-mean centered, and proportion minority in each school. The within-group effect of the verbal IQ scores is random across schools. The group mean of verbal IQ scores is centered at the grand mean and the proportion minority is centered at 0.50. The model would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}(X_{ij} - \bar{X}_{.j}) + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + \gamma_{01}(\bar{X}_{.j} - \bar{X}_{..}) + \gamma_{02}(W_j - 0.5) + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + \gamma_{11}(\bar{X}_{.j} - \bar{X}_{..}) + \gamma_{12}(W_j - 0.5) + u_{1j}$$
$$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ \tau_{10} & \tau_{11} \end{bmatrix}\right)$$

These notations should represent

- Y_{ij} = The language score of Student *i* in School *j*
- X_{ij} = The verbal IQ score of Student *i* in School *j*
- $\bar{X}_{.j}$ = The mean of verbal IQ score in School *j*
- $\bar{X}_{..}$ = The grand mean of verbal IQ score
- W_j = The proportion of minority in School *j*
- β_{0j} = The unadjusted mean of language score in School *j*
- β_{1j} = The within-school effect of verbal IQ score on language score in School *j*
- γ_{00} = The expected school average language score when the verbal IQ score equals to its grand mean and the proportion minority equals 0.5
- γ_{01} = The between-school effect of verbal IQ score on language score controlling for the proportion minority
- γ_{02} = The effect of proportion minority on the language score controlling for the verbal IQ score
- γ_{10} = The expected within-school effect of verbal IQ score on language score when the school average verbal IQ score equals the grand mean and the proportion minority equals 0.5.
- γ_{11} = The change in the within-school effect when the school average verbal IQ score increases by 1 controlling for the proportion minority
- γ_{12} = The change in the within-school effect when the proportion minority increases by 1 controlling for the school average verbal IQ score
- e_{ij} = The difference between the actual language score and the predicted language score of Student *i* in School *j*
- u_{0j} = The deviation of the actual unadjusted language score average of School *j* from the predicted language school average (from the mean of verbal IQ score and proportion minority)
- u_{1j} = The deviation of the actual within-group effect of School *j* from the predicted within-group effect (from the mean of verbal IQ score and proportion minority)
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the verbal IQ score
- τ_{00} = The language score residual variance across schools (L2 residual variance) controlling for the verbal IQ score and proportion minority
- τ_{11} = The residual variance of the slope of within-group effect of verbal IQ score across schools controlling for the school average verbal IQ score and proportion minority
- τ_{10} = The covariance between the residual of the random intercept and the residual of the random slope
- $\rho_{10} = \tau_{10}/\sqrt{\tau_{00}\tau_{11}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)

Following the code from Model 1b, we have `IQ_verb.groupMC` representing group means `IQ_verb.groupMC` representing verbal IQ centered at the group means. First, the group mean is centered at the grand mean:

```
dat$IQ_verb.groupMeanC <- dat$IQ_verb.groupMean - mean(dat$IQ_verb.groupMean)
```

Next, the proportion minority is centered at 0.50:

```
dat$sch_min.grandMC <- dat$sch_min - mean(dat$sch_min)
```

The model can be run:

```
m11 <- lmer(langPOST ~ 1 + IQ_verb.groupMC + IQ_verb.groupMeanC + sch_min.grandMC +
IQ_verb.groupMC*IQ_verb.groupMeanC + IQ_verb.groupMC*sch_min.grandMC + (1 +
IQ_verb.groupMC|schoolnr), data = dat, REML=FALSE)

summary(m11)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb.groupMC + IQ_verb.groupMeanC + sch_min.grandMC +      IQ_verb.groupMC *
IQ_verb.groupMeanC + IQ_verb.groupMC *      sch_min.grandMC + (1 + IQ_verb.groupMC | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24874 24936 -12427    24854   24860
Random effects:
 Groups   Name            Variance Std.Dev. Corr
 schoolnr (Intercept)      8.81638 2.96924
          IQ_verb.groupMC  0.16638 0.40789  -0.682
 Residual                 39.67709 6.29898
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
                                   Estimate Std. Error t value
(Intercept)                        41.23163    0.23295  177.00
IQ_verb.groupMC                     2.48311    0.06347   39.12
IQ_verb.groupMeanC                  3.70256    0.27357   13.53
sch_min.grandMC                    -0.25114    1.82997   -0.14
IQ_verb.groupMC:IQ_verb.groupMeanC -0.15700    0.07399   -2.12
IQ_verb.groupMC:sch_min.grandMC    -1.14698    0.43359   -2.65

Correlation of Fixed Effects:
              (Intr) IQ_verb.grpMC IQ_vrb.grpMnC sc_.MC IQ_.MC:I
IQ_verb.grpMC -0.277
IQ_vrb.grpMnC  0.047 -0.015
sch_mn.grMC   -0.024  0.008         0.356
IQ_.MC:IQ_.   -0.016  0.057        -0.267        -0.107
IQ_.MC:_.MC    0.009 -0.082        -0.119        -0.312  0.272
```

The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + IQ_verb.groupMC + IQ_verb.groupMeanC + sch_min.grandMC
           + IQ_verb.groupMC*IQ_verb.groupMeanC
           + IQ_verb.groupMC*sch_min.grandMC
           + (1 + IQ_verb.groupMC|schoolnr)
```

$$Y_{ij} = \gamma_{00}(1) + \gamma_{10}(X_{ij} - \bar{X}_{.j}) + \gamma_{01}(\bar{X}_{.j} - \bar{X}_{..}) + \gamma_{02}(W_j - 0.5)$$
$$+ \gamma_{11}(X_{ij} - \bar{X}_{.j})(\bar{X}_{.j} - \bar{X}_{..}) \quad \text{Fixed Effect}$$
$$+ \gamma_{12}(X_{ij} - \bar{X}_{.j})(W_j - 0.5)$$
$$+ u_{0j}(1) + u_{1j}(X_{ij} - \bar{X}_{.j}) + e_{ij} \quad \text{Random Effect}$$

Users are encouraged to run the model without centering at L2 predictors. Compare the results with the current model. Users will find that only γ_{00} and γ_{10} change the values. Other parameters and model fits remain the same.

Testing Interactions

In multilevel model, two-way interactions can be classified by the level of your predictors:

1. Both predictors are in L1. In this case, users have an option to make the lower-level interaction to be random across L2 units.
2. Both predictors are in L2. The interaction is fixed across school.
3. One predictor is in L1 but the other predictor is in L2 (cross-level interaction). This model has been discussed in Model 6 already.

The code for modeling interaction in multilevel model is similar to regression analysis. Researchers only need to specify the product term in the formula by asterisk, `*`, or colon, `:` (as mentioned in Model 6). In this section, the examples of each type of interactions will be discussed. After that, the probing interaction and testing simple slopes will be discussed.

Model 12: Lower-level Interaction

In this model, the language scores (`langPOST`) is predicted by verbal IQ (`IQ_verb`) and socioeconomic status (`ses`). Both predictors are expected to have interactive effect on the language score. All regression coefficients at level 1 are random across schools. The model with lower-level interaction would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}X_{1ij} + \beta_{2j}X_{2ij} + \beta_{3j}\left(X_{1ij} \cdot X_{2ij}\right) + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\begin{aligned} \beta_{0j} &= \gamma_{00} + u_{0j} \\ \beta_{1j} &= \gamma_{10} + u_{1j} \\ \beta_{2j} &= \gamma_{20} + u_{2j} \\ \beta_{1j} &= \gamma_{30} + u_{3j} \end{aligned} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \\ u_{2j} \\ u_{3j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & & & \\ \tau_{10} & \tau_{11} & & \\ \tau_{20} & \tau_{21} & \tau_{22} & \\ \tau_{30} & \tau_{31} & \tau_{32} & \tau_{33} \end{bmatrix}\right)$$

As a good practice, the target variable is specified first and then the moderator is specified. This practice will make the interpretation and probing interaction easier. In this case, verbal IQ is a target variable and socioeconomic status is a moderator. These notations should represent

- Y_{ij} = The language score of Student *i* in School *j*
- X_{1ij} = The verbal IQ score of Student *i* in School *j*
- X_{2ij} = The socioeconomic status of Student *i* in School *j*
- β_{0j} = The expected average of language score within School *j* when the verbal IQ score and socioeconomic status are 0.
- β_{1j} = The expected change in language score when the verbal IQ score of students in School *j* increases by 1 given that the socioeconomic status is 0. It can be referred to as the simple slope of verbal IQ score when the socioeconomic status is 0.
- β_{2j} = The expected change in language score when the socioeconomic status of students in School *j* increases by 1 given that the verbal IQ is 0. It can be referred to as the simple slope of socioeconomic status when the verbal IQ is 0.
- β_{3j} = The change in the effect of verbal IQ scores on language scores when socioeconomic status increases by 1 in School *j*. Also, the change in the effect of socioeconomic status on language scores when verbal IQ scores increases by 1 in School *j*. It can be referred to as the interaction effect in School *j*.
- γ_{00} = The average language score across all schools when the verbal IQ score and socioeconomic status are 0.
- γ_{10} = The schools' average expected change in language score when the verbal IQ score increase by 1 given that socioeconomic status is 0.
- γ_{20} = The schools' average expected change in language score when the socioeconomic status increase by 1 given that verbal IQ score is 0.
- γ_{30} = The average of the interaction effect across schools.

- e_{ij} = The difference between the actual language score and the predicted language score of Student i in School j.
- u_{0j} = The deviation of the adjusted language score average of School j (when the verbal IQ score and socioeconomic status are 0) from the average of adjusted means across schools
- u_{1j} = The deviation of the simple slope of verbal IQ score (when socioeconomic status is 0) of School j from the average simple slope of verbal IQ score across schools
- u_{2j} = The deviation of the simple slope of socioeconomic status (when verbal IQ score is 0) of School j from the average simple slope of socioeconomic status across schools
- u_{3j} = The deviation of the interaction effect of School j from the average interaction effect across schools
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the verbal IQ score and socioeconomic status
- τ_{00} = The language score residual variance across schools (L2 residual variance) controlling for the verbal IQ score and socioeconomic status
- τ_{11} = The variance of the simple slope of verbal IQ score across schools
- τ_{22} = The variance of the simple slope of socioeconomic status across schools
- τ_{33} = The variance of the interaction effect between verbal IQ score and socioeconomic status across schools
- τ_{st} (where $s \neq t$) = The covariance between u_{sj} and u_{tj}
- $\rho_{st} = \tau_{st}/\sqrt{\tau_{ss}\tau_{tt}}$ (where $s \neq t$) = The covariance mentioned above in the correlation scale (from -1 to 1)

The model with lower-level interaction can be run by the `lmer` function:

```
m12 <- lmer(langPOST ~ 1 + IQ_verb + ses + IQ_verb*ses + (1 + IQ_verb + ses +
IQ_verb*ses|schoolnr), data = dat, REML=FALSE)

summary(m12)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb + ses + IQ_verb * ses + (1 + IQ_verb +      ses + IQ_verb * ses | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24662 24756 -12316    24632   24653
Random effects:
 Groups   Name        Variance   Std.Dev. Corr
 schoolnr (Intercept) 1.0285e+01 3.206989
          IQ_verb     2.0144e-01 0.448817 -0.671
          ses         2.2252e-04 0.014917  0.464 -0.968
          IQ_verb:ses 6.4380e-04 0.025373 -0.763  0.031  0.219
 Residual             3.6978e+01 6.080966
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
             Estimate Std. Error t value
(Intercept) 41.282941   0.249130  165.71
IQ_verb      2.251518   0.064446   34.94
ses          0.173129   0.011303   15.32
IQ_verb:ses -0.021280   0.005119   -4.16

Correlation of Fixed Effects:
            (Intr) IQ_vrb ses
IQ_verb     -0.312
ses          0.070 -0.338
IQ_verb:ses -0.376  0.045 -0.165
```

Note that all effects at L1 are listed in the parenthesis so all L1 effects are random. The mapping from the formula and reduced-form equation would be

`langPOST ~ 1 + IQ_verb + ses + IQ_verb*ses` `+ (1 + IQ_verb + ses + IQ_verb*ses|schoolnr)`	
$Y_{ij} = \gamma_{00}(1) + \gamma_{10}X_{1ij} + \gamma_{20}X_{2ij} + \gamma_{30}(X_{1ij} \cdot X_{2ij})$	Fixed Effect
$+u_{0j}(1) + u_{1j}X_{1ij} + u_{2j}X_{2ij} + u_{3j}(X_{1ij} \cdot X_{2ij}) + e_{ij}$	Random Effect

The output is similar to the previous models except there are six correlation coefficients between four random effects. Similar to previous models, p-value and residual ICC can be computed. In this case, the significance of the interaction effect could be directly observed from the p value from the t-statistic. The interaction effect was significant (z = -4.16, p < .001). Users can use deviance test to check the significance of the interaction effect or the significance of the variance of random interaction effect across schools.

Model 12a: Lower-level Interaction with Group Mean Centering

(*Readers may skip this section without loss of continuity) Grand mean centering may be not appropriate with the lower-level interaction (Enders & Tofighi, 2007). In no centering or grand-mean centering, both between- and within-level effect of verbal IQ score and socioeconomic status will be accounted in β_{1j}, β_{2j}, and β_{3j}. The between-level effects are constant across schools but the within-level effects are random across schools. Because of the constant between-level effect across schools, τ_{11}, τ_{22}, and τ_{33} can be underestimated. Therefore, group-mean centering is more appropriate in lower-level interaction although the computation can be much more complex. In this example, all L1 predictors are centered at the group means. The group means of both L1 predictors are added back as the L2 predictors with grand mean centering. The model would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}(X_{1ij} - \bar{X}_{1.j}) + \beta_{2j}(X_{2ij} - \bar{X}_{2.j}) + \beta_{3j}(X_{1ij} - \bar{X}_{1.j}) \cdot (X_{2ij} - \bar{X}_{2.j}) + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\begin{aligned}
\beta_{0j} &= \gamma_{00} + \gamma_{01}(\bar{X}_{1.j} - \bar{X}_{1..}) + \gamma_{02}(\bar{X}_{2.j} - \bar{X}_{2..}) + u_{0j} \\
\beta_{1j} &= \gamma_{10} + \gamma_{11}(\bar{X}_{1.j} - \bar{X}_{1..}) + \gamma_{12}(\bar{X}_{2.j} - \bar{X}_{2..}) + u_{1j} \\
\beta_{2j} &= \gamma_{20} + \gamma_{21}(\bar{X}_{1.j} - \bar{X}_{1..}) + \gamma_{22}(\bar{X}_{2.j} - \bar{X}_{2..}) + u_{2j} \\
\beta_{1j} &= \gamma_{30} + \gamma_{31}(\bar{X}_{1.j} - \bar{X}_{1..}) + \gamma_{32}(\bar{X}_{2.j} - \bar{X}_{2..}) + u_{3j}
\end{aligned}
\qquad
\begin{bmatrix} u_{0j} \\ u_{1j} \\ u_{2j} \\ u_{3j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & & & \\ \tau_{10} & \tau_{11} & & \\ \tau_{20} & \tau_{21} & \tau_{22} & \\ \tau_{30} & \tau_{31} & \tau_{32} & \tau_{33} \end{bmatrix}\right)$$

These notations should represent

- Y_{ij} = The language score of Student i in School j
- X_{1ij} = The verbal IQ score of Student i in School j
- $\bar{X}_{1.j}$ = The verbal IQ score average in School j
- $\bar{X}_{1..}$ = The grand mean of verbal IQ score
- X_{2ij} = The socioeconomic status of Student i in School j
- $\bar{X}_{2.j}$ = The socioeconomic status average in School j
- $\bar{X}_{2..}$ = The grand mean of socioeconomic status
- β_{0j} = The unadjusted mean of language score within School j
- β_{1j} = The within-group effect of verbal IQ score in School j given that the socioeconomic status is equal to its group mean. It can be referred to as the simple slope of verbal IQ score when the socioeconomic status is equal to its group mean.

- β_{2j} = The within-group effect of socioeconomic status in School *j* given that the verbal IQ score is equal to its group mean. It can be referred to as the simple slope of socioeconomic status when the verbal IQ score is equal to its group mean.
- β_{3j} = The change in the within-group effect of verbal IQ scores when socioeconomic status increases by 1 in School *j*. Also, the change in the within-group effect of socioeconomic status when verbal IQ scores increases by 1 in School *j*. It can be referred to as the interaction of within-group effect in School *j*.
- γ_{00} = The expected language score across all schools when the verbal IQ score and socioeconomic status are equal to their grand mean. The value will be equal to the grand mean of the language score.
- γ_{01} = The between-group effect of verbal IQ score controlling for the school mean of socioeconomic status
- γ_{02} = The between-group effect of socioeconomic status controlling for the school mean of verbal IQ score
- γ_{10} = The expected within-group effect of verbal IQ score when the school means of verbal IQ score and socioeconomic status are equal to their grand mean. The value will be equal to the average of the within-group effect of verbal IQ score across schools.
- γ_{11} = The increase in the within-group effect of verbal IQ score when the school mean of verbal IQ score increases by 1, controlling for the school mean of socioeconomic status
- γ_{12} = The increase in the within-group effect of verbal IQ score when the school mean of socioeconomic status increases by 1, controlling for the school mean of verbal IQ score
- γ_{20} = The expected within-group effect of socioeconomic status when the school means of verbal IQ score and socioeconomic status are equal to their grand mean. The value will be equal to the average of the within-group effect of socioeconomic status across schools.
- γ_{21} = The increase in the within-group effect of socioeconomic status when the school mean of verbal IQ score increases by 1, controlling for the school mean of socioeconomic status
- γ_{22} = The increase in the within-group effect of socioeconomic status when the school mean of socioeconomic status increases by 1, controlling for the school mean of verbal IQ score
- γ_{30} = The expected within-group (lower-level) interaction when school means of verbal IQ scores and socioeconomic status equal their grand means. The value will be equal to the average within-group interaction across schools.
- γ_{31} = The increase in the lower-level interaction when the school mean of verbal IQ score increases by 1, controlling for the school mean of socioeconomic status
- γ_{32} = The increase in the lower-level interaction when the school mean of socioeconomic status increases by 1, controlling for the school mean of verbal IQ score
- e_{ij} = The difference between the actual language score and the predicted language score of Student *i* in School *j*.
- u_{0j} = The deviation of the actual average of language score in School *j* from the predicted language score in School *j* (from the school values of verbal IQ score and socioeconomic status)
- u_{1j} = The deviation of the actual within-group effect of verbal IQ score in School *j* from the predicted within-group effect in School *j* (from the school values of verbal IQ score and socioeconomic status)

- u_{2j} = The deviation of the actual within-group effect of socioeconomic status in School j from the predicted within-group effect in School j (from the school values of verbal IQ score and socioeconomic status)
- u_{3j} = The deviation of the lower-level interaction in School j from the predicted lower-level interaction in School j (from the school values of verbal IQ score and socioeconomic status)
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the verbal IQ score and socioeconomic status
- τ_{00} = The language score residual variance across schools (L2 residual variance) controlling for the verbal IQ score and socioeconomic status
- τ_{11} = The residual variance of the simple within-group slope of verbal IQ score across schools
- τ_{22} = The residual variance of the simple within-group slope of socioeconomic status across schools
- τ_{33} = The residual variance of the lower-level interaction effect between verbal IQ score and socioeconomic status across schools
- τ_{st} (where $s \neq t$) = The covariance between u_{sj} and u_{tj}
- $\rho_{st} = \tau_{st}/\sqrt{\tau_{ss}\tau_{tt}}$ (where $s, t = 0$, 1, 2, or 3 and $s \neq t$) = The covariance mentioned above in the correlation scale (from -1 to 1)

The model with lower-level interaction can be run by the `lmer` function:

```
m12b <- lmer(langPOST ~ 1 + IQ_verb.groupMC + ses.groupMC + IQ_verb.groupMC*ses.groupMC
+ IQ_verb.groupMeanC + ses.groupMeanC
+ IQ_verb.groupMC*IQ_verb.groupMeanC + IQ_verb.groupMC*ses.groupMeanC
+ ses.groupMC*IQ_verb.groupMeanC + ses.groupMC*ses.groupMeanC
+ IQ_verb.groupMC*ses.groupMC*IQ_verb.groupMeanC*
+ IQ_verb.groupMC*ses.groupMC*ses.groupMeanC
+ (1 + IQ_verb.groupMC + ses.groupMC + IQ_verb.groupMC*ses.groupMC|schoolnr),
data = dat, REML=FALSE)

summary(m12b)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + IQ_verb.groupMC + ses.groupMC + IQ_verb.groupMC * ses.groupMC
+ IQ_verb.groupMeanC + ses.groupMeanC
+ IQ_verb.groupMeanC * IQ_verb.groupMC + ses.groupMeanC * IQ_verb.groupMC
+ IQ_verb.groupMeanC * ses.groupMC + ses.groupMeanC * ses.groupMC
+ IQ_verb.groupMeanC * IQ_verb.groupMC * ses.groupMC
+ ses.groupMeanC * IQ_verb.groupMC * ses.groupMC
+ (1 + IQ_verb.groupMC + ses.groupMC + IQ_verb.groupMC * ses.groupMC | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 24668 24812 -12311    24622   24694
Random effects:
 Groups   Name                        Variance   Std.Dev. Corr
 schoolnr (Intercept)                 9.2187e+00 3.036230
          IQ_verb.groupMC             2.5328e-01 0.503274 -0.538
          ses.groupMC                 4.5736e-04 0.021386 -0.060 -0.809
          IQ_verb.groupMC:ses.groupMC 7.4477e-04 0.027291 -0.482 -0.479  0.903
 Residual                             3.6827e+01 6.068512
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
                                       Estimate Std. Error t value
(Intercept)                          41.3228831  0.2374470  174.03
IQ_verb.groupMC                       2.2359184  0.0673550   33.20
ses.groupMC                           0.1714991  0.0120377   14.25
IQ_verb.groupMeanC                    3.5087717  0.3094178   11.34
ses.groupMeanC                        0.0776109  0.0449142    1.73
IQ_verb.groupMC:ses.groupMC          -0.0199923  0.0067104   -2.98
IQ_verb.groupMC:IQ_verb.groupMeanC   -0.0390839  0.0916282   -0.43
IQ_verb.groupMC:ses.groupMeanC       -0.0167653  0.0126505   -1.33
```

```
ses.groupMC:IQ_verb.groupMeanC                     0.0038817  0.0177408    0.22
ses.groupMC:ses.groupMeanC                         0.0002484  0.0024205    0.10
IQ_verb.groupMC:ses.groupMC:IQ_verb.groupMeanC -0.0247756  0.0099071   -2.50
IQ_verb.groupMC:ses.groupMC:ses.groupMeanC       0.0013683  0.0013992    0.98
```

(*The output of the correlation between fixed effect is not shown here) The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + IQ_verb.groupMC + ses.groupMC +
           + IQ_verb.groupMC*ses.groupMC
           + IQ_verb.groupMeanC + ses.groupMeanC
           + IQ_verb.groupMC*IQ_verb.groupMeanC
           + IQ_verb.groupMC*ses.groupMeanC
           + ses.groupMC*IQ_verb.groupMeanC
           + ses.groupMC*ses.groupMeanC
           + IQ_verb.groupMC*ses.groupMC*IQ_verb.groupMeanC
           + IQ_verb.groupMC*ses.groupMC*ses.groupMeanC
           + (1 + IQ_verb.groupMC + ses.groupMC +
              IQ_verb.groupMC*ses.groupMC|schoolnr)
```

$$
\begin{aligned}
Y_{ij} = {} & \gamma_{00}(1) + \gamma_{10}(X_{1ij} - \bar{X}_{1.j}) + \gamma_{20}(X_{2ij} - \bar{X}_{2.j}) \\
& + \gamma_{30}(X_{1ij} - \bar{X}_{1.j}) \cdot (X_{2ij} - \bar{X}_{2.j}) \\
& + \gamma_{01}(\bar{X}_{1.j} - \bar{X}_{1..}) + \gamma_{02}(\bar{X}_{2.j} - \bar{X}_{2..}) \\
& + \gamma_{11}(X_{1ij} - \bar{X}_{1.j}) \cdot (\bar{X}_{1.j} - \bar{X}_{1..}) && \text{Fixed Effect} \\
& + \gamma_{12}(X_{1ij} - \bar{X}_{1.j}) \cdot (\bar{X}_{2.j} - \bar{X}_{2..}) \\
& + \gamma_{21}(X_{2ij} - \bar{X}_{2.j}) \cdot (\bar{X}_{1.j} - \bar{X}_{1..}) \\
& + \gamma_{22}(X_{2ij} - \bar{X}_{2.j}) \cdot (\bar{X}_{2.j} - \bar{X}_{2..}) \\
& + \gamma_{31}(X_{1ij} - \bar{X}_{1.j}) \cdot (X_{2ij} - \bar{X}_{2.j}) \cdot (\bar{X}_{1.j} - \bar{X}_{1..}) \\
& + \gamma_{32}(X_{1ij} - \bar{X}_{1.j}) \cdot (X_{2ij} - \bar{X}_{2.j}) \cdot (\bar{X}_{2.j} - \bar{X}_{2..}) \\
& + u_{0j}(1) + u_{1j}(X_{1ij} - \bar{X}_{1.j}) + u_{2j}(X_{2ij} - \bar{X}_{2.j}) && \text{Random Effect} \\
& + u_{3j}(X_{1ij} - \bar{X}_{1.j}) \cdot (X_{2ij} - \bar{X}_{2.j}) + e_{ij}
\end{aligned}
$$

The results are very complex. The fixed effect involves with three-way interaction such that the lower-level interaction was moderated by the school means of verbal IQ or socioeconomic status. The average lower-level interaction, γ_{30}, was significant. Furthermore, the lower-level interaction was significantly moderated by the school mean of verbal IQ, γ_{31}. Users should be very careful in interpreting the three-way interaction.

Model 13: Upper-level Interaction

In this model, the language scores (`langPOST`) is predicted by school average of students' socioeconomic status (`sch_ses`) and type of schools (`denomina`). Both predictors are measured at the upper level and expected to have interactive effect on the language score. The model with upper-level interaction would be

L1 $$Y_{ij} = \beta_{0j} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} + \gamma_{05}W_{5j} + \gamma_{06}(W_{1j} \cdot W_{2j}) + \gamma_{07}(W_{1j} \cdot W_{3j}) + \gamma_{08}(W_{1j} \cdot W_{4j}) + \gamma_{09}(W_{1j} \cdot W_{5j}) + u_{0j} \qquad u_{0j} \sim N(0, \tau_{00})$$

These notations should represent

- Y_{ij} = The language score of Student *i* in School *j*
- W_{1j} = The average socioeconomic status across students in School *j*
- W_{2j} = A dummy variable whether School *j* is classified as Type 2
- W_{3j} = A dummy variable whether School *j* is classified as Type 3
- W_{4j} = A dummy variable whether School *j* is classified as Type 4
- W_{5j} = A dummy variable whether School *j* is classified as Type 5
- β_{0j} = The average of language score across students in School *j*
- γ_{00} = The expected language score when the type of schools is 1 and the school's socioeconomic status is 0.
- γ_{10} = The expected increase in language score if the school's socioeconomic status increases by 1 in the type-1 school.
- γ_{20} = The expected difference in language score between the type-2 school and type-1 school when the school's socioeconomic status is 0.
- γ_{30} = The expected difference in language score between the type-3 school and type-1 school when the school's socioeconomic status is 0.
- γ_{40} = The expected difference in language score between the type-4 school and type-1 school when the school's socioeconomic status is 0.
- γ_{50} = The expected difference in language score between the type-5 school and type-1 school when the school's socioeconomic status is 0.
- γ_{60} = The change in the effect of the school's socioeconomic status when type of schools change from Type 1 to Type 2. Also, the change in the difference between the type-2 school and type-1 school when the school's socioeconomic status increases by 1.
- γ_{70} = The change in the effect of the school's socioeconomic status when type of schools change from Type 1 to Type 3. Also, the change in the difference between the type-3 school and type-1 school when the school's socioeconomic status increases by 1.
- γ_{80} = The change in the effect of the school's socioeconomic status when type of schools change from Type 1 to Type 4. Also, the change in the difference between the type-4 school and type-1 school when the school's socioeconomic status increases by 1.
- γ_{90} = The change in the effect of the school's socioeconomic status when type of schools change from Type 1 to Type 5. Also, the change in the difference between the type-5 school and type-1 school when the school's socioeconomic status increases by 1.
- e_{ij} = The deviation between the actual language score of Student *i* of School *j* from the School *j* average of language score.
- u_{0j} = The deviation of the mean of actual language score in School *j* from the predicted language score of School *j* (using school's socioeconomic status and type of school in prediction).
- σ^2 = The language score variance within schools (L1 variance)
- τ_{00} = The language score residual variance across schools (L2 residual variance) controlling for the type of schools and school's socioeconomic status

The model with upper-level interaction can be run by the `lmer` function:

```
m13 <- lmer(langPOST ~ 1 + sch_ses + denomina + sch_ses*denomina + (1|schoolnr), data = dat,
REML=FALSE)
```

```
summary(m13)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + sch_ses + denomina + +sch_ses * denomina + (1 |      schoolnr)
   Data: dat
   AIC    BIC logLik deviance REMLdev
 26537 26611 -13256    26513   26520
Random effects:
 Groups   Name        Variance Std.Dev.
 schoolnr (Intercept) 11.037   3.3222
 Residual             62.822   7.9261
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
                   Estimate Std. Error t value
(Intercept)       39.370240   0.500581   78.65
sch_ses            0.405549   0.070779    5.73
denomina2          3.561452   0.684965    5.20
denomina3          1.130100   0.728239    1.55
denomina4          3.824208   1.692785    2.26
denomina5          2.076073   1.214670    1.71
sch_ses:denomina2 -0.106074   0.109058   -0.97
sch_ses:denomina3 -0.002988   0.114329   -0.03
sch_ses:denomina4 -0.306565   0.213239   -1.44
sch_ses:denomina5 -0.055471   0.205669   -0.27

Correlation of Fixed Effects:
            (Intr) sch_ss denmn2 denmn3 denmn4 denmn5 sch_:2 sch_:3 sch_:4
sch_ses      0.019
denomina2   -0.731 -0.014
denomina3   -0.687 -0.013  0.502
denomina4   -0.296 -0.006  0.216  0.203
denomina5   -0.412 -0.008  0.301  0.283  0.122
sch_ss:dnm2 -0.012 -0.649  0.136  0.008  0.004  0.005
sch_ss:dnm3 -0.012 -0.619  0.009  0.045  0.003  0.005  0.402
sch_ss:dnm4 -0.006 -0.332  0.005  0.004 -0.583  0.003  0.215  0.205
sch_ss:dnm5 -0.006 -0.344  0.005  0.004  0.002 -0.206  0.223  0.213  0.114
```

The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + sch_ses + denomina
             + sch_ses*denomina
             + (1 + IQ_verb|schoolnr)
langPOST ~ 1 + sch_ses + d2 + d3 + d4 + d5
             + sch_ses*d2 + sch_ses*d3 + sch_ses*d4 + sch_ses*d5
             + (1|schoolnr)
```

$$Y_{ij} = \gamma_{00}(1) + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} + \gamma_{05}W_{5j}$$
$$+\gamma_{06}W_{1j}W_{2j} + \gamma_{07}W_{1j}W_{3j} + \gamma_{08}W_{1j}W_{4j} + \gamma_{09}W_{1j}W_{5j}$$
Fixed Effect

$$+u_{0j}(1) + e_{ij}$$
Random Effect

Similar to previous models, *p*-value and residual ICC can be computed. Because the interaction effects involve with multiple fixed effects, a deviance test between the models including and not including the interaction effects would be helpful for testing interaction. Thus, I made the baseline model that did not include the interaction effect and used the `anova` function to implement the deviance test.

```
m13a <- lmer(langPOST ~ 1 + sch_ses + denomina + (1|schoolnr), data = dat, REML=FALSE)

anova(m13, m13a)
```

```
Data: dat
Models:
m13a: langPOST ~ 1 + sch_ses + denomina + (1 | schoolnr)
m13: langPOST ~ 1 + sch_ses + denomina + sch_ses * denomina + (1 |
m13:     schoolnr)
     Df   AIC   BIC logLik  Chisq Chi Df Pr(>Chisq)
m13a  8 26532 26581 -13258
m13  12 26537 26611 -13256 2.8375      4     0.5854
```

In this case, the interaction effect between school's average in socioeconomic status and type of school on the language scores was not significant, $\chi^2(4) = 2.84$, $p = .59$.

Model 14: Cross-level Interaction

In this model, the language scores (`langPOST`) is predicted by sex (`sex`) and type of schools (`denomina`). Sex is measured at the lower level but type of schools is measured at the upper level. Both predictors are expected to have interactive effect on the language score. Note that this model is very similar to Model 6. The only difference is that sex is a categorical variable. The model with cross-level interaction would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}X_{ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + \gamma_{11}W_{1j} + \gamma_{12}W_{2j} + \gamma_{13}W_{3j} + \gamma_{14}W_{4j} + u_{1j}$$
$$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ \tau_{10} & \tau_{11} \end{bmatrix}\right)$$

These notations should represent

- Y_{ij} = The language score of Student *i* in School *j*
- X_{ij} = A dummy variable whether Student *i* in School *j* is classified as female
- W_{1j} = A dummy variable whether School *j* is classified as Type 2
- W_{2j} = A dummy variable whether School *j* is classified as Type 3
- W_{3j} = A dummy variable whether School *j* is classified as Type 4
- W_{4j} = A dummy variable whether School *j* is classified as Type 5
- β_{0j} = The average of language score within School *j* across males (when sex is 0)
- β_{1j} = The expected change in language score when the sex changes from 0 to 1, which is the sex difference in language score in School *j*.
- γ_{00} = The expected male language score across all type-1 schools
- γ_{01} = The difference in male language score between all schools in Type 2 and all schools in Type 1
- γ_{02} = The difference in male language score between all schools in Type 3 and all schools in Type 1
- γ_{03} = The difference in male language score between all schools in Type 4 and all schools in Type 1
- γ_{04} = The difference in male language score between all schools in Type 5 and all schools in Type 1
- γ_{10} = The expected sex difference in language score across type-1 schools.
- γ_{11} = The difference between type-2 and type-1 schools in the sex difference in language score
- γ_{12} = The difference between type-3 and type-1 schools in the sex difference in language score
- γ_{13} = The difference between type-4 and type-1 schools in the sex difference in language score
- γ_{14} = The difference between type-5 and type-1 schools in the sex difference in language score
- e_{ij} = The difference between the actual language score of Student *i* in School *j* and the sex-specific average language score in School *j*
- u_{0j} = The deviation of the average male language score of School *j* from the mean of male averages across schools in the same Type that School *j* is in

- u_{1j} = The deviation of the sex difference in language score of School *j* from the expected difference across schools in the same type that School *j* is in
- σ^2 = The language score residual variance within schools (L1 residual variance) controlling for the sex
- τ_{00} = The language score residual variance across schools (L2 residual variance) controlling for the sex and the type of schools
- τ_{11} = The residual variance of the sex difference in language score across schools controlling for the type of schools
- τ_{10} = The covariance between the residual of the random intercept and the residual of the random slope
- $\rho_{10} = \tau_{10}/\sqrt{\tau_{00}\tau_{11}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)

To run the analysis in R, initially, the sex variable needs to be transformed into a factor.

```
dat$sex <- factor(dat$sex)
```

Next, the model with cross-level interaction can be run by the `lmer` function:

```
m14 <- lmer(langPOST ~ 1 + sex + denomina + sex*denomina + (1 + sex|schoolnr), data = dat,
REML=FALSE)

summary(m14)
```

```
Linear mixed model fit by maximum likelihood
Formula: langPOST ~ 1 + sex + denomina + sex * denomina + (1 + sex | schoolnr)
   Data: dat
   AIC   BIC logLik deviance REMLdev
 26516 26603 -13244    26488   26475
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 schoolnr (Intercept) 17.9743  4.2396
          sex1         3.2988  1.8163   -0.435
 Residual             60.7440  7.7938
Number of obs: 3758, groups: schoolnr, 211

Fixed effects:
               Estimate Std. Error t value
(Intercept)     38.1934     0.6516   58.61
sex1             2.2610     0.5579    4.05
denomina2        3.4008     0.8793    3.87
denomina3        0.4579     0.9431    0.49
denomina4        4.6992     1.7087    2.75
denomina5        3.1020     1.5481    2.00
sex1:denomina2  -0.4827     0.7407   -0.65
sex1:denomina3   1.2290     0.8078    1.52
sex1:denomina4  -0.4698     1.4422   -0.33
sex1:denomina5  -1.0974     1.2918   -0.85

Correlation of Fixed Effects:
            (Intr) sex1   denmn2 denmn3 denmn4 denmn5 sx1:d2 sx1:d3 sx1:d4
sex1        -0.494
denomina2   -0.741  0.366
denomina3   -0.691  0.342  0.512
denomina4   -0.381  0.189  0.283  0.263
denomina5   -0.421  0.208  0.312  0.291  0.161
sex1:denmn2  0.372 -0.753 -0.493 -0.257 -0.142 -0.157
sex1:denmn3  0.341 -0.691 -0.253 -0.488 -0.130 -0.144  0.520
sex1:denmn4  0.191 -0.387 -0.142 -0.132 -0.472 -0.081  0.291  0.267
sex1:denmn5  0.214 -0.432 -0.158 -0.148 -0.081 -0.501  0.325  0.298  0.167
```

The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + sex + denomina
           + sex*denomina
```

```
              + (1 + sex|schoolnr)
langPOST ~ 1 + sex1 + d2 + d3 + d4 + d5
              + sex1*d2 + sex1*d3 + sex1*d4 + sex1*d5
              + (1 + sex1|schoolnr)
```

$$Y_{ij} = \gamma_{00}(1) + \gamma_{10}X_{ij} + \gamma_{01}W_{1j} + \gamma_{02}W_{2j} + \gamma_{03}W_{3j} + \gamma_{04}W_{4j} + \gamma_{11}X_{ij}W_{1j} + \gamma_{12}X_{ij}W_{2j} + \gamma_{13}X_{ij}W_{3j} + \gamma_{14}X_{ij}W_{4j}$$ Fixed Effect

$$+u_{0j}(1) + u_{1j}X_{ij} + e_{ij}$$ Random Effect

Because the interaction effect involved multiple fixed effects, I made the baseline model that does not include the interaction effect and used the `anova` function to implement the deviance test.

```
m14a <- lmer(langPOST ~ 1 + sex + denomina + (1 + sex|schoolnr), data = dat, REML=FALSE)

anova(m14, m14a)
```

```
Data: dat
Models:
m14a: langPOST ~ 1 + sex + denomina + (1 + sex | schoolnr)
m14: langPOST ~ 1 + sex + sex * denomina + (1 + sex | schoolnr)
     Df   AIC   BIC logLik  Chisq Chi Df Pr(>Chisq)
m14a 10 26514 26576 -13247
m14  14 26516 26603 -13244 6.2886      4     0.1786
```

In this case, the interaction effect between student's sex and type of school on the language scores was not significant, $\chi^2(4) = 6.29$, $p = .18$. In this cross-level interaction, users may try group-mean centering for the sex variable and investigate the differences between two models.

Probing Two-Way Interaction

After any types of interactions are significant, researchers would like to know the meanings behind the interactions. In this section, I will show you how to create a helpful plot for probing interactions and how to calculate the simple slopes and implement the null hypothesis testing of simple slopes. Also, I will show you how to use centering as another method for probing interaction. Note that, in multilevel modeling, the methods for probing interaction, testing simple slopes, or centering for probing interactions are similar regardless of the (lower or upper) levels of the predictors. Please refer to Bauer and Curran (2005) for more details in probing interactions in multilevel modeling.

Two Continuous Predictors

There are at least three ways in probing interactions between continuous predictors: 1) use the online utility from Kris Preacher's webpage (http://www.quantpsy.org/interact/index.html; Preacher, Curran, & Bauer, 2006), 2) use the `rockchalkMultilevel` package, and 3) centering. Model 12 with the interaction between two continuous predictors is used here. The verbal IQ score is treated as the target variable and the socioeconomic status is treated as a moderator.

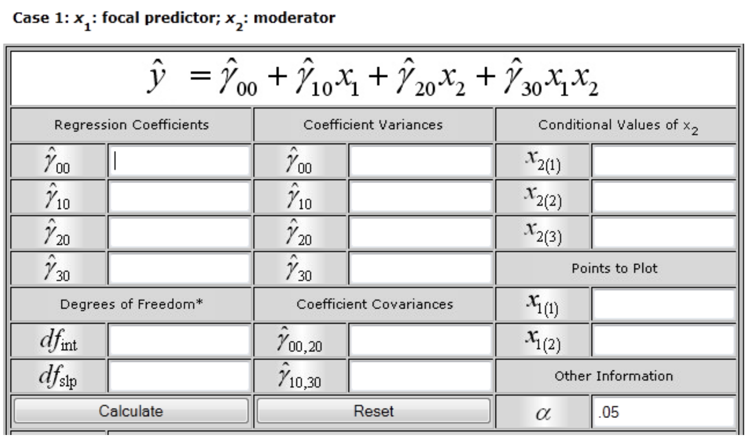
Kris Preacher's Online Utility

If you go to the website, there are multiple options for probing interaction based on difference types of models. Because the multilevel model is used with two-way interaction, choose the link of `Simple slopes and the region of significance for HLM 2-way interactions`. Next, go down the page. You will see three java applets representing three interaction cases based on the levels of predictors:

1. X_1: focal predictor, X_2: moderator. This case is a lower-level interaction.
2. W_1: focal predictor, W_2: moderator. This case is an upper-level interaction.

3. X_1: focal predictor, W_1: moderator. This case is a cross-level interaction.

Because Model 12 involves with lower-level interaction, the first case is selected. The Java applet should look like the following picture.



Users need to put numbers for those boxes. We will go over how to find the appropriate numbers for these boxes.

1. Regression coefficients are the fixed-effect estimates. If you run `summary(m11)`, you will see the fixed-effect output:

```
Fixed effects:
            Estimate Std. Error t value
(Intercept) 41.282941   0.249130  165.71
IQ_verb      2.251518   0.064446   34.94
ses          0.173129   0.011303   15.32
IQ_verb:ses -0.021280   0.005119   -4.16
```

From the applet, $\hat{\gamma}_{00}$ is the intercept estimate, 41.282941. $\hat{\gamma}_{10}$ is the regression coefficient of your target variable, `IQ_verb`, which is 2.251518. $\hat{\gamma}_{20}$ is the regression coefficient from the moderator, `ses`, which is 0.173129. $\hat{\gamma}_{30}$ is the regression coefficient from the interaction, `IQ_verb:ses`, which is -0.021280.

2. Degrees of freedom. Because we use Wald statistic (*z* approximation), we will leave these boxes blank.
3. Coefficient variances. This is simply the squared standard error of each fixed effect. Users may calculate the variances from the standard errors by hand. However, it is easier to directly request the asymptotic covariance matrix between fixed effects from the output by the `vcov` function. Because the result of the `vcov` function involves scientific notation, I will use the `round` function (using 10 digits) to make the outputs as numbers:

```
round(vcov(m11), 10)
```

```
4 x 4 Matrix of class "dpoMatrix"
              [,1]          [,2]          [,3]          [,4]
[1,]  0.0620658219 -0.0050169005  0.0001968993 -0.0004794081
[2,] -0.0050169005  0.0041532295 -0.0002458554  0.0000148274
[3,]  0.0001968993 -0.0002458554  0.0001277613 -0.0000095634
[4,] -0.0004794081  0.0000148274 -0.0000095634  0.0000262045
```

The orders of the rows and the columns match with the order of the fixed effects listed above. That is, the rows or columns represent $\hat{\gamma}_{00}$, $\hat{\gamma}_{10}$, $\hat{\gamma}_{20}$, and $\hat{\gamma}_{30}$, respectively. In the coefficient variances boxes, the diagonal elements of the matrix will be used. $\hat{\gamma}_{00}$ is the variance of the intercept estimate, 0.0620658219. $\hat{\gamma}_{10}$ is the variance of the regression coefficient of your target variable, 0.0041532295. $\hat{\gamma}_{20}$ is the variance of the regression coefficient from the moderator, 0.0001277613. $\hat{\gamma}_{30}$ is the variance of the regression coefficient from the interaction, 0.0000262045.

4. Coefficient covariance. The covariances represent the off-diagonal elements of the asymptotic covariance matrix. $\hat{\gamma}_{00,20}$ represents the covariance between $\hat{\gamma}_{00}$ and $\hat{\gamma}_{20}$, which is the element `[1, 3]` of the matrix. The number is 0.0001968993. $\hat{\gamma}_{10,30}$ represents the covariance between $\hat{\gamma}_{10}$ and $\hat{\gamma}_{30}$, which is the element `[2, 4]` of the matrix. The number is 0.0000148274.
5. Conditional values of X_2. The value of socioeconomic status that users wish to probe. Theoretically, the effect of the target variable should be investigated at meaningful levels of a moderator. If users do not have theoretical numbers, users may pick M, $M + SD$, and $M - SD$ or 25th, 50th, and 75th percentile ranks. I will pick the values based on percentile ranks by the `quantile` function:

```
quantile(dat$ses)
```

```
    0%    25%    50%    75%   100%
-17.73  -7.73  -1.73   9.27  22.27
```

Then, I will put -7.73, -1.73, and 9.27 for $X_{2(1)}$, $X_{2(2)}$, and $X_{2(3)}$, respectively.

6. Points to plot. The range of the verbal IQ scores that users wish to see in the plot. I usually use the minimum and maximum values, which can be calculated by the `range` function:[6]

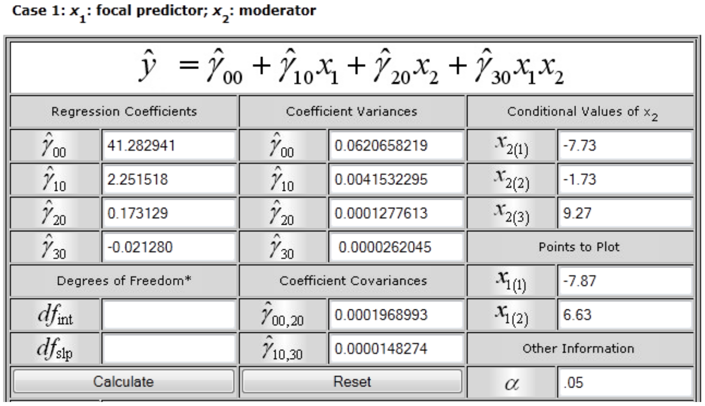
```
range(dat$IQ_verb)
```

```
[1] -7.87  6.63
```

Therefore, I put -7.87 and 6.63 for $X_{1(1)}$ and $X_{1(2)}$, respectively.

The filled boxes should look similar to the following picture:

Case 1: x_1: focal predictor; x_2: moderator



Then, click on `Calculate`. You will see the outputs listed in three different big boxes below. The first box indicates the simple slopes and their significance testing. The second box will show the R code for

[6] Note that, in the applet for cross-level interaction (Case 3), three values are needed for the focal variable. Users may simply put the minimum and maximum values for Points 1 and 3. For Point 2, users may put any arbitrary values, such as the average of the focal variable.

plotting the simple-slope graph. The third box will show the R code for plotting confidence intervals of the simple slopes.

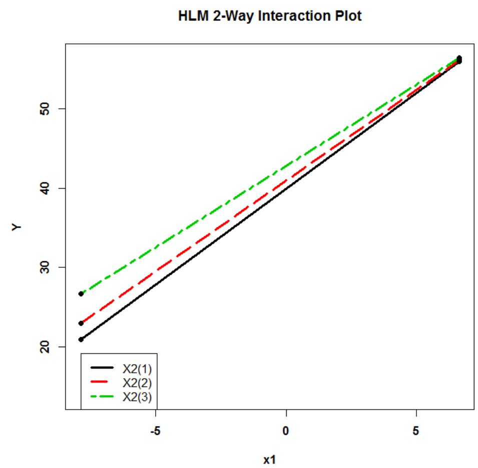
In the first box, we will focus only two parts: `region of significance` and `simple intercepts and slopes at conditional values`.

```
Region of Significance
=======================================================
  x2 at lower bound of region = 71.3697
  x2 at upper bound of region = 201.0655
  (simple slopes are significant *outside* this region.)

Simple Intercepts and Slopes at Conditional Values
=======================================================
  At x2(1)...
    simple intercept = 39.9447(0.2582), z=154.7175, p=0
    simple slope     = 2.416(0.0741), z=32.6078, p=0
  At x2(2)...
    simple intercept = 40.9834(0.2485), z=164.9037, p=0
    simple slope     = 2.2883(0.0647), z=35.3926, p=0
  At x2(3)...
    simple intercept = 42.8878(0.2769), z=154.8639, p=0
    simple slope     = 2.0543(0.0817), z=25.1343, p=0
```

The region of significance shows the range of moderator values that provides significance results. From the output, the simple slopes were not significant if the moderator values were in between 71.37 and 201.07. The simple slopes are significant if the moderator values were below 71.37 or above 201.07. Because all observed values of socioeconomic status were below 71.37, all simple slopes were all significant.

The simple intercept is the expected value of dependent variable when the target variable equals 0 and the moderator is equal to the specified values. In this case, the expected values of language scores when verbal IQ score was equal to 0 and socioeconomic status was equal to the 25th, 50th, and 75th percentile ranks were 39.94, 40.98, and 42.89, respectively. All simple intercepts were significant. The simple slope is the expected change in dependent variable when the target variable increases by 1 at the specified values of moderator. In this case, the expected change in language score when verbal IQ score increased by 1 at the 25th, 50th, and 75th percentile ranks were 2.42, 2.29, and 2.05, respectively. That is, when the socioeconomic status increased, the effect of verbal IQ score on language score was lower. All simple slopes were significant.

The second box provides the R code for plotting simple slopes. You may hit `submit above to Rweb` or copy the R code and paste in your R program. The graph should be



From the graph, the slope of X_1 (verbal IQ) was shallower when X_2 (socioeconomic status) increased.

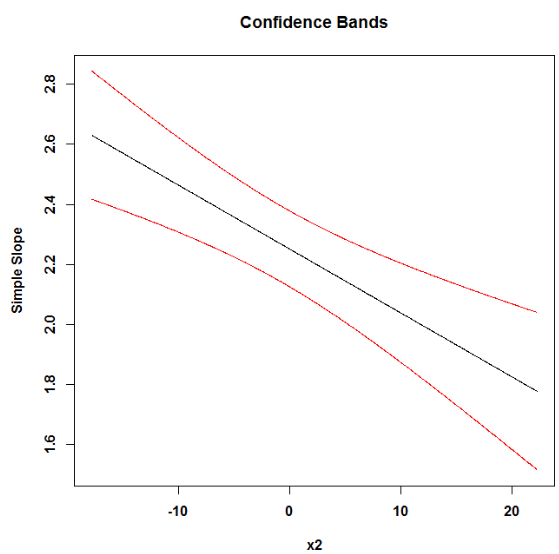
The third box provides the R code for the confidence intervals of the simple slopes. Before using this code, please make sure to take a look at the first two lines. As a default, you see the following lines:

```
z1=-10  #supply lower bound for x2 here
z2=10   #supply upper bound for x2 here
```

That is, the graph assumes that the minimum and maximum values of the moderator are -10 and 10. These are not the correct values for the current data. The minimum and maximum values of the socioeconomic status can be calculated by the `range` function, which are -17.73 and 22.27. Then, change the two lines above to reflect the range of the current moderator:

```
z1=-17.73  #supply lower bound for x2 here
z2=22.27   #supply upper bound for x2 here
```

Next, you may hit `submit above to Rweb` or copy the R code and paste in your R program. The graph should be



This graph shows the confidence interval of the simple slope (the effect of verbal IQ on language score) at different levels of X_2 (socioeconomic status) on the X axis.

Using the **`rockchalkMultilevel`** *package*

If users use the `lmer` function to get the outputs containing the interaction term (which is what we are doing in this paper), the functions in the `rockchalkMultilevel` package (Pornprasertmanit, 2013) can be used to probe interactions. Before installing the package, users need to make sure that they have two dependent packages in their computer: `rockchalk` (Johnson, 2012) and `phia` (De Rosario-Martinez, 2012):

```
install.packages("rockchalk")

install.packages("phia")
```

After these packages are installed, the `rockchalkMultilevel` package can be installed from the KU repository:[7]

```
install.packages("rockchalkMultilevel", repos="http://rweb.quant.ku.edu/kran", type="source")
```

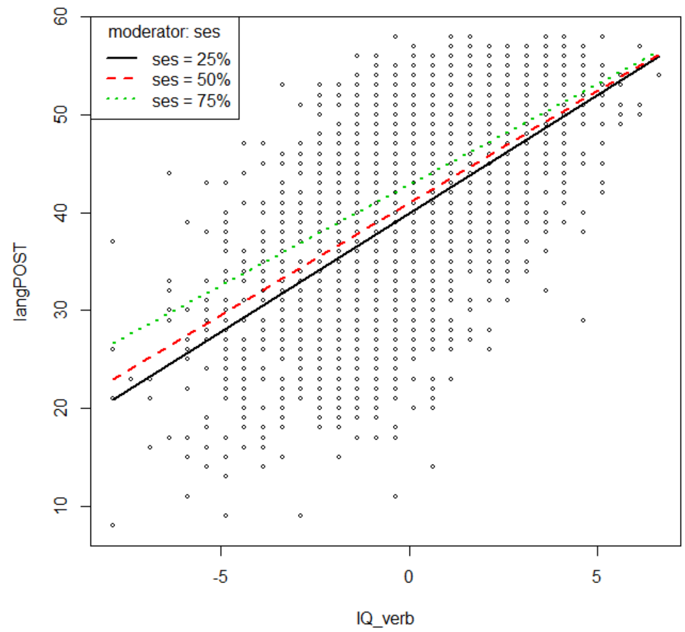
As usual, the package needs to be called in the R workspace by using the `library` function:

```
library(rockchalkMultilevel)
```

For the interaction between two continuous predictors, the `plotSlopes.mlm` can be used to visualize the simple slopes:

```
plotSlopes.mlm(m12, "IQ_verb", "ses")
```

[7] The `rockchalkMultilevel` package is a temporary package that I compiled it for this paper. The functions inside the packages were modified from the `rockchalk` and `phia` packages to work with the output from multilevel analysis (specifically the output from the `lmer` function). If those packages include the multilevel feature, I will delete the package. I have conducted a brief test on those functions and found that the output matched with the results from the Kris Preacher's website, the centering approach and the multivariate Wald test. For the cases similar to the examples I illustrate in this paper, the output can be trusted. Because I simply modify the functions, full credits of these functions should be given to the authors of the `rockchalk` and `phia` packages.



The first argument is the output from the `lmer` function. The second argument is the target variable. The third argument is the moderator. This graph is similar to the graphs provided from the online applet. The default values of moderators are its 25th, 50th, and 75th percentile ranks. If users wish to have different values, the `modxVals` argument can be used:

```
plotSlopes.mlm(m12, "IQ_verb", "ses", modxVals = c(-10, -5, 0, 5, 10))
```

Next, the simple slopes of each value of the moderator can be investigated and tested for significance by the `testSlopes.mlm` function. First, the output from the `plotSlopes.mlm` is saved as an object:

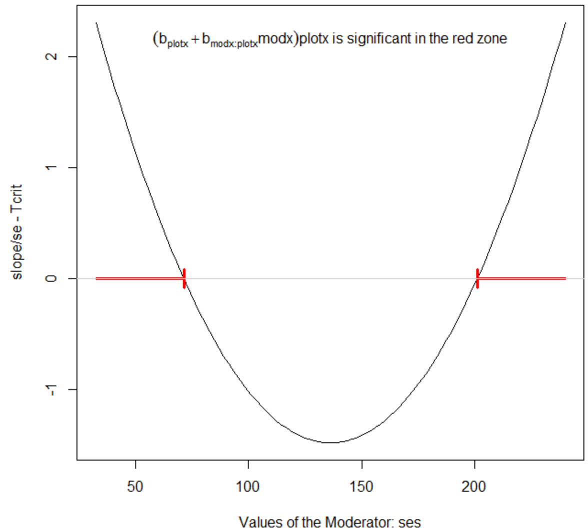
```
simpleSlope12 <- plotSlopes.mlm(m12, "IQ_verb", "ses")
```

Then, the `testSlopes.mlm` function is implemented on the saved object:

```
testSlopes.mlm(simpleSlope12)
```

```
These are the straight-line "simple slopes" of the variable IQ_verb
 for the selected moderator values.
    "ses"    slope Std. Error  z value      Pr(>|z|)
25% -7.73 2.416010 0.07409312 32.60775 3.184950e-233
50% -1.73 2.288332 0.06465566 35.39260 2.219387e-274
75%  9.27 2.054255 0.08173100 25.13434 2.096100e-139
Values of modx OUTSIDE this interval:
       lo        hi
 71.37318 201.05039
cause the slope of (b1 + b2modx)plotx to be statistically significant
```

This function does not list the simple intercept. The simple slope here matched with the result from the online utility. That is, the expected change in language score when verbal IQ score increased by 1 at the 25th, 50th, and 75th percentile ranks of socioeconomic status were 2.42, 2.29, and 2.05, respectively. All simple slopes were significant. The second portion of the output is the region of moderator values providing significant simple slopes. If the moderator values were in between 71.37 to 201.05, the simple slopes were not significant. The output was accompanied by the graph:



The area of *X* axis in the red horizontal line provided significant slopes.

Centering

The regression coefficients of the main effects are the effect of one variable (target variable) given that another variable (moderator) is 0. Therefore, if we center a moderator variable at a specific value, the regression coefficient will represent the effect of target variable given a specific value of moderator. For example, researchers wish to know the simple slope of verbal IQ score given the socioeconomic status level of 9.27. The centering can be implemented on the SES variable (moderator) and the centered variable is used in the model.

```
dat$ses.c <- dat$ses - 9.27

m12c <- lmer(langPOST ~ 1 + IQ_verb + ses.c + IQ_verb*ses.c + (1 + IQ_verb + ses.c +
IQ_verb*ses.c|schoolnr), data = dat, REML=FALSE)

summary(m12c)
```

```
Fixed effects:
              Estimate Std. Error t value
(Intercept)   42.90128    0.27948  153.50
IQ_verb        2.05171    0.08419   24.37
ses.c          0.17445    0.01177   14.83
IQ_verb:ses.c -0.02136    0.00528   -4.05
```

The fixed effects are only shown here. You may notice that the slope of verbal IQ score is 2.052, which is significant. This value represents the effect of verbal IQ score given the socioeconomic status level of 9.27. Researchers can use this technique to investigate the effect of socioeconomic status given the level of verbal IQ score.

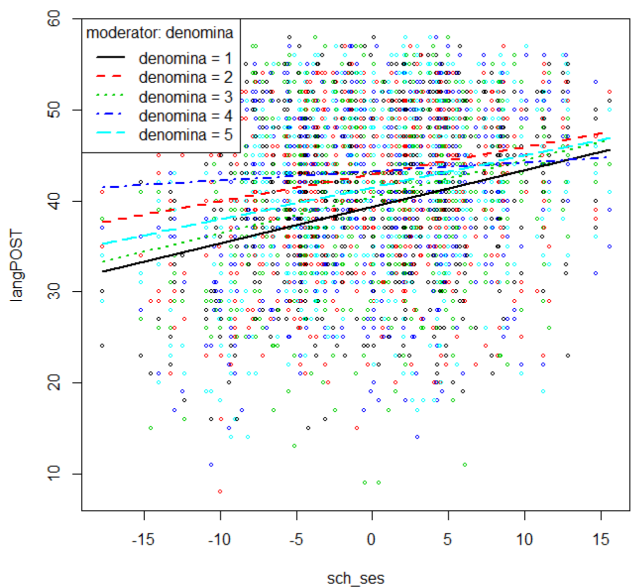
One Continuous Predictor and One Categorical Predictor

Model 13 is used here. The school's average in socioeconomic status is treated as the target variable and the type of schools is treated as a moderator. Note that the probing interaction is implemented here for illustration although the interaction effect was not significant. The three methods for probing the interaction between two continuous predictors are applicable for the interactions between one continuous predictor and one categorical variable. The online utility was limited to only dichotomous variable so I will illustrate the method in the `rockchalkMultilevel` package and centering only.

Using the **`rockchalkMultilevel`** *package*

Similarly, the `plotSlopes.mlm` can be used to visualize the simple slopes. In this function, the continuous variable must be always used as the target variable and the categorical variable must be always used as the moderator:

```
plotSlopes.mlm(m13, "sch_ses", "denomina")
```



Next, the simple slopes of each type of schools can be tested by the `testSlopes.mlm` function using the similar step mentioned previously.

```
simpleSlope13 <- plotSlopes.mlm(m12, "sch_ses", "denomina")

testSlopes.mlm(simpleSlope13)
```

```
These are the straight-line "simple slopes" of the variable sch_ses
 for the selected moderator values.
         "denomina"      slope Std. Error   z value     Pr(>|z|)
1           sch_ses 0.40554906 0.07077942 5.7297598 1.005729e-08
2 sch_ses:denomina2 0.29947471 0.08296936 3.6094615 3.068333e-04
3 sch_ses:denomina3 0.40256152 0.08978574 4.4835798 7.340114e-06
4 sch_ses:denomina4 0.09898413 0.20114918 0.4920931 6.226535e-01
5 sch_ses:denomina5 0.35007785 0.19310583 1.8128807 6.985022e-02
```

The simple slopes for schools of Type 1, 2, and 3 were significant whereas the simple slopes for schools of Type 4 and 5 were not significant. Users may notice that the simple slope in type-5 school was strong but the slope was not significant. The reason is that the number of schools classified as Type 5 was low. Several steps can be used to get the number of schools in each type. First, use the table function to create a crosstab between school type and school ID:

```
ctab <- table(dat$schoolnr, dat$denomina)
```

If users run the `ctab` object, users will see the table of school type and school ID where each cell represents the number of students. Next, we will check which cell is greater than 0.

```
typeschool <- ctab > 0
```

The `typeschool` object is a table indicating the type of each school. Finally, the `apply` function is used to sum the number of schools classified as `TRUE` in each column:

```
apply(typeschool, 2, sum)
```

```
 1  2  3  4  5
61 72 55 10 13
```

The `apply` function is used to apply the `sum` function at each column vector (the second argument is 2 which means columns) on the `typeschool` object. The numbers of schools in Types 1 and 2 were high leading to higher power in simple slopes significance testing. The numbers of schools in Type 4 and 5 were low, however, leading to lower power in simple slopes significance testing. Note that the moderator is a categorical variable so the region of moderator values giving significance simple slopes is not provided.

Centering

By centering approach, users may select either the continuous variable or the grouping variable as a moderator. When the grouping variable is a moderator, the regression coefficient of the main effect of the continuous variable represents the effect at the reference group. Therefore, the reference group of the grouping variable can be simply changed. Then, the regression coefficient of the target variable will represent the effect at the new reference group. To change the reference group, the `relevel` function can be used:

```
dat$denomina.c <- relevel(dat$denomina, "2")
```

The first argument is a factor variable. The second argument is the name of the reference group. In this case, School type 2 is used as the reference group. The model can be rerun by the centered variable:

```
m13c <- lmer(langPOST ~ 1 + sch_ses + denomina.c + sch_ses*denomina.c + (1|schoolnr), data = dat,
REML=FALSE)

summary(m13c)
```

```
Fixed effects:
                    Estimate Std. Error t value
(Intercept)         42.93169    0.46754   91.82
sch_ses              0.29947    0.08297    3.61
denomina.c1         -3.56145    0.68497   -5.20
denomina.c3         -2.43135    0.70594   -3.44
denomina.c4          0.26276    1.68331    0.16
denomina.c5         -1.48538    1.20143   -1.24
sch_ses:denomina.c1  0.10607    0.10906    0.97
sch_ses:denomina.c3  0.10309    0.12225    0.84
sch_ses:denomina.c4 -0.20049    0.21759   -0.92
sch_ses:denomina.c5  0.05060    0.21018    0.24
```

The fixed effects are only shown here. You may notice that the slope of verbal IQ score is 0.30, which is significant. This value represents the effect of verbal IQ score at the school type 2.

When a continuous variable is picked as a moderator, the regression coefficients of the dummy variables are interpreted as the effects when the continuous predictor is 0. The continuous can be centered at a specific value so that the regression coefficients of the dummy variables representing the effects at the given value. For example, the effects of type of schools when the socioeconomic status is 5 can be calculated:

```
dat$sch_ses.c5 <- dat$sch_ses - 5

m13d <- lmer(langPOST ~ 1 + sch_ses.c5 + denomina + sch_ses.c5*denomina + (1|schoolnr), data =
dat, REML=FALSE)

summary(m13d)
```

```
Fixed effects:
                       Estimate Std. Error t value
(Intercept)           41.397986   0.618462   66.94
sch_ses.c5             0.405549   0.070779    5.73
denomina2              3.031080   0.931781    3.25
denomina3              1.115163   0.945772    1.18
denomina4              2.291383   1.378197    1.66
denomina5              1.798717   1.421093    1.27
sch_ses.c5:denomina2 -0.106074   0.109058   -0.97
sch_ses.c5:denomina3 -0.002988   0.114329   -0.03
sch_ses.c5:denomina4 -0.306565   0.213239   -1.44
sch_ses.c5:denomina5 -0.055471   0.205669   -0.27
```

The fixed effects are only shown here. The main effects of the dummy variables are used here. For example, the difference between school type-2 and type-1 (`denomina2`) when socioeconomic status was 5 was 3.03, which was significant.[8]

Two Categorical Predictors

Model 14 is used here. The student's sex and type of schools are used to predict language scores. Note that the probing interaction is implemented here for illustration although the interaction effect was not significant. With the interaction between two categorical variables, the situation is similar to two-way factorial analysis of variance. If the interaction is significant, the simple main effect is used.[9] The simple main effect tests whether the effect of one predictor is significant given different levels of another predictor.

The online utility does not work in this case (except the interaction between two dummy variables). Users may use the `rockchalkMultilevel` package or the centering approach, which will be mentioned later. The `interactionMeans.mlm` can be used to see the expected (or adjusted) means of each condition:

```
interactionMeans.mlm(m14)
```

```
   sex denomina adjusted mean
1    0        1      38.19341
2    1        1      40.45440
3    0        2      41.59424
4    1        2      43.37249
5    0        3      38.65132
6    1        3      42.14131
7    0        4      42.89258
```

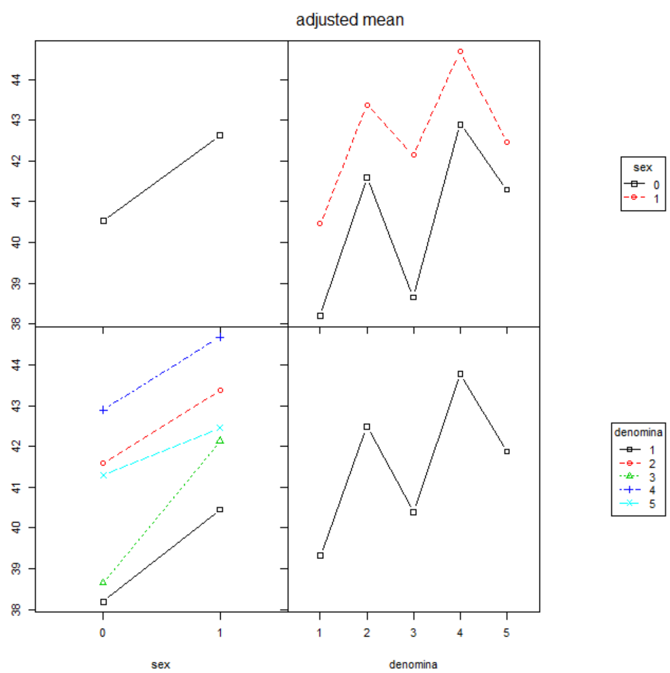
[8] If users wish to test whether all types of schools are significantly different at a given level of the school SES, further steps are needed after the centering. One way is to use the multivariate Wald test. Check the `wald.mlm` function in the `rockchalkMultilevel` package to see an example: `?wald.mlm`

[9] Simple main effects are conceptually similar to simple slopes. However, simple main effects are designed to have an output layout appropriate for categorical variables.

```
8    1         4      44.68382
9    0         5      41.29540
10   1         5      42.45902
```

The plot function can be directly applied to the expected means of each condition:

```
plot(interactionMeans.mlm(m14))
```



The on-diagonal graphs represent the expected means of each group in each factor. The off-diagonal graphs represent the interaction effects.

In testing simple main effect, one factor is selected as a target variable and another factor is selected as moderator. For example, the sex differences in each type of schools are investigated. The testInteractions.mlm function can be used for the simple main effect testing.

```
testInteractions.mlm(m14, fixed="denomina", across="sex", adjustment = "none")
```

```
Chisq Test:
P-value adjustment method: none
    Value Df   Chisq Pr(> Chisq)
1 -2.2610  1 16.4252   5.061e-05 ***
2 -1.7783  1 13.3168    0.000263 ***
3 -3.4900  1 35.6903   2.313e-09 ***
4 -1.7912  1  1.8142    0.178008
5 -1.1636  1  0.9974    0.317929
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The first argument is the output from the lmer function. The second argument, fixed, is the moderator variable. The third argument, across, is the target variable that users wish to test for the difference. The fourth argument, adjustment, is the method for correcting for familywise error rate. If the adjustment argument is specified as "none", no *p*-value correction is implemented. Users may specify "bonferroni" or "holm" (or other methods listed in the help page). I recommend Howell (2007) for the details of each type of familywise-error-rate corrections. In the output, the chi-square tests for the schools type 1, 2, or 3 were significant whereas the chi-square tests in the schools type 4 and 5 were not significant (because lower power from lower number of schools in these groups).[10]

The type of schools differences in each group of sex can be investigated as well:

```
testInteractions.mlm(m14, fixed="sex", across="denomina", adjustment = "bonferroni")
```

```
Chisq Test:
P-value adjustment method: bonferroni
  denomina1 denomina2 denomina3 denomina4 Df  Chisq Pr(> Chisq)
0   -3.1020   0.29884  -2.64408    1.5972  4 23.025   0.0002504 ***
1   -2.0046   0.91347  -0.31771    2.2248  4 15.360   0.0080216 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The Bonferroni adjustment was used. In the output, the `denomina1`, `denomina2`, `denomina3`, and `denomina4` represent the mean of that group (1, 2, 3, or 4) compared with the last group (type = 5). Note that the effects of school types within each sex were significant after the Bonferroni adjustment.

Because the `denomina` variable has more than two groups, users may wish to implement post-hoc pairwise comparison. Users can simply specify the `pairwise` argument in the `testInteractions.mlm` function:

```
testInteractions.mlm(m14, fixed="sex", pairwise="denomina", adjustment = "holm")
```

```
Chisq Test:
P-value adjustment method: holm
          Value Df   Chisq Pr(> Chisq)
1-2 : 0 -3.4008  1 14.9584    0.002198 **
1-3 : 0 -0.4579  1  0.2358    1.000000
1-4 : 0 -4.6992  1  7.5629    0.101290
1-5 : 0 -3.1020  1  4.0151    0.631317
2-3 : 0  2.9429  1 10.6488    0.019826 *
2-4 : 0 -1.2983  1  0.5928    1.000000
2-5 : 0  0.2988  1  0.0385    1.000000
3-4 : 0 -4.2413  1  6.0773    0.205399
3-5 : 0 -2.6441  1  2.8692    1.000000
4-5 : 0  1.5972  1  0.5711    1.000000
1-2 : 1 -2.9181  1 12.5384    0.007575 **
1-3 : 1 -1.6869  1  3.5638    0.767678
1-4 : 1 -4.2294  1  6.6929    0.154879
1-5 : 1 -2.0046  1  1.9477    1.000000
2-3 : 1  1.2312  1  2.0911    1.000000
2-4 : 1 -1.3113  1  0.6616    1.000000
2-5 : 1  0.9135  1  0.4194    1.000000
3-4 : 1 -2.5425  1  2.3780    1.000000
3-5 : 1 -0.3177  1  0.0479    1.000000
4-5 : 1  2.2248  1  1.2427    1.000000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The Holm method was used for controlling familywise error rate. In this code, the `across` argument was substituted by the `pairwise` argument. Notice that schools type 1 and type 2 were significantly different in the language scores regardless of sex.

Researchers may use centering approach to probe the simple main effect by the `relevel` function. However, the regression coefficients are not directly interpretable for any variables with more than two categories. I found that the simple main effect was much simpler than the centering so I do not discuss the centering approach here.

[10] The last group of the sex variable (`"1"`) is used as the reference group in this function. That is why the `Value` column has different signs from the summary of the result.

Growth Curve Model

In the following sections, we will use different data sets. Users may import the data by the following script:

```
long <- read.csv("C:/Users/student/Desktop/mathgrowth.csv", header = TRUE, na.strings="-999999")
```

Because the data have missing observations, we need to specify which values represent missing observations in the `na.strings` argument. In the data, -999999 is used to represent missing observations. After the data are imported, the missing observations will be represented as `NA` in the `long` object.

Model 15: Linear Trajectory

In this model, the change of math achievement scores (`mathach`) across grade (`grade`) is modeled. Measurements are nested in students (`caseid`). Note that the schools are ignored here. We will take the schools into account in the three-level model section.

The grade variable is ranged from Grade 7 to 12. To make the intercept (β_{0j}) interpretable, the grade variable is centered at Grade 7:

```
long$gradec <- long$grade - 7
```

By this centering, the intercept will represent the math achievement of each student at Grade 7. The intercepts and slopes (linear change) are random across students. The model of linear trajectory would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}(t_{ij} - 7) + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + u_{1j}$$
$$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ \tau_{10} & \tau_{11} \end{bmatrix})$$

These notations should represent

- Y_{ij} = The math achievement score of Measurement *i* in Student *j*
- t_{ij} = The grade that the Measurement *i* in Student *j* was observed
- β_{0j} = The math achievement score of Student *j* at Grade 7
- β_{1j} = The expected change in math achievement score when grade increases by 1 for Student *j*, which is the rate of change for Student *j*
- γ_{00} = The average of math achievement scores in Grade 7 across students
- γ_{10} = The average rate of change in math achievement scores across students
- e_{ij} = The difference between the actual math achievement score of Measurement *i* in Student *j* and the expected score of Student *j* at a given grade level
- u_{0j} = The deviation of the actual math achievement score of Student *j* at Grade 7 from the average math achievement score at Grade 7 across students
- u_{1j} = The deviation of the rate of change of Student *j* from the average rate of change across students
- σ^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade
- τ_{00} = The variance of math achievement scores at Grade 7 across students

- τ_{11} = The variance of the rate of change in math achievement score across students
- τ_{10} = The covariance between the math achievement score at Grade 7 (initial status) and the rate of change
- $\rho_{10} = \tau_{10}/\sqrt{\tau_{00}\tau_{11}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)

Next, the model with linear trajectory can be run by the `lmer` function:

```
m15 <- lmer(mathach ~ 1 + gradec + (1 + gradec|caseid), data=long, REML=FALSE)

summary(m15)
```

```
Linear mixed model fit by maximum likelihood
Formula: mathach ~ 1 + gradec + (1 + gradec | caseid)
   Data: long
    AIC     BIC logLik deviance REMLdev
 130778 130825 -65383   130766  130773
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 caseid   (Intercept) 90.7239  9.5249
          gradec       2.3968  1.5482   0.315
 Residual             20.5660  4.5350
Number of obs: 19041, groups: caseid, 5858

Fixed effects:
            Estimate Std. Error t value
(Intercept) 50.80376    0.15039   337.8
gradec       3.39160    0.03529    96.1

Correlation of Fixed Effects:
       (Intr)
gradec -0.248
```

The mapping from the formula and reduced-form equation would be

```
mathach ~ 1 + gradec + (1 + gradec|caseid)
```

$$Y_{ij} = \underbrace{\gamma_{00}(1) + \gamma_{10}(t_{ij} - 7)}_{\text{Fixed Effect}} + \underbrace{u_{0j}(1) + u_{1j}(t_{ij} - 7)}_{\text{Random Effect}} + e_{ij}$$

Fixed Effect + Random Effect

The random effect of the slope (rate of change) can be tested by making the reference model without the random slope and comparing the reference model with Model 15 by deviance test:

```
m15a <- lmer(mathach ~ 1 + gradec + (1|caseid), data=long, REML=FALSE)

anova(m15a, m15)
```

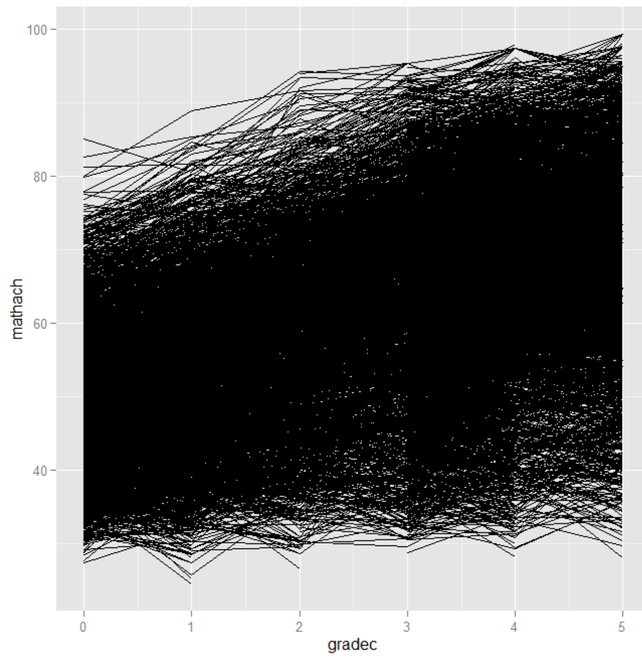
```
Data: long
Models:
m15a: mathach ~ 1 + gradec + (1 | caseid)
m15: mathach ~ 1 + gradec + (1 + gradec | caseid)
     Df    AIC    BIC logLik  Chisq Chi Df Pr(>Chisq)
m15a  4 132448 132480 -66220
m15   6 130778 130825 -65383 1674.6      2  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Students had different rates of change because the deviance test was significant, $\chi^2(2) = 1674.6$, $p < .001$. From the output of Model 15, students increased the math achievement scores by 3.39 points per grade. The rate of increases was different across students. In additional to the numeric output, a plot of individual trajectories would be helpful. We will use the `ggplot2` package to make the plot of individual trajectories, which, sometimes, is referred to as spaghetti plot.

```
library(ggplot2)
```

The plot of individual trajectories can be made:

```
ggplot(long, aes(x = gradec, y = mathach, group=caseid)) + geom_line()
```



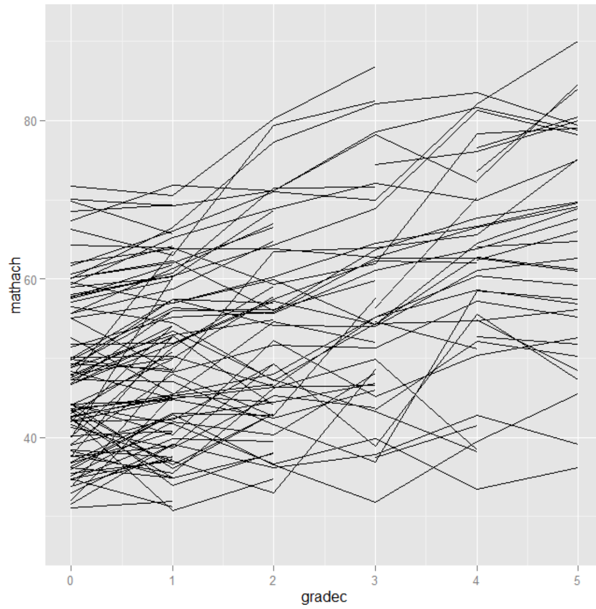
The framework of the `ggplot2` package is to build a template of graphic object by the `ggplot` function first and then add other components by the + sign. The template built by the `ggplot` function has two arguments. The first argument is the target dataset. The second argument is the list of attributes in the plot wrapped by the `aes` function. In this list, `x` is the variable on the x-axis, `y` is the variable on the y-axis, `group` represents the variable used for making separate lines. The template is added by the `geom_line` function, which is used to draw a line between individual data points in each group.

The graph above has too many lines. Let's make the plot for only first 100 students. A new data set of 100 students can be made:

```
long1 <- long[1:(6*100),]
```

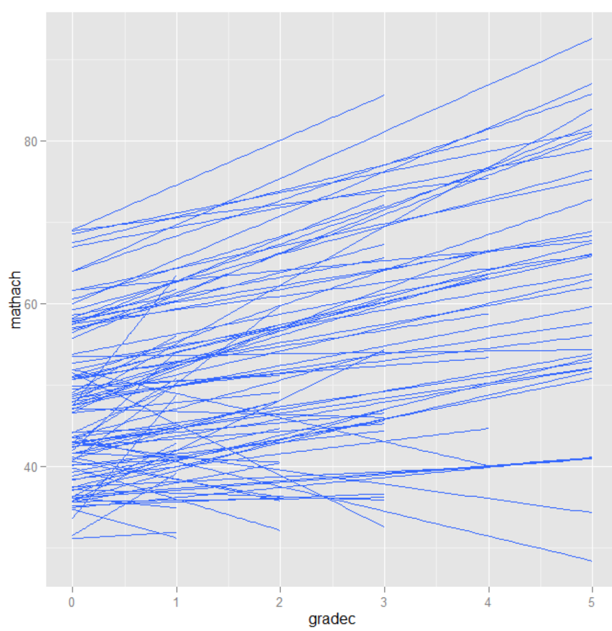
The first 600 rows (6 time points × 100 students) are selected. The `ggplot` function can be run again on the new data set.

```
ggplot(long1, aes(x = gradec, y = mathach, group=caseid)) + geom_line()
```



You may see the increasing trend from the graph. Users may wish to plot the linear trends of individual observations instead of the connecting lines between points. The plot of linear trajectories can be made by using the `geom_smooth` function:

```
ggplot(long1, aes(x = gradec, y = mathach, group=caseid)) + geom_smooth(method=lm, se=FALSE)
```



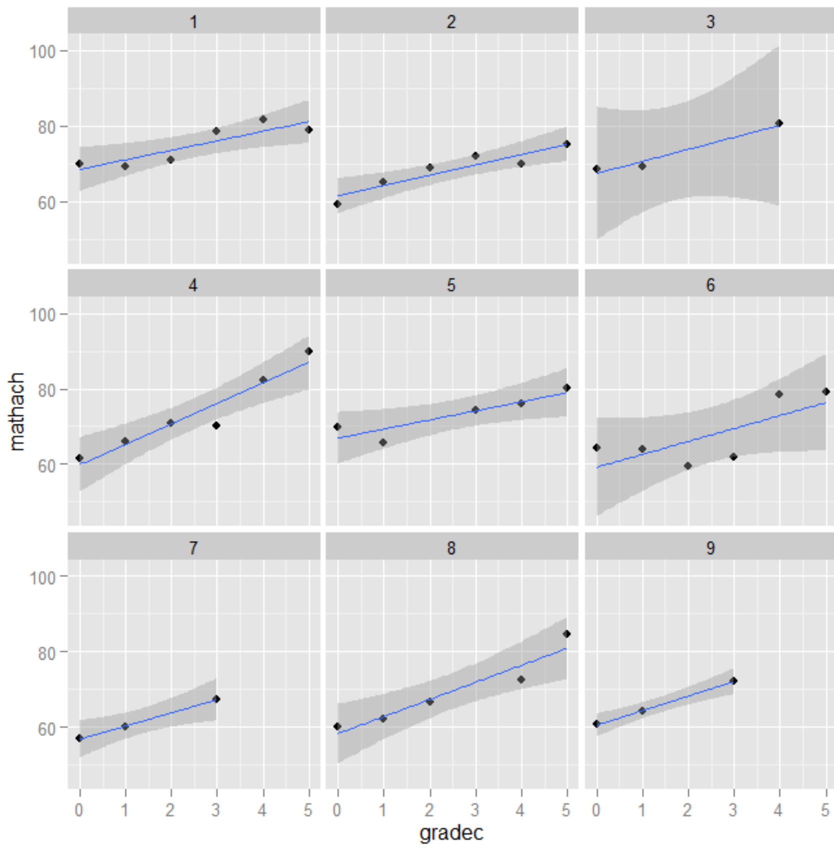
The `geom_smooth` function has two arguments. The first argument, `method`, is the function used to make each trajectory. The predicted values from the `lm` function will be used to make each trajectory. The second argument, `se`, is whether to plot the confidence band. Because we have many trajectories already, adding confidence bands will be not helpful. Note that you may see that most students have increasing trends where not many students have decreasing trends.[11]

Users may want to plot individual trajectories in different plots. Let's make 9 different plots for the first 9 students' trajectories. Initially, a data set of 9 students can be created:

```
long2 <- long[1:(6*9),]
```

Then, the plots can be created

```
ggplot(long2, aes(x = gradec, y = mathach)) + facet_wrap(~caseid) + geom_point() +
geom_smooth(method=lm, se=TRUE)
```



Within the template, the `group` argument in the `aes` function is not specified anymore because we want different plots (instead of different lines) for different students. Instead, the `facet_wrap` function is used to make different plots. The argument of the `facet_wrap` function is the grouping variable. The

[11] The regression coefficients obtained from the `lm` function (used to create linear trajectories) and the predicted value of the slope of each individual, β_{1j}, can be different. Multilevel model estimates the slope of each individual by using both math achievement scores of each student (information from Level 1) and the predicted value of the slope across student (information from L2). If a student has more measurements of math achievement, the combined slope will be leaning toward the information from Level 1 more. This concept can be referred to as Bayesian estimators (Raudenbush & Byrk, 2002). I use the `lm` function here to simply see the trend of the individual changes, not to find accurate rate of change of each student.

variable must begin with tilde, ~. The `geom_point` function is used to plot points in each graph. The `geom_smooth` function is used to plot a linear line in each graph with a confidence band (`se = TRUE`).

Model 16: Quadratic Trajectory

In this model, the change of math achievement scores (`mathach`) across grade (`grade`) is modeled as a quadratic trend. Grade is centered at Grade 7. The model of quadratic trajectory would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}\left(t_{ij} - 7\right) + \beta_{2j}\left(t_{ij} - 7\right)^2 + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\begin{aligned} \beta_{0j} &= \gamma_{00} + u_{0j} \\ \beta_{1j} &= \gamma_{10} + u_{1j} \\ \beta_{2j} &= \gamma_{20} + u_{2j} \end{aligned} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \\ u_{2j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & & \\ \tau_{10} & \tau_{11} & \\ \tau_{20} & \tau_{21} & \tau_{22} \end{bmatrix} \right)$$

These notations should represent (the blue lines indicate that the meanings changed from Model 15)

- Y_{ij} = The math achievement score of Measurement *i* in Student *j*
- t_{ij} = The grade that the Measurement *i* in Student *j* was observed
- β_{0j} = The math achievement score of Student *j* at Grade 7
- β_{1j} = The linear change (slope) in math achievement score for Student *j* at Grade 7
- β_{2j} = The change in linear slope of math achievement score for Student *j* when grade increases by 1, which is the curvature of the change for Student *j*
- γ_{00} = The average of math achievement score in Grade 7 across students
- γ_{10} = The average linear change (slope) in math achievement scores at Grade 7 across students
- γ_{20} = The average curvature of the change of math achievement scores across students
- e_{ij} = The difference between the actual math achievement score of Measurement *i* in Student *j* and the expected score of Student *j* at a given grade level
- u_{0j} = The deviation of the actual math achievement score of Student *j* at Grade 7 from the average math achievement score at Grade 7 across students
- u_{1j} = The deviation of the linear slope at Grade 7 of Student *j* from the average linear slope at Grade 7 across students
- u_{2j} = The deviation of the curvature of the change for Student *j* from the average curvature of the change across students
- σ^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade
- τ_{00} = The variance of math achievement score at Grade 7 across students
- τ_{11} = The variance of the linear slopes in math achievement score at Grade 7 across students
- τ_{22} = The variance of the curvature of the change across students
- τ_{10} = The covariance between the math achievement scores at Grade 7 (initial status) and the linear changes at Grade 7
- τ_{20} = The covariance between the math achievement scores at Grade 7 (initial status) and the curvatures of the change
- τ_{21} = The covariance between the linear changes at Grade 7 and the curvatures of the change

- $\rho_{st} = \tau_{st}/\sqrt{\tau_{ss}\tau_{tt}}$ (where $s, t = 0$, 1, or 2 and $s \neq t$) = The covariance mentioned above in the correlation scale (from -1 to 1)

Next, the model with quadratic trajectory can be run by the `lmer` function:

```
m16 <- lmer(mathach ~ 1 + gradec + I(gradec^2) + (1 + gradec + I(gradec^2)|caseid), data=long,
REML=FALSE)

summary(m16)
```

```
Linear mixed model fit by maximum likelihood
Formula: mathach ~ 1 + gradec + I(gradec^2) + (1 + gradec + I(gradec^2) |      caseid)
   Data: long
    AIC     BIC logLik deviance REMLdev
 130190 130269 -65085   130170  130184
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 caseid   (Intercept) 80.37148 8.96501
          gradec       9.05050 3.00841   0.394
          I(gradec^2)  0.20704 0.45501  -0.352 -0.855
 Residual             18.09922 4.25432
Number of obs: 19041, groups: caseid, 5858

Fixed effects:
            Estimate Std. Error t value
(Intercept) 49.89783    0.15120   330.0
gradec       4.57557    0.08503    53.8
I(gradec^2) -0.23520    0.01555   -15.1

Correlation of Fixed Effects:
            (Intr) gradec
gradec      -0.285
I(gradec^2)  0.203 -0.912
```

In the formula, the `gradec` variable is squared representing the quadratic term. However, if the `gradec^2` is simply put in the formula, R will evaluate the expression first (before feeding in the function) and the error will occur. Alternatively, we want R to evaluate the expression inside the function so the `I` function is needed to bracket the squared term to make R hold the expression and evaluate inside the function. The quadratic term, `I(gradec^2)`, is added in both fixed effect and random effect.

The mapping from the formula and reduced-form equation would be

mathach ~ **1** + **gradec** + I(**gradec**^2) + (**1** + **gradec** + I(**gradec**^2)|**caseid**)
$Y_{ij} = \gamma_{00}(1) + \gamma_{10}(t_{ij} - 7) + \gamma_{20}(t_{ij} - 7)^2 + u_{0j}(1) + u_{1j}(t_{ij} - 7) + u_{2j}(t_{ij} - 7)^2 + e_{ij}$ Fixed Effect + Random Effect

In adding the quadratic term, we can test whether the curvature of the change is different from 0 on average and whether the curvature of the change is random across students. For the first test, the reference model with a fixed curvature is created and then compared with the model with the linear model, Model 15:

```
m16a <- lmer(mathach ~ 1 + gradec + I(gradec^2) + (1 + gradec|caseid), data=long, REML=FALSE)

anova(m15, m16a)
```

```
Data: long
Models:
m15: mathach ~ 1 + gradec + (1 + gradec | caseid)
m16a: mathach ~ 1 + gradec + I(gradec^2) + (1 + gradec | caseid)
     Df    AIC    BIC logLik  Chisq Chi Df Pr(>Chisq)
m15   6 130778 130825 -65383
m16a  7 130445 130500 -65215 335.32      1  < 2.2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

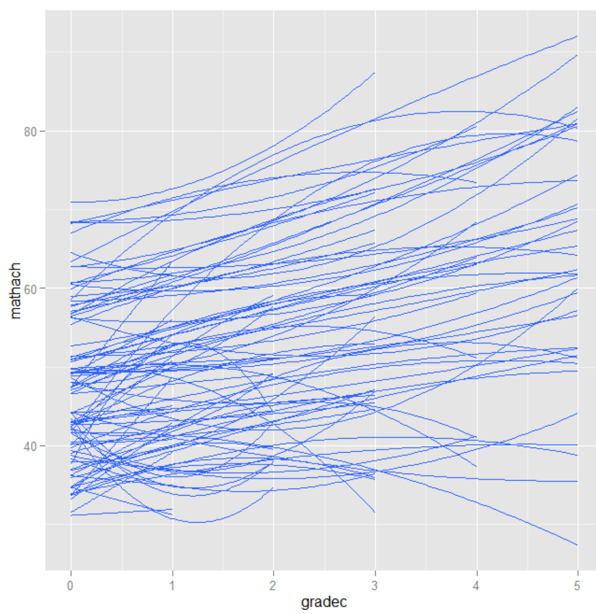
Note that the reference model does not have the quadratic term in the random effect (in parenthesis) part. Adding the fixed curvature explained the model better than the model with linear trend, $\chi^2(1) = 335.32$, $p < .001$. The reference model with the fixed curvature can be compared with the current model, Model 16, which has random curvatures:

```
anova(m16a, m16)
```

```
Data: long
Models:
m16a: mathach ~ 1 + gradec + I(gradec^2) + (1 + gradec | caseid)
m16: mathach ~ 1 + gradec + I(gradec^2) + (1 + gradec + I(gradec^2) |
m16:     caseid)
     Df    AIC    BIC logLik  Chisq Chi Df Pr(>Chisq)
m16a  7 130445 130500 -65215
m16  10 130190 130269 -65085 260.28      3  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Because the test statistic was significant, $\chi^2(1) = 260.28$, $p < .001$, the curvature of the change was random across students. Users may compare Model 15 and Model 16 directly (please imagine what does this deviance test represent). We can use the `ggplot2` package to investigate the quadratic changes of the students. Note that we will use the subsets of the whole data set (`long1` and `long2`) that we created in Model 15. First, the changes can be plotted in a single graph:

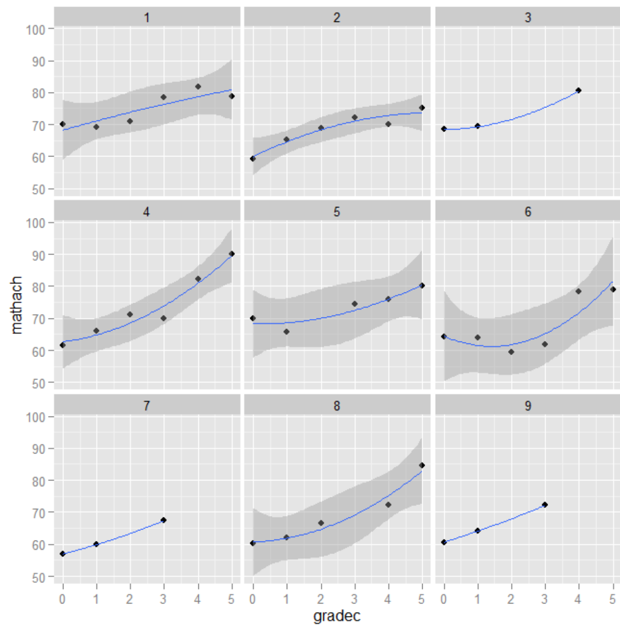
```
ggplot(long1, aes(x = gradec, y = mathach, group=caseid)) + geom_smooth(method = "lm", formula =
y ~ x + I(x^2), se=FALSE)
```



The code is similar to plotting linear trend in Model 15; however, the formula argument is added in the `geom_smooth` function. The formula has the squared term, `I(x^2)`, to represent a quadratic change. We used `x` and `y` in the formula (instead of `gradec` and `mathach`) because x and y were defined in the graph template from the `ggplot` function. The graph showed that some students have a concaving-up change where other students have a concaving-down change.

The `formula` argument can be specified in the `geom_smooth` function when plotting individual changes:

```
ggplot(long2, aes(x = gradec, y = mathach)) + facet_wrap(~caseid) + geom_point() +
geom_smooth(method = "lm", formula = y ~ x + I(x^2), se=TRUE)
```



Note that standard errors cannot be calculated when a student has only three observations. A quadratic trend will fit the data with three observations perfectly.

Model 17: Linear Trajectory with Time-Invariant Covariate

In this model, similar to Model 15, the linear change of math achievement scores (`mathach`) across grade (`grade`) is modeled. The grade variable is centered at Grade 7 so the intercept will represent the math achievement of each student at Grade 7. The intercepts and slopes (linear change) are random across students and predicted by gender (females as the reference group), which is a time-invariant covariate. The model of linear trajectory with time-invariant covariate would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}(t_{ij} - 7) + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + \gamma_{01}W_j + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + \gamma_{11}W_j + u_{1j} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ \tau_{10} & \tau_{11} \end{bmatrix}\right)$$

These notations should represent (the blue lines indicate that the meanings changed from Model 15)

- Y_{ij} = The math achievement score of Measurement *i* in Student *j*
- t_{ij} = The grade that the Measurement *i* in Student *j* was observed
- β_{0j} = The math achievement score of Student *j* at Grade 7
- β_{1j} = The expected change in math achievement score when grade increases by 1 for Student *j*, which is the rate of change for Student *j*
- γ_{00} = The average of math achievement scores in Grade 7 across female students

- γ_{01} = The difference of the average math achievement scores in Grade 7 between male and female students
- γ_{10} = The average rate of change in math achievement score across female students
- γ_{11} = The difference of the average rate of change in math achievement scores between male and female students
- e_{ij} = The difference between the actual math achievement score of Measurement *i* in Student *j* and the expected score of Student *j* at a given grade level
- u_{0j} = The deviation of the actual math achievement score of Student *j* at Grade 7 from the average math achievement score at Grade 7 across students with the same sex as Student *j*
- u_{1j} = The deviation of the rate of change of Student *j* from the average rate of change across students with the same sex as Student *j*
- σ^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade
- τ_{00} = The residual variance of math achievement score at Grade 7 across students controlling for sex
- τ_{11} = The residual variance of the rate of change in math achievement score across students controlling for sex
- τ_{10} = The residual covariance between the math achievement score at Grade 7 (initial status) and the rate of change controlling for sex
- $\rho_{10} = \tau_{10}/\sqrt{\tau_{00}\tau_{11}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)

Before running the model, we need to transform the gender variable into the factor format:

```
long$gender <- factor(long$gender, labels=c("female", "male"))
```

Next, the model of linear trajectory with time-invariant covariate can be run by the `lmer` function:

```
m17 <- lmer(mathach ~ 1 + gradec + gender + gradec*gender + (1 + gradec|caseid), data=long,
REML=FALSE)

summary(m17)
```

```
Linear mixed model fit by maximum likelihood
Formula: mathach ~ 1 + gradec + gender + gradec * gender + (1 + gradec |      caseid)
   Data: long
    AIC    BIC logLik deviance REMLdev
 130755 130817 -65369   130739  130750
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 caseid   (Intercept) 90.3056  9.5029
          gradec       2.3717  1.5400   0.322
 Residual             20.5786  4.5364
Number of obs: 19041, groups: caseid, 5858

Fixed effects:
                  Estimate Std. Error t value
(Intercept)       51.32442    0.21464  239.12
gradec             3.22606    0.04976   64.83
gendermale        -1.02010    0.30019   -3.40
gradec:gendermale  0.32983    0.07038    4.69

Correlation of Fixed Effects:
            (Intr) gradec gndrml
gradec      -0.248
gendermale  -0.715  0.177
grdc:gndrml  0.175 -0.707 -0.246
```

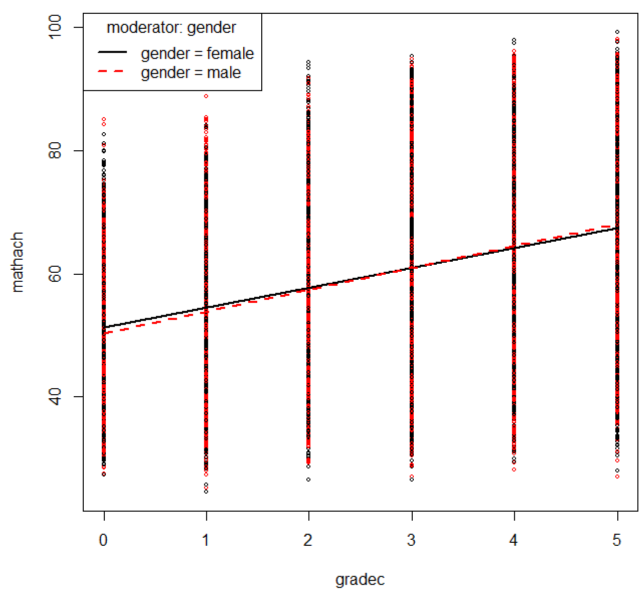
The mapping from the formula and reduced-form equation would be

```
mathach ~ 1 + gradec + gender + gradec*gender + (1 + gradec|caseid)
```

$$Y_{ij} = \underbrace{\gamma_{00}(1) + \gamma_{10}(t_{ij} - 7) + \gamma_{01}W_j + \gamma_{11}(t_{ij} - 7)W_j}_{\text{Fixed Effect}} + \underbrace{u_{0j}(1) + u_{1j}(t_{ij} - 7)}_{\text{Random Effect}} + e_{ij}$$

The effect of sex on both random intercept and random slope can be tested simultaneously by comparing the current model with Model 15 by the deviance test. Using the `anova` function, the test was significant, $\chi^2(2) = 27.24$, $p < .001$. From the fixed effects, male students had significantly lower math achievement scores than females in Grade 7. Male students, however, had significantly higher rate of increase in math achievement than females. To probe the rate of change in each gender, we probe the cross-level interaction (`gradec*gender`), which relates with one continuous variable (`gradec`) and one categorical variable (`gender`). We can use the `rockchalkMultilevel` package to probe the interaction as explained above:

```
simpleSlope17 <- plotSlopes.mlm(m17, "gradec", "gender")
```



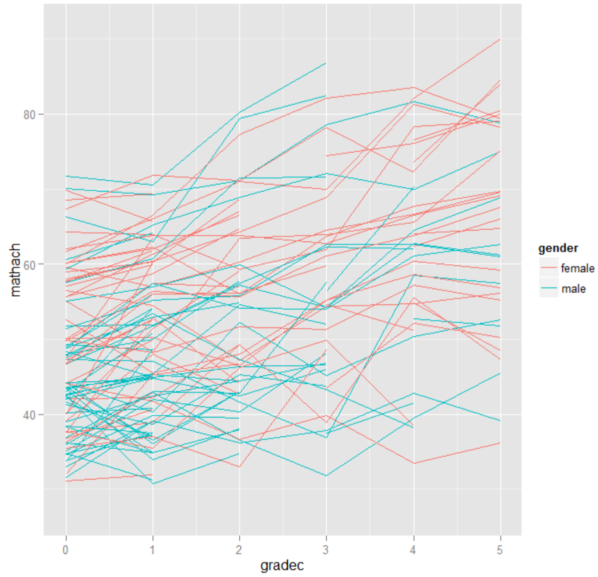

```
testSlopes.mlm(simpleSlope17)
```

```
These are the straight-line "simple slopes" of the variable gradec
 for the selected moderator values.
                "gender"     slope Std. Error  z value Pr(>|z|)
female            gradec 3.226064 0.04976438 64.82676        0
male   gradec:gendermale 3.555891 0.04976258 71.45713        0
```

We can see from the graph that, even though the effects of sex were significant, the sizes of the effects were not large. Note that 5,858 students were observed in this data so the test statistic was significant even though the size of effect was small. From the result of testing simple slopes, the rates of change of each gender were significantly greater than 0. We may plot individual trajectories where the lines of each gender have different colors by the `ggplot2` package:

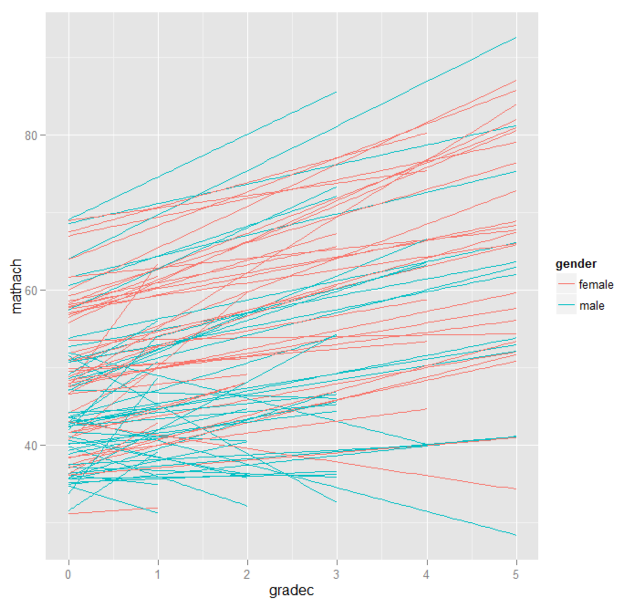
```
long1 <- long[1:(6*100),]
```

```
ggplot(long1, aes(x = gradec, y = mathach, group=caseid, colour=gender)) + geom_line()
```



The `long1` dataset needs to be extracted again because we change the `gender` variable to the factor format in the original data. In the graph template, we can simply put the `gender` variable in the `colour` argument in the `aes` function. Similarly, we can plot the linear trajectories instead:

```
ggplot(long1, aes(x = gradec, y = mathach, group=caseid, colour=gender)) + geom_smooth(method=lm,
se=FALSE)
```



Readers may try to plot different plots for different trajectories where the colors of the lines are varied by gender.

Model 18: Linear Trajectory with Time-Varying Covariate

In this model, similar to Model 15, the linear change of math achievement scores (`mathach`) across grade (`grade`) is modeled. In this model, the parent encouragement (`parentpush`) and peer encouragement (`peerpush`) on studying math are used as time-varying covariates. The `grade` variable is centered at Grade 7 and the parent and peer encouragements are centered at their grand mean. Group-mean centering may be more appropriate in this case but I use grand-mean centering for the sake of simplicity. The effects of peer and parent encouragements are fixed across students. The model with time-varying covariates would be

L1
$$Y_{ij} = \beta_{0j} + \beta_{1j}(t_{ij} - 7) + \beta_{2j}(X_{1ij} - \bar{X}_{1..}) + \beta_{3j}(X_{2ij} - \bar{X}_{2..}) + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2
$$\beta_{0j} = \gamma_{00} + u_{0j}; \quad \beta_{1j} = \gamma_{10} + u_{1j} \qquad \beta_{2j} = \gamma_{20}; \quad \beta_{3j} = \gamma_{30} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ \tau_{10} & \tau_{11} \end{bmatrix}\right)$$

These notations should represent (the blue lines indicate that the meanings changed from Model 15)

- Y_{ij} = The math achievement score of Measurement *i* in Student *j*
- t_{ij} = The grade that the Measurement *i* in Student *j* was observed
- X_{1ij} = The parent encouragement score of Measurement *i* in Student *j*
- X_{2ij} = The peer encouragement score of Measurement *i* in Student *j*
- β_{0j} = The math achievement score of Student *j* at Grade 7 given that the parent and peer encouragements equal to their grand means
- β_{1j} = The expected change in math achievement score when grade increases by 1 controlling for parent and peer encouragement scores for Student *j*, which is the adjusted rate of change for Student *j*
- β_{2j} = The increase in math achievement score when parent encouragement score increases by 1 at the same grade controlling for peer encouragement score for Student *j*
- β_{2j} = The increase in math achievement score when peer encouragement score increases by 1 at the same grade controlling for parent encouragement score for Student *j*
- γ_{00} = The average of math achievement scores in Grade 7 across students when parent and peer achievement scores equal to their grand mean
- γ_{10} = The average adjusted rate of change in math achievement scores controlling for parent and peer achievement scores across students
- γ_{20} = The effect of parent encouragement controlling for grade and peer encouragement, which is constant across students
- γ_{30} = The effect of peer encouragement controlling for grade and parent encouragement, which is constant across students
- e_{ij} = The difference between the actual math achievement score of Measurement *i* in Student *j* and the expected score of Student *j* at given grade, parent encouragement, and peer encouragement
- u_{0j} = The deviation of the actual adjusted math achievement score of Student *j* at Grade 7 from the average adjusted math achievement score at Grade 7 across students

- u_{1j} = The deviation of the adjusted rate of change of Student *j* from the average adjusted rate of change across students
- σ^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade, parent encouragement, and peer encouragement
- τ_{00} = The residual variance of math achievement scores at Grade 7 across students controlling for parent and peer encouragements
- τ_{11} = The residual variance of the rate of change in math achievement score across students controlling for parent and peer encouragements
- τ_{10} = The residual covariance between the math achievement score at Grade 7 (initial status) and the rate of change controlling for parent and peer encouragements
- $\rho_{10} = \tau_{10}/\sqrt{\tau_{00}\tau_{11}}$ = The residual covariance mentioned above in the correlation scale (from -1 to 1)

Before running the model, parent and peer encouragements need to be centered at their grand mean:

```
long$parentpushC <- long$parentpush - mean(long$parentpush, na.rm=TRUE)

long$peerpushC <- long$peerpush - mean(long$peerpush, na.rm=TRUE)
```

The `na.rm` argument of the `mean` function is specified as `TRUE` to find the mean by skipping missing observations.[12] Next, the model of linear trajectory with time-varying covariates can be run by the `lmer` function:

```
m18 <- lmer(mathach ~ 1 + gradec + parentpushC + peerpushC + (1 + gradec|caseid), data=long,
REML=FALSE)

summary(m18)
```

```
Linear mixed model fit by maximum likelihood
Formula: mathach ~ 1 + gradec + parentpushC + peerpushC + (1 + gradec |      caseid)
   Data: long
    AIC    BIC logLik deviance REMLdev
 119891 119953 -59938   119875  119890
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 caseid   (Intercept) 91.2917  9.5547
          gradec       2.3217  1.5237   0.284
 Residual             19.6052  4.4278
Number of obs: 17441, groups: caseid, 5833

Fixed effects:
            Estimate Std. Error t value
(Intercept) 50.49264    0.15799   319.6
```

---

[12] For the group-mean centering, a little trick is needed to find the group means by skipping missing observations. First, a new function, `meanc`, is defined as the `mean` function with the `na.rm` argument specified as `TRUE`:

```
meanc <- function(x) mean(x, na.rm=TRUE)
```

Next, the `meanc` function is used in the `ave` function to find the mean by skipping missing observations:

```
long$parentpushGroupM <- ave(long$parentpush, long$caseid, FUN=meanc)

long$peerpushGroupM <- ave(long$peerpush, long$caseid, FUN=meanc)

long$parentpushGroupC <- long$parentpush - long$parentpushGroupM

long$peerpushGroupC <- long$peerpush - long$peerpushGroupM
```

```
gradec       3.54045    0.03979    89.0
parentpushC  0.36923    0.05448     6.8
peerpushC   -0.12745    0.04676    -2.7

Correlation of Fixed Effects:
            (Intr) gradec prntpC
gradec      -0.356
parentpushC -0.235  0.322
peerpushC   -0.127  0.185 -0.142
```

The mapping from the formula and reduced-form equation would be

```
mathach ~ 1 + gradec + parentpushC + peerpushC + (1 + gradec|caseid)
```

$$Y_{ij} = \gamma_{00}(1) + \gamma_{10}(t_{ij} - 7) + \gamma_{20}(X_{1ij} - \bar{X}_{1..}) + \gamma_{30}(X_{2ij} - \bar{X}_{2..}) + u_{0j}(1) + u_{1j}(t_{ij} - 7) + e_{ij}$$

Fixed Effect + Random Effect

This model can be compared with Model 15 by deviance test. The result was significant, $\chi^2(2) = 10891$, $p < .001$, meaning that parent and peer encouragements significantly explained math achievement scores. Next, the current model can be compared with the model with the random slopes of parent and peer encouragements. Readers may try it. The result of the deviance test would indicate significant random effects, $\chi^2(7) = 23.628$, $p = .001$.

Model 19: Linear Trajectory with Heterogeneity of Variance

This model is similar to Model 15 but the error variances at each time point are not equal, which is referred to as heteroscedastic error variances. The equations are similar to Model 15, but the error variances, σ^2, depends on time point, $\sigma^2_{(e_{ij}|t_{ij})}$. The `lme4` package cannot analyze the model with heteroscedasitc L1 error variances. We will use the `nlme` package instead.

```
library(nlme)
```

Before analyzing the model with heteroscedastic L1 error variances, let's run Model 15 by the `nlme` package first. We will use the `lme` function. The structure is very similar to the `lmer` function from the `lme4` package that we have discussed so far:

```
m15nlme <- lme(mathach ~ 1 + gradec, random = ~1 + gradec|caseid, data=long, method="ML",
na.action=na.omit)

summary(m15nlme)
```

Output from `nlme` package (`m15nlme`)

```
Linear mixed-effects model fit by maximum likelihood
 Data: long
       AIC      BIC    logLik
  130777.8 130825 -65382.92

Random effects:
 Formula: ~1 + gradec | caseid
 Structure: General positive-definite, Log-Cholesky
parametrization
            StdDev   Corr
(Intercept) 9.524898 (Intr)
gradec      1.548169 0.315
Residual    4.534982

Fixed effects: mathach ~ 1 + gradec
               Value  Std.Error    DF  t-value p-value
(Intercept) 50.80378 0.15039556 13182 337.8011       0
gradec       3.39158 0.03529208 13182  96.1004       0
 Correlation:
       (Intr)
gradec -0.248
```

Output from `lme4` package (`m15`)

```
Linear mixed model fit by maximum likelihood
Formula: mathach ~ 1 + gradec + (1 + gradec | caseid)
   Data: long
    AIC    BIC logLik deviance REMLdev
 130778 130825 -65383   130766  130773
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 caseid   (Intercept) 90.7239  9.5249
          gradec       2.3968  1.5482   0.315
 Residual             20.5660  4.5350
Number of obs: 19041, groups: caseid, 5858

Fixed effects:
            Estimate Std. Error t value
(Intercept) 50.80376    0.15039   337.8
gradec       3.39160    0.03529    96.1

Correlation of Fixed Effects:
       (Intr)
gradec -0.248
```

```
Standardized Within-Group Residuals:
      Min        Q1        Med        Q3       Max 
-5.943033 -0.475696  0.0172878  0.499919  4.136147 

Number of Observations: 19041
Number of Groups: 5858
```

The major difference between the `lme` and `lmer` functions in this code is the specification of the random effects. In the `lmer` function, random effects are specified in a parenthesis and added in the formula. The `lme` function, however, random effects are specified in the `random` argument. Users simply remove the parenthesis, begin the code with the tilde, and put in the `random` argument. The `method` argument is the method of estimation, which can be specified as `"ML"` (Full maximum likelihood) or `"REML"` (residual maximum likelihood). The `na.action` is to specify how to handle missing observations where `na.omit` is to use listwise deletion. Users can see that the results from the `lme4` and `nlme` packages are almost identical.

Next, we can specify the different L1 error variances across time from the output of the `lme` function. The `update` function is used to update the original model by releasing the constraints of equal error variances across time:

```
m19 <- update(m15nlme, weight=varIdent(form = ~1|gradec))

summary(m19)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: long 
       AIC      BIC    logLik
  130691.7 130778.1 -65334.83

Random effects:
 Formula: ~1 + gradec | caseid
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev   Corr  
(Intercept) 9.448993 (Intr)
gradec      1.574986 0.304 
Residual    4.342976       

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | gradec 
 Parameter estimates:
        0         1         2         3         4         5 
1.0000000 1.0046543 1.0498829 1.1845033 0.9226115 1.0300098 
Fixed effects: mathach ~ 1 + gradec 
               Value  Std.Error    DF  t-value p-value
(Intercept) 50.79434 0.14963805 13182 339.4480       0
gradec       3.40003 0.03526212 13182  96.4217       0
 Correlation: 
       (Intr)
gradec -0.248

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max 
-5.86326644 -0.47615508  0.02134337  0.50000963  4.24934381 

Number of Observations: 19041
Number of Groups: 5858
```

The `weight` argument is used to set the different error variances where `varIdent(form = ~1|gradec)` means that the variances are set to be equal for those observations coming from the same grade. That is, the observations from different grades can have different variances.[13]

[13] Instead of using the `update` function, users may run the heteroscedastic model directly by specifying the `weight` argument in the `lme` function:

The output will provide the `Variance function` section. In the `Parameter estimates` subsection, those values mean the ratio of the standard deviations of residuals of a given `gradec` value over the standard deviations of residuals with `gradec` of 0 (Grade 7). The following R script can be used to calculate the standard deviation of residual variances at each time point:

```
l1sd <- as.numeric(VarCorr(m19)[3, 2])

l1sd * coef(m19$modelStruct$varStruct, uncons = FALSE)
```

```
       1        2        3        4        5
4.363190 4.559616 5.144269 4.006880 4.473308
```

The `VarCorr` function is used to extract the standard deviations (or variances) of the random effects. The element `3, 2` is the L1 residual standard deviation of Grade 7. Because the value is in the text (string) format, the `as.numeric` function is used to change the text to number. Next, loosely speaking, `coef(m19$modelStruct$varStruct, uncons = FALSE)` is used to extract the parameter estimates of the variance function provided from the summary of the output, which is the ratio of standard deviations across time points. The ratio can be multiplied by the L1 residual standard deviation of Grade 7 to get the residual standard deviation of all time points.

This model can be compared with the model with equal L1 error variances, Model 15, by the `anova` function:

```
anova(m15nlme, m19)
```

```
        Model df      AIC      BIC    logLik   Test L.Ratio p-value
m15nlme     1  6 130777.8 130825.0 -65382.92
m19         2 11 130691.7 130778.1 -65334.83 1 vs 2 96.1805  <.0001
```

The test statistic was significant, $\chi^2(5) = 96.18$, $p < .001$, meaning that the L1 error variances were significantly different across time points. The degree of freedom can be calculated from the difference between degrees of freedom of two models (11 – 6 = 5).

Model 20: Linear Trajectory with First-Order Autocorrelation

Multilevel models assume that errors are independent. In longitudinal model, the errors from adjacent time points can be more similar than the errors from the distant time points. Thus, the error correlation structure will be specified in this model.

From Model 15, the errors of math achievement score across time points are assumed to be correlated by the first-order autocorrelation. That is, the error correlation matrix would be

---

```
m19 <- lme(mathach ~ 1 + gradec, random = ~1 + gradec|caseid, data=long, method="ML",
na.action=na.omit, weight=varIdent(form = ~1|gradec))
```

The `update` function is more convenient when users wish to adjust the original model. Users simply adjust the original model and the following models will be automatically adjusted.

$$\begin{bmatrix} 1 & & & & & \\ \rho & 1 & & & & \\ \rho^2 & \rho & 1 & & & \\ \rho^3 & \rho^2 & \rho & 1 & & \\ \rho^4 & \rho^3 & \rho^2 & \rho & 1 & \\ \rho^5 & \rho^4 & \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$$

where rows and columns of the matrix represent the errors at Grade 7-12. The interpretation of this model is similar to Model 15. Because the `lme4` package cannot be used for specifying L1 error structure, the autocorrelation is specified by the `nlme` package:

```
m20 <- update(m15nlme, correlation=corARMA(p = 1))

summary(m20)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: long
       AIC      BIC    logLik
  130473.8 130528.8 -65229.92

Random effects:
 Formula: ~1 + gradec | caseid
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev   Corr
(Intercept) 8.717791 (Intr)
gradec      1.093237 0.764
Residual    5.578114

Correlation Structure: AR(1)
 Formula: ~1 | caseid
 Parameter estimate(s):
      Phi
0.3716054
Fixed effects: mathach ~ 1 + gradec
               Value  Std.Error    DF  t-value p-value
(Intercept) 50.82123 0.14999493 13182 338.8197       0
gradec       3.34795 0.03526062 13182  94.9487       0
 Correlation:
       (Intr)
gradec -0.255

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-4.87351623 -0.42069465  0.04564341  0.50304165  3.74030927
```

The `correlation` argument is used to specify the L1 error correlation structure where `corARMA` is the correlation structure based on auto regression and moving average. In the `corARMA` function, `p` is the order of autocorrelation[14], which is specified as `1` here. Users may use `corAR1()`, which is the same thing as `corARMA(p = 1)`.[15]

The autocorrelation is provided in the `Correlation Structure` section, which is .37. That is, the errors of adjacent time points are more similar than distant time points. This model can be compared with Model 15 by the deviance test:

```
anova(m15nlme, m20)
```

```
        Model df      AIC      BIC    logLik   Test L.Ratio p-value
m15nlme     1  6 130777.8 130825.0 -65382.92
```

[14] Users may specify the order of moving average by the `q` argument.
[15] Instead of using the `update` function, the `lme` function can be run directly by specifying the `correlation` argument in the function.

```
m20          2  7 130473.8 130528.8 -65229.92 1 vs 2      306  <.0001
```

The autocorrelation was significant, $\chi^2(1) = 306$, $p < .001$. Readers are encouraged to specify the autocorrelation greater than the first order or specify the autocorrelation with heteroscedastic errors.

Model 21: Piecewise Linear Trajectory

In this model, the change of math achievement scores (`mathach`) across grade (`grade`) is separated into two phases: junior high school (Grade 7-9) and high school (Grade 10-12). Researchers may think that the rates of change are different during two periods of time. The `grade` variable is separated into two variables to represent two different periods.

$$t_{ij} - 7 = t_{1ij} + t_{2ij}$$

where t_{ij} is the grade that the Measurement *i* in Student *j* was observed, t_{1ij} represents the change during junior high, and t_{2ij} represents the change during high school. We may specify the values of t_{1ij} and t_{2ij} as in the following table:

Grade (t_{ij})	Junior High School (t_{1ij})	High School (t_{2ij})
7	0	0
8	1	0
9	2	0
10	2	1
11	2	2
12	2	3

Notice that the values in each row satisfy the equation. Two variables (`gradec1` and `gradec2`) can be created to represent the changes during junior high school and high school:

```
long$gradec1 <- long$gradec

long$gradec1[long$gradec1 %in% c(3, 4, 5)] <- 2

long$gradec2 <- long$gradec - long$gradec1
```

First, we create the `gradec1` variable as a replicate of the centered grade variable, `gradec`. Second, any values of the `gradec1` variable that are equal to 3, 4, or 5 are recoded as 2. The `%in%` operator is to check whether a value on the left hand side is equal to any values on the right hand side. In this case, if any values in the variable on the left hand side are equal to 3, 4, or 5, the results will be `TRUE` and those cases are selected by the square bracket and recoded as 2. In any values in the variable on the right hand side are not equal to 3, 4, or 5 (i.e., 0, 1, or 2), the results will be `FALSE` and those cases are not selected. Finally, the `gradec2` variable is simply calculated by subtracting the `gradec` variable by the `gradec1` variable.

The changes in each phase are random in this model. The model of piecewise linear trajectory would be

L1 $$Y_{ij} = \beta_{0j} + \beta_{1j}t_{1ij} + \beta_{2j}t_{2ij} + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2)$$

L2 $$\beta_{0j} = \gamma_{00} + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + u_{1j}$$
$$\beta_{2j} = \gamma_{20} + u_{2j}$$
$$\begin{bmatrix} u_{0j} \\ u_{1j} \\ u_{2j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & & \\ \tau_{10} & \tau_{11} & \\ \tau_{20} & \tau_{21} & \tau_{22} \end{bmatrix}\right)$$

These notations should represent (the blue lines indicate that the meanings changed from Model 15)

- Y_{ij} = The math achievement score of Measurement i in Student j
- β_{0j} = The math achievement score of Student j at Grade 7 (both t_{1ij} and t_{2ij} are 0).
- β_{1j} = The expected change in math achievement score when grade during junior high school increases by 1 for Student j, which is the rate of change during junior high school for Student j
- β_{2j} = The expected change in math achievement score when grade during high school increases by 1 for Student j, which is the rate of change during high school for Student j
- γ_{00} = The average of math achievement scores in Grade 7 across students
- γ_{10} = The average rate of change during junior high school in math achievement scores across students
- γ_{20} = The average rate of change during high school in math achievement scores across students
- e_{ij} = The difference between the actual math achievement score of Measurement i in Student j and the expected score of Student j at a given grade level
- u_{0j} = The deviation of the actual math achievement score of Student j at Grade 7 from the average math achievement score at Grade 7 across students
- u_{1j} = The deviation of the rate of change during junior high school of Student j from the average rate of change during junior high school across students
- u_{2j} = The deviation of the rate of change during high school of Student j from the average rate of change during high school across students
- σ^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade
- τ_{00} = The variance of math achievement scores at Grade 7 across students
- τ_{11} = The variance of the rate of change in math achievement score during junior high school across students
- τ_{22} = The variance of the rate of change in math achievement score during high school across students
- τ_{10} = The covariance between the math achievement score at Grade 7 (initial status) and the rate of change during junior high school
- τ_{20} = The covariance between the math achievement score at Grade 7 (initial status) and the rate of change during high school
- τ_{21} = The covariance between the rate of change during junior high school and the rate of change during high school
- $\rho_{st} = \tau_{st}/\sqrt{\tau_{ss}\tau_{tt}}$ (where $s, t = 0$, 1, or 2 and $s \neq t$) = The covariance mentioned above in the correlation scale (from -1 to 1)

Next, the model with linear trajectory can be run by the `lmer` function (from the `lme4` package):

```
m21 <- lmer(mathach ~ 1 + gradec1 + gradec2 + (1 + gradec1 + gradec2|caseid), data=long,
REML=FALSE)

summary(m21)
```

```
Linear mixed model fit by maximum likelihood
Formula: mathach ~ 1 + gradec1 + gradec2 + (1 + gradec1 + gradec2 | caseid)
   Data: long
    AIC    BIC logLik deviance REMLdev
 130311 130390 -65146   130291  130301
Random effects:
```

```
 Groups   Name        Variance Std.Dev. Corr
 caseid   (Intercept) 81.3419  9.0190
          gradec1      6.3483  2.5196   0.408
          gradec2      3.4733  1.8637   0.002 0.134
 Residual             18.1122  4.2558
Number of obs: 19041, groups: caseid, 5858

Fixed effects:
            Estimate Std. Error t value
(Intercept) 50.13468    0.15022   333.7
gradec1      4.09437    0.06834    59.9
gradec2      2.98733    0.05176    57.7

Correlation of Fixed Effects:
        (Intr) gradc1
gradec1 -0.272
gradec2 -0.040 -0.287
```

The mapping from the formula and reduced-form equation would be

```
mathach ~ 1 + gradec1 + gradec2 + (1 + gradec1 + gradec2|caseid)
```

$$Y_{ij} = \underbrace{\gamma_{00}(1) + \gamma_{10}t_{1ij} + \gamma_{20}t_{2ij}}_{\text{Fixed Effect}} + \underbrace{u_{0j}(1) + u_{1j}t_{1ij} + u_{2j}t_{2ij}}_{\text{Random Effect}} + e_{ij}$$

The linear growth model, Model 15, is nested in this model. If the linear change in the first and second phases, β_{1j} and β_{2j}, are equal in every student, this model will be Model 15. Therefore, we can test whether the two-phase change is better than the one-phase change by the deviance test:

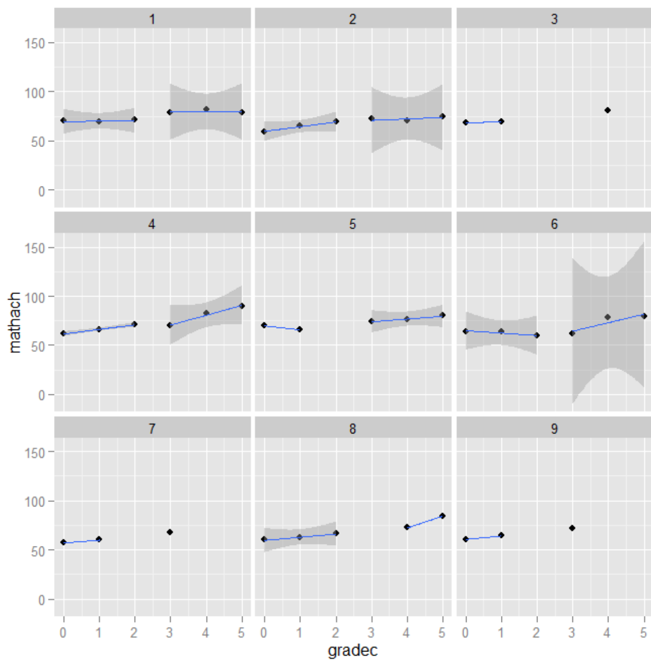
```
> anova(m15, m21)
```

```
Data: long
Models:
m15: mathach ~ 1 + gradec + (1 + gradec | caseid)
m21: mathach ~ 1 + gradec1 + gradec2 + (1 + gradec1 + gradec2 | caseid)
    Df    AIC    BIC logLik  Chisq Chi Df Pr(>Chisq)
m15  6 130778 130825 -65383
m21 10 130311 130390 -65146 474.76      4  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The two-phase change model was significantly better than the one-phase change model, $\chi^2(4) = 474.76$, $p < .001$. From the output, the change in junior high school was steeper than the change in high school on average. The correlation between linear change in junior high school and linear change in high school was .134. The correlation between the math achievement score at Grade 7 and linear change at junior high school was .408. The correlation between math achievement score at Grade 7 and linear change at high school was .002. Readers may wonder whether the piecewise linear growth model or quadratic growth model fitted the data better. Readers are encouraged to take a look at AIC or BIC from both models.

The piecewise linear growth model can be visualized by the `ggplot2` package. I cannot find how to make a single meaningful plot with multiple lines. However, multiple plots of individual growths can be created:

```
ggplot(long2, aes(x = gradec, y = mathach, group = gradec > 2.5)) + facet_wrap(~caseid) +
geom_point() + geom_smooth(method = "lm", se=TRUE)
```



The trick is to create a group in the template such that Group 1 represents the observations that `gradec > 2.5` (high school) and Group 2 represents the observations that `gradec < 2.5` (junior high school).

Rearrange Data Structure

In dealing with longitudinal data, researchers may need to transform between *long* and *wide* formats frequently. In the situations which times are nested in cases, the *long* format has each row representing the observation at each time point. The *long* format is used in the `lme4` and `nlme` package, as well as most multilevel model programs:

Time	Case	DV	TIV	CIV
1	**1**	5	4	4
2	**1**	6	1	4
3	**1**	2	2	4
4	**1**	3	6	4
1	**2**	8	8	8
2	**2**	9	9	8
3	**2**	5	4	8
4	**2**	4	2	8
1	**3**	1	3	6
2	**3**	7	4	6
3	**3**	5	5	6
4	**3**	3	7	6

where DV is dependent variable, TIV is time-varying covariate, and CIV is time-invariant (case-level) covariate. On the other hand, in *wide* format, each row represents each case. If a variable is measured at different time points, the variable will be spanned in different columns to represent the variable values at different time points. The *wide* format is usually used in structural equation modeling, which also has features to deal with longitudinal model:

Case	DV1	DV2	DV3	DV4	TIV1	TIV2	TIV3	TIV4	CIV
1	5	6	2	3	4	1	2	6	4
2	8	9	5	4	8	9	4	2	8
3	1	7	5	3	3	4	5	7	6

In multiple imputation, sometimes, the *wide* format is more appropriate if the number of different time points is not many. In transforming between the data in long and wide formats, the `reshape` function will be used. I will illustrate this function using the `long` data set:

```
long <- read.csv("mathgrowth.csv", header = TRUE, na.strings="-999999")

head(long)
```

```
  caseid schoolid grade mathach parentpush peerpush likemath gender race
1      1      101     7   70.05          2        0        1      2    3
2      1      101     8   69.23          2        2        1      2    3
3      1      101     9   71.07          2        0        1      2    3
4      1      101    10   78.52          2        1        1      2    3
5      1      101    11   81.66          2        0        1      2    3
6      1      101    12   78.77          2        0        1      2    3
```

The `long` data set is in the long format, which grades are nested in students. Here is the information of the data set:

- Student ID: `caseid`
- Time variable: `grade`
- Student-level variables: `schoolid`, `likemath`, `gender`, `race`
- Time-varying variables: `mathach`, `parentpush`, `peerpush`

Before restructuring a data set, classifying variables in the data set into four types listed above is really helpful. Note that `schoolid` can be viewed as Level-3 ID. I will ignore the school ID by treating it as student-level variable for the sake of simplicity.

First, let's change the `long` data set from the long format to the wide format by the `reshape` function:

```
wide <- reshape(data = long, idvar = "caseid", timevar="grade", v.names=c("mathach",
"parentpush", "peerpush"), direction="wide")
```

- `data`: The target data set in the long format
- `idvar`: The ID of L2 units, which is `caseid`.
- `timevar`: Time variable, which is `grade`.
- `v.names`: Time-varying variables, which are `mathach`, `parentpush`, `peerpush`
- `direction`: The direction of restructuring. In this case, the change is to wide format.

All other variables that are not specified in `idvar`, `timevar`, and `v.names` will be treated as L2 variable (time-invariant variables), which are `schoolid`, `likemath`, `gender`, and `race`. The `head` function can be used to investigate the resulting data set:

```
head(wide)
```

```
   caseid schoolid likemath gender race mathach.7 parentpush.7 peerpush.7 mathach.8 parentpush.8 peerpush.8
1       1      101        1      2    3     70.05            2          0     69.23            2          2
7       2      101        3      2    3     59.36            2          2     65.20            0          1
13      3      101        3      1    3     68.47            0          0     69.31            2          0
19      4      101        0      1    1     61.66            0          0     66.08            0          1
25      5      101        3      1    3     69.87            2          3     65.75            2          3
31      6      101        2      1    3     64.22            2          0     63.88            2          0
   mathach.9 parentpush.9 peerpush.9 mathach.10 parentpush.10 peerpush.10 mathach.11 parentpush.11 peerpush.11
1      71.07            2          0      78.52             2           1      81.66             2           0
7      68.92            1          0      72.06             2           2      70.00             2           1
13        NA           NA         NA         NA             0          NA      80.57             0           0
19     71.02            1          0      70.00             1           1      82.12             2           0
25        NA           NA         NA      74.47            NA           1      76.04             1           0
31     59.37            2          0      61.90             2           0      78.31             2           0
   mathach.12 parentpush.12 peerpush.12
1       78.77             2           0
7       75.05             2           0
13         NA            NA          NA
19      90.01             1           0
25      80.02             0           1
31      79.07             2           0
```

The time-varying variables (`mathach`, `parentpush`, and `peerpush`) are spanned in the columns based on different grades.

Let's change the `wide` data set back to the long format. A little trick is needed. First, we need to create a vector of variable names that represent the same variable measured at different time. From the wide data set, `mathach.7`, `mathach.8`, `mathach.9`, `mathach.10`, `mathach.11`, and `mathach.12` are all measured math achievement. Thus, the vector name can be created:

```
mathach <- paste0("mathach.", 7:12)

mathach
```

```
[1] "mathach.7"  "mathach.8"  "mathach.9"  "mathach.10" "mathach.11" "mathach.12"
```

The `paste0` function is used to simply concatenate `"mathach."` with numbers 7 to 12. The vector names of `parentpush` and `peerpush` can be created as well:

```
parentpush <- paste0("parentpush.", 7:12)

peerpush <- paste0("peerpush.", 7:12)
```

Next, the vectors of names are combined into a list:

```
timevarying <- list(mathach, parentpush, peerpush)
```

Finally, the `reshape` function is used to transform the data in the wide format back into the long format:

```
long2 <- reshape(data = wide, idvar = "caseid", times = 7:12, timevar="grade", varying =
timevarying, v.names=c("mathach", "parentpush", "peerpush"), direction="long")
```

- `data`: The target data set in the wide format
- `idvar`: The ID of L2 units, which is `caseid`.
- `times`: The unit of time that each variable was measured in the specified vectors above, which is from Grade 7 to Grade 12
- `timevar`: The name of the time variable, which is `grade` (or any other names)
- `varying`: The list of vectors containing names of the same variable measured at different time points, which is the `timevarying` object created above

- `v.names`: The names of the time-varying variables in each element of the list, which are `mathach`, `parentpush`, and `peerpush` (or any other names).
- `direction`: The direction of restructuring. In this case, the change is to long format.

All other variables that are not specified in `idvar` and `varying` will be treated as L2 variables (time-invariant variables), which are `schoolid`, `likemath`, `gender`, and `race`. The `head` function can be used to investigate the resulting data set:

```
head(long2)
```

```
    caseid schoolid likemath gender race grade mathach parentpush peerpush
1.7      1      101        1      2    3     7   70.05          2        0
2.7      2      101        3      2    3     7   59.36          2        2
3.7      3      101        3      1    3     7   68.47          0        0
4.7      4      101        0      1    1     7   61.66          0        0
5.7      5      101        3      1    3     7   69.87          2        3
6.7      6      101        2      1    3     7   64.22          2        0
```

Users may check whether the `long` and `long2` data sets are the same. The easy way to check is to use the `summary` function on both data sets. The results should be equivalent.

Missing Data

Both `lme4` and `nlme` package simply uses listwise deletion when any observations are missing. Listwise deletion provides an accurate result in very restricted situation (missing completely at random) and usually provides lower power than more advanced techniques. In this section, I will illustrate how to use multiple imputation by the `mice` package (Van Buuren & Groothuis-Oudshoorn, 2011) to handle data with missing observations. Because the data with missing observations are imputed into multiple complete data sets, I will show how to analyze multiply-imputed data and pool the results from different analysis results. I assume that readers know the basic of missing data mechanisms and multiple imputation (in a single level) here. We will use the `long` dataset from the growth curve model section. Let's load the `mice` package and reload the `long` data set.

```
library(mice)

long <- read.csv("mathgrowth.csv", header = TRUE, na.strings="-999999")
```

If you run the `summary` function on the data set, you will notice many `NA` observations in the `mathach`, `parentpush`, `peerpush`, `likemath`, and `race` variables. The `mathach`, `parentpush`, `peerpush`, and `likemath` variables are assumed to be continuous variables. The `race` variable is a categorical variable with three categories (hispanic, black, and others). Here are the recommended steps for multilevel multiple imputation using `mice`:

1. Create dummy variables for categorical variables
2. Center your variables except group-mean centering
3. Create interactions (including cross-level interaction) or other transformations (e.g., quadratic terms)
4. Identify the L2 ID variable, used, and unused variables
5. Classify the used variables based on types of effects (fixed vs. random), types of measurement (continuous vs. dummy), and having missing observations (yes vs. no)

6. Specify the relations among variables in the imputation model (from Step 3)
7. Run multiple imputation with a specified number of imputations
8. Check for convergence of the imputation results
9. Analyze each imputed data
10. Pool the analysis results

Step 1: Dummy Variables

Because the `mice` package cannot impute the factor variable with more than two categories directly in the multilevel imputation, categorical variables are transformed as dummy variables. The `gender` and `race` variables are categorical variables at L2 so they are transformed to dummy variables:

```
long$gender <- long$gender == 2 # 1 = Male; 0 = Female

long$hispanic <- long$race == 1 # 1 = Hispanic

long$black <- long$race == 2 # 1 = Black
```

The `hispanic` and `black` variables are new variables in the `long` data set.

Step 2: Centering

The `grade` variable needs to be centered at Grade 7:

```
long$gradec <- long$grade - 7
```

Step 3: Interactions and Transformations

The interactions are created. In this case, I expect that different gender groups and different racial groups have different linear rate of change. Therefore, three interactions are created: `intgender` (`gradec*gender`), `inthisp` (`gradec*hispanic`), `intblack` (`gradec*black`).

```
long <- data.frame(long, intgender = long$gradec * long$gender, inthisp = long$gradec *
long$hispanic, intblack = long$gradec * long$black)
```

The `data.frame` function is used to bind three extra variables into the `long` data set and retains the data frame format. Note that, in the products of two variables, if a case has missing values in either of two variables, the interaction will be a missing value. The relationship between the interaction variable and the main effect variables must be retained (e.g., inthisp must be equal to the product of gradec and hispanic) after imputation. Therefore, users must specify the relationship between variables in the imputation model, which will be shown in Step 6.

Step 4: L2 ID, Used, and Unused Variables

L2 ID variable is the `caseid` variable. There are three variables that we will not use at all in the imputation models: `schoolid` (ignored for the sake of simplicity), `grade` (which was transformed to `gradec`), and `race` (which was transformed to `hispanic` and `black`). These three variables will be simply retained in the data set and have no role during the imputation process. Thus, we have three types of variables:

1. L2 ID: `caseid`
2. Unused variables: `schoolid`, `grade`, `race`

3. Used variables: `mathach`, `parentpush`, `peerpush`, `likemath`, `gender`, `hispanic`, `black`, `gradec`, `intgender`, `inthisp`, `intblack`

Let's make new objects as shortcuts of three types of variables:

```
l2id <- "caseid"

unused <- c("schoolid", "grade", "race")

used <- setdiff(colnames(long), c(l2id, unused))
```

The `setdiff` function is used to delete any elements of the vector in the first argument that are redundant with any elements in the second argument. I started with all variable names and delete L2 ID and unused variables. Hence, the remaining variables are the used variables. You may type them out manually.

From three groups of variables, we need to change the imputation model according to each type of variables. Initially, the imputation model template is created by the `mice` function:

```
ini <- mice(long, maxit = 0)
```

The first argument is the target data set. The `maxit` argument is the number of iterations. The argument is set to 0 so the multiple imputation has not been run yet. We simply want to create a template of the imputation model. Then, the prediction model, `pred`, and the method of imputations, `meth`, were extracted:

```
pred <- ini$pred

meth <- ini$meth
```

The prediction model can be investigated by simply typing `pred` in the R console. You will see the following matrix:

	caseid	schoolid	grade	mathach	parentpush	peerpush	likemath	gender	race	hispanic	black	gradec	intgender	inthisp	intblack
caseid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
schoolid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
grade	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mathach	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1
parentpush	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1
peerpush	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1
likemath	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1
gender	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
race	1	1	1	1	1	1	1	1	0	1	1	0	1	1	1
hispanic	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1
black	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
gradec	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
intgender	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
inthisp	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1
intblack	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0

Rows represent the predicted variable ands columns represent the predictors. We need to edit this matrix according to each type of variable. Here are the meanings of the values in each cell:

- `0` is to not use as a predictor
- `1` is to use as a fixed predictor (not random across L2 units)
- `2` is to use as a random predictor
- `-2` is the L2 ID

For example,

	caseid	schoolid	grade
mathach	-2	0	1

The missing values of the `mathach` variable is predicted by `grade` (as a fixed predictor) and not predicted by `schoolid` where and the `caseid` variable is the L2 ID.

In the prediction matrix, all rows and columns relating to unused variables must be 0 (**grey highlight**):

```
pred[unused, ] <- 0

pred[, unused] <- 0
```

In all rows of used variables, the column of L2 ID must be -2 (**orange highlight**):

```
pred[used, l2id] <- -2
```

All values in the row of L2 ID must be 0 (**green highlight**), which means having no predictors:

```
pred[l2id, ] <- 0
```

The resulting matrix will be

	caseid	schoolid	grade	mathach	parentpush	peerpush	likemath	gender	race	hispanic	black	gradec	intgender	inthisp	intblack
caseid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
schoolid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
grade	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mathach	-2	0	0	0	1	1	1	1	0	1	1	0	1	1	1
parentpush	-2	0	0	1	0	1	1	1	0	1	1	0	1	1	1
peerpush	-2	0	0	1	1	0	1	1	0	1	1	0	1	1	1
likemath	-2	0	0	1	1	1	0	1	0	1	1	0	1	1	1
gender	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
race	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hispanic	-2	0	0	1	1	1	1	1	0	0	1	0	1	1	1
black	-2	0	0	1	1	1	1	1	0	1	0	0	1	1	1
gradec	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
intgender	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
inthisp	-2	0	0	1	1	1	1	1	0	1	1	0	1	0	1
intblack	-2	0	0	1	1	1	1	1	0	1	1	0	1	1	0

The method of imputations can be investigated. You will find the following vector:

```
meth
```

```
    caseid   schoolid      grade    mathach parentpush   peerpush   likemath     gender       race   hispanic
        ""         ""         ""      "pmm"      "pmm"      "pmm"      "pmm"         ""      "pmm"   "logreg"
     black     gradec  intgender    inthisp   intblack
  "logreg"         ""         ""      "pmm"      "pmm"
```

The L2 ID and unused variables should not have any methods of imputations so they are specified as `""`.

```
meth[c(l2id, unused)] <- ""
```

Step 5: Types of Used Variables

The used variables are classified based on three dimensions: 1) types of effect, 2) types of measurement, and 3) having missing observations. Users can use the `summary` function on the target data set to see which variables have missing observations (`NA`). Users will notice that `gender`, `gradec`, and `intgender` do not have any missing observations. Here are the classifications of used variables based on three dimensions:

	Missing		No Missing	
	Continuous	Dummy	Continuous	Dummy
Fixed Effect (1)	`likemath, inthisp, intblack`	`hispanic, black`	`intgender`	`gender`
Random Effect (2)	`mathach, parentpush, peerpush`		`gradec`	

Note that all cross-level interactions must be treated as fixed effects. Let's create lists of variables with missing/no missing observations and fixed/random effects:

```
nomiss <- c("gender", "gradec", "intgender")

miss <- setdiff(used, nomiss)

random <- c("mathach", "parentpush", "peerpush", "gradec")

fixed <- setdiff(used, random)
```

In the prediction matrix, the rows of used variables without any missing observations must be 0 indicating that no imputation model is applied to those variables (**purple highlight**):

```
pred[nomiss, ] <- 0
```

The cells on the rows of used variables with missing observations and the columns of fixed effects are specified as 1 (**blue highlight**):

```
pred[miss, fixed] <- 1
```

The cells on the rows of used variables with missing observations and the columns of random effects are specified as 2 (**red highlight**):

```
pred[miss, random] <- 2
```

Finally, all variables cannot be predicted by themselves so all diagonal elements must be 0:

```
diag(pred) <- 0
```

The resulting prediction matrix will be

	caseid	schoolid	grade	mathach	parentpush	peerpush	likemath	gender	race	hispanic	black	gradec	intgender	inthisp	intblack
caseid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
schoolid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
grade	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mathach	-2	0	0	0	2	2	1	1	0	1	1	2	1	1	1
parentpush	-2	0	0	2	0	2	1	1	0	1	1	2	1	1	1
peerpush	-2	0	0	2	2	0	1	1	0	1	1	2	1	1	1
likemath	-2	0	0	2	2	2	0	1	0	1	1	2	1	1	1
gender	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
race	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hispanic	-2	0	0	2	2	2	1	1	0	0	1	2	1	1	1
black	-2	0	0	2	2	2	1	1	0	1	0	2	1	1	1
gradec	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
intgender	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
inthisp	-2	0	0	2	2	2	1	1	0	1	1	2	1	0	1
intblack	-2	0	0	2	2	2	1	1	0	1	1	2	1	1	0

Next, we will edit the methods of imputation. Here are the options for imputation methods:

- ***L1 continuous variables***: There are two options for L1 continuous variables: `"2l.pan"` as two-level regression with homoscedastic L1 errors and `"2l.norm"` as two-level regression with heteroscedastic L1 errors. I will use `"2l.pan"` here because it is faster (for illustration).
- ***L2 continuous variables***: `"2lonly.norm"` is to apply the two-level regression first, then find the average of imputed values within one L2 unit, and finally impute all missing observations in the L2 unit by the average.
- ***L2 dummy variables***: `"2lonly.pmm"` is to apply the two-level regression first, then find the average of imputed values within one L2 unit, find the closet observed values (0 and 1) to the average, and finally impute all missing observations in the L2 unit by the closet value. Loosely speaking, if the average is less than 0.5, 0 is imputed. Otherwise, 1 is imputed.[16]
- ***L1 dummy variables***: I have not found any methods in the `mice` package yet.

Therefore, we can adjust the methods of imputation accordingly:

```
meth[c("likemath", "inthisp", "intblack")] <- "2lonly.norm" # L2 continuous variables

meth[c("hispanic", "black")] <- "2lonly.pmm" # L2 dummy variables

meth[c("mathach", "parentpush", "peerpush")] <- "2l.pan" # L1 continuous variables
```

For the used variables without missing observations, the methods of imputation must be specified as `""`.

```
meth[nomiss] <- ""
```

Step 6: Remain Interactions and Transformations in Imputation Model

The relations between interactions and main effects must be specified here. The interaction variables that do not have missing observations do not need to specify anything here—you may specify it but the specification is not necessary. Thus, the `intgender` variable is left alone. Thus, the relations between `inthisp` and `intblack` variables and their main effects must be specified. The relations are specified in the method of imputation:

```
meth["inthisp"] <- "~I(gradec*hispanic)"

meth["intblack"] <- "~I(gradec*black)"
```

The resulting methods of imputation would be

```
meth
```

```
          caseid           schoolid              grade             mathach          parentpush
              ""                 ""                 ""            "2l.pan"            "2l.pan"
        peerpush           likemath             gender                race            hispanic
        "2l.pan"      "2lonly.norm"                 ""                  ""        "2lonly.pmm"
           black             gradec          intgender             inthisp            intblack
    "2lonly.pmm"                 ""                 "" "~I(gradec*hispanic)"  "~I(gradec*black)"
```

The `~I()` is used to crop the transformation from other variables in the data frame, which can be any relations beside interactions (e.g., `~I(gradec^2)`). Because `hispanic` and `black` are used to create `inthisp` and `intblack`, `hispanic` and `black` cannot be predicted by `inthisp` and `intblack`. The prediction matrix must be changed accordingly:

---

[16] Note that this method is not similar to Graham's (2009) suggestion that the resulting imputed values for dummy varaibles should not be rounded. If you follow Graham's suggestion, `"2lonly.norm"` should be used.

```
pred[c("hispanic", "black"), c("inthisp", "intblack")] <- 0
```

The resulting prediction matrix would be (see changes in the texts with yellow highlights)

	caseid	schoolid	grade	mathach	parentpush	peerpush	likemath	gender	race	hispanic	black	gradec	intgender	inthisp	intblack
caseid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
schoolid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
grade	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mathach	-2	0	0	0	2	2	1	1	0	1	1	2	1	1	1
parentpush	-2	0	0	2	0	2	1	1	0	1	1	2	1	1	1
peerpush	-2	0	0	2	2	0	1	1	0	1	1	2	1	1	1
likemath	-2	0	0	2	2	2	0	1	0	1	1	2	1	1	1
gender	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
race	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hispanic	-2	0	0	2	2	2	1	1	0	0	1	2	1	0	0
black	-2	0	0	2	2	2	1	1	0	1	0	2	1	0	0
gradec	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
intgender	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
inthisp	-2	0	0	2	2	2	1	1	0	1	1	2	1	0	1
intblack	-2	0	0	2	2	2	1	1	0	1	1	2	1	1	0

Step 7: Start Multiple Imputation

The `mice` function is used again but the `maxit` argument is not fixed as 0:

```
imp <- mice(long, m = 5, maxit = 10, meth = meth, pred = pred, seed = 123321)
```

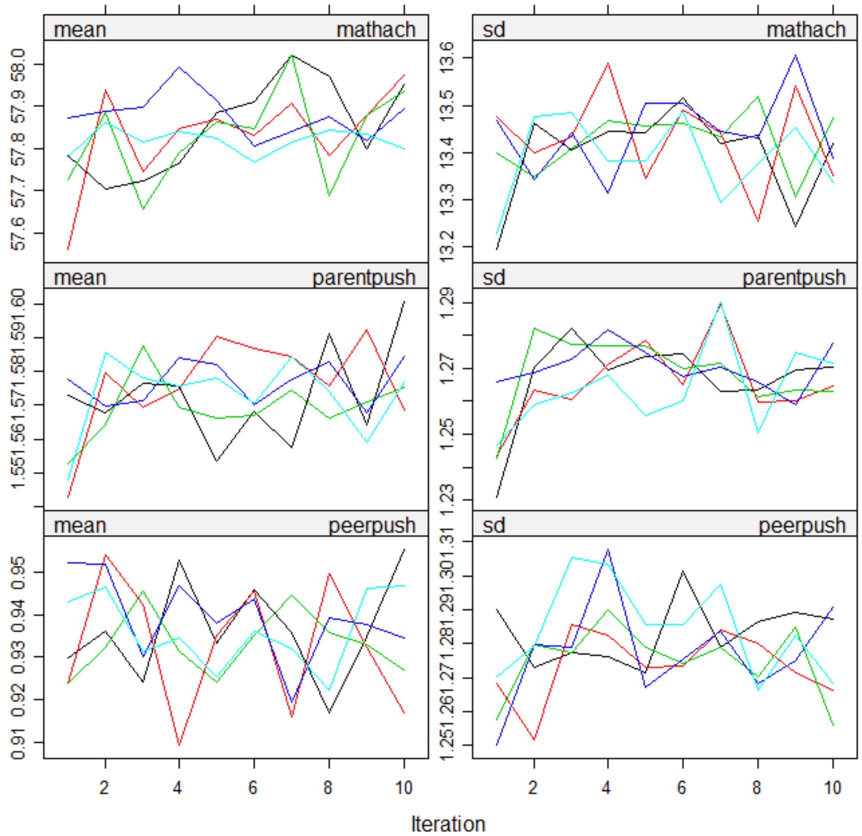
Disclaimer: This command will take a long time so that you may finish having a meal.

The first argument is the target data set. The `m` argument is the number of imputations. I use 5 here to save time. Users are encouraged to run more than 5 imputations. The `maxit` argument is the number of iterations. The number of iterations should be high enough so that the imputation is convergent. I will show you how to examine the convergence status in Step 8. The `meth` argument is the method of imputations. The `pred` argument is the prediction matrix. The `seed` argument is the random seed number. Researchers are expected to get the same results if the same seed number is applied (given that other arguments remain the same).

Step 8: Checking for Convergence

The easy way to investigate the convergence of an imputation model is to plot a graph:

```
plot(imp, c("mathach", "parentpush", "peerpush"))
```



The first argument is the resulting imputation. The second argument is the target variables to be investigated. The graphs will show the mean and standard deviations of each variables where each line indicates different imputations. In a convergent imputation, different lines should be freely intermingled with each other, without showing any definite trends. If lines have not crossed between each other for a multiple times yet, a higher number of iterations are needed. For these three variables, the convergence status should be good. Users are encouraged to check the plots of the `likemath`, `hispanic`, and `black` variables.

Step 9: Analyze Each Imputed Data

Each imputed data set can be analyzed by a target model. The `mice` package provides the `with` function to help analyze each imputed data set. I analyze the imputed data sets by two models: linear model with fixed slope and linear model with random slopes:

```
fit1 <- with(imp, lmer(mathach ~ gradec + (1|caseid), REML=FALSE))

fit2 <- with(imp, lmer(mathach ~ gradec + (1 + gradec|caseid), REML=FALSE))
```

The first argument is the resulting imputation model. The second argument is a target analysis model. In this case, the `lmer` command (or `lme` command from the `nlme` package) is written as if we analyze a data set. The only difference is to not specify the `data` argument.

Step 10: Pooling Results

For the fixed effects, researchers can use the `pool` function to combine analysis results from multiply imputed data sets:

```
out1 <- pool(fit1)

summary(out1)
```

```
                  est         se        t         df     Pr(>|t|)     lo 95     hi 95 nmis       fmi   lambda
(Intercept) 50.806450 0.17115114 296.8514 119.229666 0.000000e+00 50.467560 51.145339   NA 0.1960859 0.182713
gradec       3.384409 0.03107243 108.9200   8.008251 5.484502e-14  3.312768  3.456049    0 0.7597566 0.706418
```

```
> out2 <- pool(fit2)

> summary(out2)
```

```
                  est         se         t       df Pr(>|t|)     lo 95     hi 95 nmis       fmi    lambda
(Intercept) 50.806450 0.14802933 343.21880 66.79988        0 50.510966 51.101934   NA 0.2659041 0.2442494
gradec       3.384409 0.03574667  94.67759 14.02259        0  3.307751  3.461066    0 0.5885336 0.5337539
```

The output of the `pool` function can be investigated by the `summary` function. In the output, `fmi` is the fraction of missing information and `lambda` is the proportion of variation attributed to the missing data. The fraction of missing information tends to be higher if the proportion of missing observations is higher.

Because two models are nested models, the deviance test can be used by the `anovaMI` function from the `rockchalkMultilevel` package:[17]

```
anovaMI(fit1, fit2)
```

```
      F     df1     df2     p.F
658.325   2.000  11.460   0.000
```

Group-Mean Centering

The method of creating new variables to represent group means or group-mean centered variables is not easy for multiply-imputed data. Thus, we need to create group means or group-mean centered variables within a formula using the `I()` command. For example, the math achievement is predicted by parent encouragement, which is group-mean centered. The group means of parent encouragement are added back and centered at the grand mean. Researchers can analyze each imputed data by

```
fit3 <- with(imp, lmer(mathach ~ I(parentpush - ave(parentpush, caseid)) + I(ave(parentpush,
caseid) - mean(parentpush)) + (1|caseid), REML=FALSE))

out3 <- pool(fit3)

summary(out3)
```

```
> summary(out3)
                                                  est         se         t          df     Pr(>|t|)
(Intercept)                                 59.267471 0.14816171 400.01880 1166.452443 0.000000e+00
I(parentpush - ave(parentpush, caseid))     -3.079125 0.08673658 -35.49973    9.218892 3.546763e-11
I(ave(parentpush, caseid) - mean(parentpush))  3.764696 0.19319455  19.48655  361.647246 0.000000e+00
                                                lo 95     hi 95 nmis        fmi     lambda
(Intercept)                                 58.976778 59.558164   NA 0.05893957 0.05732741
I(parentpush - ave(parentpush, caseid))     -3.274629 -2.883621   NA 0.71431792 0.65840538
I(ave(parentpush, caseid) - mean(parentpush))  3.384770  4.144622   NA 0.10936386 0.10445200
```

---

[17] The anovaMI function computes the deviance tests for each imputed data and pool the chi-square values by the Li, Meng, Raghunathan, & Rubin (1991) method. I have not tested the performance of this method yet.

The `I(parentpush - ave(parentpush, caseid))` is the group-mean centered parent encouragement (L1 effect) and the `I(ave(parentpush, caseid) - mean(parentpush))` is the group means of parent encouragement (L2 effect) that is centered at the grand mean. Interestingly, the measurement-level effect was significantly negative but the student-level effect was significantly positive.

As the last note on multiple imputation, if the data set is a longitudinal data that the time variable is constant across cases. For example, all students are measured at Grade 7 to Grade 12. Researchers may change the data into the wide format and use multiple imputation on the data with wide format. The benefits of using multiple imputation on the data with wide format is that 1) the changes are not restricted to linear change (or any specified types of changes) and 2) more appropriate options are available for different types of variables (e.g., ordinal logistic regression for `parentpush` and `peerpush`, which are measured in the Likert scale).

Alternative Statistical Tests

Multiparameter Test

Researchers may have a hypothesis that could be tested by the addition or subtraction among regression coefficients of the fixed effects. I will provide three examples of using the multiparameter test.

Example 1: The Difference in Linear Rates of Change

In the piecewise linear growth model, Model 21, researchers may wish to test whether the linear rates of change at junior high school and high school are different. In this case, the hypothesis can be written as

$$H_0: \gamma_{10} - \gamma_{20} = 0 \quad or \quad \gamma_{10} = \gamma_{20}$$

From the `summary` function of the multilevel output, the coefficients of fixed effects are arranged as γ_{00}, γ_{10}, and γ_{20}. To test a multiparameter test, users need to build a contrast matrix that contains the coefficients of each parameter in the contrast. To find the coefficients, users try to make 0 on either side of the equation, which is $\gamma_{10} - \gamma_{20} = 0$. Then, the coefficients of γ_{00}, γ_{10}, and γ_{20} are 0, 1, and -1, respectively. Users need to make the following matrix:

	γ_{00} (Intercept)	γ_{10} (gradec1)	γ_{20} (gradec2)
$\gamma_{10} - \gamma_{20} = 0$	0	1	-1

The rows of contrast matrix represents each contrast (I will show multiple contrasts later) and the columns of contrast matrix represents each regression coefficient listed in the order of the fixed effects from the `summary` function. The matrix can be made:

```
ctr <- matrix(c(0, 1, -1), 1)
```

We will use the `multcomp` package (Hothorn, Bretz, & Westfall, 2008) to test the contrast. The `glht` function will be used to test the contrast:[18]

[18] If users open the `lme4` package and `nlme` package at the same time, researchers will have a problem in running the `glht` function. Researchers need to detach the `nlme` package from the R workspace by typing `detach(package:nlme)`

```
library(multcomp)

phasediff <- glht(m21, linfct = ctr)
```

In the `glht` function, the first argument is the result from the `lmer` function. The `linfct` argument is the contrast matrix. Then, the result can be investigated by the `summary` function:

```
summary(phasediff)
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: lmer(formula = mathach ~ 1 + gradec1 + gradec2 + (1 + gradec1 +
    gradec2 | caseid), data = long, REML = FALSE)

Linear Hypotheses:
       Estimate Std. Error z value Pr(>|z|)
1 == 0  1.10704    0.09684   11.43   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

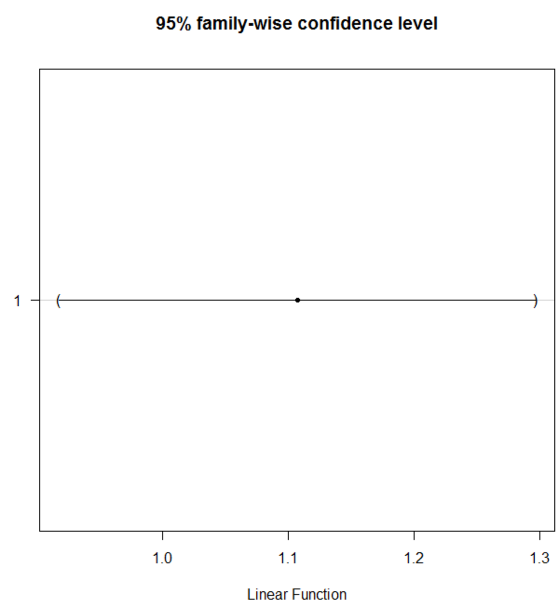
The confidence interval of the contrast can be calculated by the `confint` function:

```
confint(phasediff)
```

```
         Simultaneous Confidence Intervals

Fit: lmer(formula = mathach ~ 1 + gradec1 + gradec2 + (1 + gradec1 +
    gradec2 | caseid), data = long, REML = FALSE)

Quantile = 1.96
95% family-wise confidence level


Linear Hypotheses:
       Estimate lwr    upr
1 == 0 1.1070   0.9172 1.2968
```

The confidence interval can be ploted by the following script:

```
plot(confint(phasediff))
```



From the result, the rate of change at the junior high school was significantly steeper than the rate of change at high school. Instead of specifying the contrast matrix, users may specify the syntax, `"gradec1 - gradec2 = 0"`, instead:

```
phasediff2 <- glht(m21, linfct = "gradec1 - gradec2 = 0")

summary(phasediff2)
```

Example 2: The Influence of Significant Others

In the linear growth model with time-varying covariates, Model 18, researchers may wish to investigate whether 1) the average of the parents influence and peers influence is different from 0 and 2) the influences from parents and peer are different. The hypotheses can be written as

$$H_0: \frac{\gamma_{20} + \gamma_{30}}{2} = 0 \quad or \quad 0.5\gamma_{20} + 0.5\gamma_{30} = 0$$

$$H_0: \gamma_{20} - \gamma_{30} = 0 \quad or \quad \gamma_{20} = \gamma_{30}$$

The contrast matrix will have two rows representing two contrasts:

	γ_{00} (Intercept)	γ_{10} (gradec)	γ_{20} (parentpushC)	γ_{30} (peerpushC)
$0.5\gamma_{20} + 0.5\gamma_{30} = 0$	0	0	0.5	0.5
$\gamma_{20} - \gamma_{30} = 0$	0	0	1	-1

The matrix can be made:

```
ctr1 <- c(0, 0, 1/2, 1/2)

ctr2 <- c(0, 0, 1, -1)

ctr <- rbind(ctr1, ctr2)
```

The `rbind` function is used to concatenate vectors as rows of a matrix. These contrasts can be simultaneously tested:

```
pushctr <- glht(m18, linfct = ctr)

summary(pushctr)
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: lmer(formula = mathach ~ 1 + gradec + parentpushC + peerpushC +
    (1 + gradec | caseid), data = long, REML = FALSE)

Linear Hypotheses:
           Estimate Std. Error z value Pr(>|z|)
ctr1 == 0  0.12089    0.03327   3.633 0.000559 ***
ctr2 == 0  0.49668    0.07668   6.477 1.87e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

The *p*-values have been controlled for familywise error rates. By default, the single-step approach is used for the adjustement. The adjusted *p* values is computed from the joint normal distribution of the *z* statistics such that the *p* value represents the probability of getting at least one significant result by chance if all *z* values are the same in all contrasts.[19]

[19] The single-step method is similar to Tukey method in pairwise comparisons in Analysis of Variance.

Users may change the method of controlling familywise error rates by specifying the test argument in the `summary` function:

```
summary(pushctr, test = adjusted(type = "bonferroni"))
```

In this case, the Bonferroni method is used. Researchers can investigate the confidence intervals by the `confint` function. The resulting confidence intervals are simultaneous confidence intervals, which is the probability over repeated sampling that all confidence intervals will bracket the population values simultanouesly. The confidence intervals can be plotted by the same method shown in the previous example.

The syntax format can be used to specify the contrasts. For the `glht` function, specify different contrasts in the `linfct` argument as different elements in a vector:

```
pushctr2 <- glht(m18, linfct = c("0.5*parentpushC + 0.5*peerpushC = 0", "parentpushC - peerpushC
= 0"))

summary(pushctr2, test = adjusted(type = "bonferroni"))
```

Example 3: The Difference between Types of Schools

Researchers may have a hypothesis about the direction of the difference in language scores across types of schools, controlling for verbal IQ score. Model 3 is used here. The hypothesis is that the schools Type 1, 2, and 3 have different language scores average from the schools Type 4 and 5. The hypothesis can be written as

$$H_0: \frac{\mu_1 + \mu_2 + \mu_3}{3} - \frac{\mu_4 + \mu_5}{2} = 0 \quad or \quad \frac{\mu_1 + \mu_2 + \mu_3}{3} = \frac{\mu_4 + \mu_5}{2}$$

where μ_j represents the means of schools type *j* (*j* = 1, 2, 3, 4, or 5).

Note that the means are not the regression coefficients. We cannot write a contrast directly from the hypothesis of the means. We know the relations between the means and regression coefficients, however, such that $\mu_1 = \gamma_{00}$, $\mu_2 = \gamma_{00} + \gamma_{01}$, $\mu_3 = \gamma_{00} + \gamma_{02}$, $\mu_4 = \gamma_{00} + \gamma_{03}$, and $\mu_5 = \gamma_{00} + \gamma_{04}$. Thus, the hypothesis can be rewritten as

$$H_0: \frac{(\gamma_{00}) + (\gamma_{00} + \gamma_{01}) + (\gamma_{00} + \gamma_{02})}{3} - \frac{(\gamma_{00} + \gamma_{03}) + (\gamma_{00} + \gamma_{04})}{2} = 0$$

or

$$H_0: \frac{1}{3}\gamma_{01} + \frac{1}{3}\gamma_{02} - \frac{1}{2}\gamma_{03} - \frac{1}{2}\gamma_{04} = 0$$

Therefore, the contrast matrix will be

	γ_{00} (Intercept)	γ_{10} (IQ_verb)	γ_{01} (denomina2)	γ_{02} (denomina3)	γ_{03} (denomina4)	γ_{04} (denomina5)
Contrast	0	0	1/3	1/3	-1/2	-1/2

This contrast can be tested by the `glht` function:

```
ctr <- matrix(c(0, 0, 1/3, 1/3, -1/2, -1/2), 1)

typediff <- glht(m3, linfct = ctr)

summary(typediff)
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: lmer(formula = langPOST ~ 1 + IQ_verb + denomina + (1 | schoolnr),
    data = dat, REML = FALSE)

Linear Hypotheses:
       Estimate Std. Error z value Pr(>|z|)
1 == 0  -0.6451     0.7536  -0.856    0.392
(Adjusted p values reported -- single-step method)
```

The contrast was not significant. Users are encouraged to specify this contrast by the syntax approach.

Multivariate Wald Test

Researchers can test different contrasts simultaneously by multivariate Wald test in the same fashion as in *F*-test in ANOVA. For example, two contrasts in the example of parents and peers influences can be tested simultaneously. The restriction in multivariate Wald test is that contrasts need to be linearly independent (not require in the multiparameter test). Users can use the `rankMatrix` function to check whether the contrasts are linearly independent. If the rank is equal to the number of contrasts, the specified contrasts are linearly independent. Let's try to check the rank of the contrast specified in Example 2:

```
ctr1 <- c(0, 0, 1/2, 1/2)

ctr2 <- c(0, 0, 1, -1)

ctr <- rbind(ctr1, ctr2)

rankMatrix(ctr)
```

```
[1] 2
attr(,"method")
[1] "tolNorm2"
attr(,"useGrad")
[1] FALSE
attr(,"tol")
[1] 1.256074e-15
```

The rank of this matrix is 2, which is equal to the number of contrasts. Therefore, the contrasts are linearly independent, which is good. The multivariate Wald test can be done by the `wald.mlm` function in the `rockchalkMultilevel` package:

```
wald.mlm(m18, ctr)
```

```
       chisq           df            p
4.909711e+01 2.000000e+00 2.181214e-11
```

The first argument is a result from the `lmer` function. The second argument is the contrast matrix. The simultaneous test was significant, $\chi^2(2) = 49.10$, $p < .001$. Loosely speaking, at least one contrast was significant.

Three-Level Model

In this section, we will discuss how to analyze data with three levels of nesting. For example, students are nested in classrooms and classrooms are nested in schools. We will use the long data set that we have used in the growth curve model section. We used only two-level model (measurements are nested in students) and ignored the school level. The school level will be accounted for here.

Data Structure for Three-Level Model

The data must be in the long format such that rows represent L1 units. Two variables are needed for L2 ID and L3 ID. For the `lme4` pacakge, the L2 ID from different L3 units must be listed in different numbers. For example, the following data are not appropriate:

L1ID	L2ID	L3ID	DV
1	1	1	5
2	1	1	7
3	2	1	8
4	2	1	9
5	1	2	5
6	1	2	9
7	2	2	7
8	2	2	8
9	1	3	4
10	1	3	6
11	2	3	7
12	2	3	8

Notice that `1` and `2` are used to represent different L2 units within all three L3 units. The `lme4` package will assume that the rows coded as 1 (in L2ID) from the first, second, and third L3 units are similar (which will be a cross-classified model). The L2 units must be transformed to have different values across different L3 units:

L1ID	L2ID	L3ID	DV
1	1	1	5
2	1	1	7
3	2	1	8
4	2	1	9
5	3	2	5
6	3	2	9
7	4	2	7
8	4	2	8
9	5	3	4
10	5	3	6
11	6	3	7
12	6	3	8

For example, we will use the `posaffect` data set. Twenty participants answered the same positive affect scale five times a day for 10 days. Thus, measurements (L1) are nested in days (L2) and days are nested in participants (L3). The data set can be imported:

```
posaffect <- read.csv("posaffect.csv")
```

Let's check Rows 1-10 and 51-60 of the data:

```
posaffect[c(1:10, 51:60), ]
```

```
   id day measure posaffect
1   1   1       1        53
2   1   1       2        62
3   1   1       3        46
4   1   1       4        57
5   1   1       5        53
6   1   2       1        56
7   1   2       2        68
8   1   2       3        57
9   1   2       4        58
10  1   2       5        67
51  2   1       1        50
52  2   1       2        43
53  2   1       3        38
54  2   1       4        50
55  2   1       5        58
56  2   2       1        53
57  2   2       2        45
58  2   2       3        48
59  2   2       4        39
60  2   2       5        66
```

The id variable is L3 ID (participants). The day variable is L2 ID. The measure variable is L1 ID. Notice that at different values of the id variable, the values of the day variable (L2 ID) are duplicated. The easy method to create different L2 ID is to use the paste function to concatenate values of L3 ID and L2 ID (separated by a space):

```
posaffect$l2unit <- paste(posaffect$id, posaffect$day)

posaffect[c(1:10, 51:60), ]
```

```
   id day measure posaffect l2unit
1   1   1       1        53    1 1
2   1   1       2        62    1 1
3   1   1       3        46    1 1
4   1   1       4        57    1 1
5   1   1       5        53    1 1
6   1   2       1        56    1 2
7   1   2       2        68    1 2
8   1   2       3        57    1 2
9   1   2       4        58    1 2
10  1   2       5        67    1 2
51  2   1       1        50    2 1
52  2   1       2        43    2 1
53  2   1       3        38    2 1
54  2   1       4        50    2 1
55  2   1       5        58    2 1
56  2   2       1        53    2 2
57  2   2       2        45    2 2
58  2   2       3        48    2 2
59  2   2       4        39    2 2
60  2   2       5        66    2 2
```

Notice that Day 1 for Participant 1 is "1 1" in the l2unit variable and Day 1 for Participant 2 is "2 1" in the l2unit variable. The l2unit variable will be used to represent L2 ID.

Model 22: Three-Level Null Model

The long data set is used here. We will investigate the math achievement scores (mathach) based on three levels: measurements, students (caseid), and schools (schoolid). No predictors (including the grade variable) will be added in this model. The three-level null model would be

L1	$Y_{ijk} = \pi_{0jk} + r_{ijk}$	$r_{ijk} \sim N(0, \sigma^2)$
L2	$\pi_{0jk} = \beta_{00k} + e_{0jk}$	$e_{0jk} \sim N(0, \tau_{00}^{(2)})$
L3	$\beta_{00k} = \gamma_{000} + u_{00k}$	$u_{00k} \sim N(0, \tau_{00}^{(3)})$

These notations represent

- Y_{ijk} = The math achievement score of Measurement *i* in Student *j* in School *k*
- π_{0jk} = The average of math achievement score across measurements within Student *j* in School *k*
- β_{00k} = The average of math achievement score across students within School *k*
- γ_{000} = The average of math achievement scores across schools
- r_{ijk} = The deviation of the math achievement score of Measurement *i* in Student *j* in School *k* from the Student *j* in School *k* average
- e_{0jk} = The deviation of the math achievement score of Student *j* in School *k* from the School *k* average
- u_{00k} = The deviation of the math achievement score of School *k* from the grand mean
- σ^2 = The language score variance within participants (L1 variance)
- $\tau_{00}^{(2)}$ = The language score variance across students but within schools (L2 variance)
- $\tau_{00}^{(3)}$ = The language score variance across schools (L3 variance)

Let's load the data in the workspace:

```
long <- read.csv("mathgrowthclass.csv", header = TRUE, na.strings="-999999")
```

I use a different file here. This data set is not different from `"mathgrowth.csv"` (that we have used before) except it contains an additional L3 predictor.

Next, the model with linear trajectory can be run by the `lmer` function:

```
m22 <- lmer(mathach ~ 1 + (1|caseid) + (1|schoolid), data=long, REML=FALSE)

summary(m22)
```

```
Linear mixed model fit by maximum likelihood
Formula: mathach ~ 1 + (1 | caseid) + (1 | schoolid)
   Data: long
    AIC    BIC logLik deviance REMLdev
 142040 142071 -71016   142032  142031
Random effects:
 Groups   Name        Variance Std.Dev.
 caseid   (Intercept) 110.289  10.5019
 schoolid (Intercept)  34.839   5.9024
 Residual              55.671   7.4613
Number of obs: 19041, groups: caseid, 5858; schoolid, 95

Fixed effects:
            Estimate Std. Error t value
(Intercept)  60.0687     0.6257      96
```

There are two random effects specification: `(1|caseid)` and `(1|schoolid)`.[20] These notations mean that intercepts (`1`) are random across both participants and schools. The mapping from the formula and reduced-form equation would be

[20] The alternative method to specify three-level model by the `lme4` package is

```
m22 <- lmer(mathach ~ 1 + (1|schoolid/caseid), data=long, REML=FALSE)
```

Note that the L3 ID must be listed before the forward slash. This method is not convenient when researchers have random L2 predictors across L3 units. Thus, I will illustrate the specification using parentheses in the following examples.

```
mathach ~ 1 + (1|caseid) + (1|schoolid)
```

$$Y_{ij} = \underbrace{\gamma_{00}(1)}_{\text{Fixed Effect}} + \underbrace{u_{00k}(1) + e_{0jk}(1) + r_{ijk}}_{\text{Random Effect}}$$

The intraclass correlations (the proportion of variances explained at each level) can be computed by the steps similar to those in Model 0:

1. Save the summary of the multilevel output.

```
out22 <- summary(m22)
```

2. Put `@REmat` after the summary output to get the random effect matrix

```
ranef22 <- out22@REmat

ranef22
```

```
  Groups     Name          Variance  Std.Dev.
 "caseid"   "(Intercept)" "110.289" "10.5019"
 "schoolid" "(Intercept)" " 34.839" " 5.9024"
 "Residual" ""            " 55.671" " 7.4613"
```

3. Extract appropriate values for $\tau_{00}^{(3)}$ (`tau3`), $\tau_{00}^{(2)}$ (`tau2`), and σ^2 (`sigma2`). Use the `as.numeric` function to change the string format to number:

```
tau3 <- as.numeric(ranef22[1, 3])

tau2 <- as.numeric(ranef22[2, 3])

sigma2 <- as.numeric(ranef22[3, 3])
```

4. Compute intraclass correlation at the school level, $\rho_3 = \tau_{00}^{(3)}/(\tau_{00}^{(3)} + \tau_{00}^{(2)} + \sigma^2)$:

```
icc3 <- tau3/(tau3 + tau2 + sigma2)

icc3
```

```
[1] 0.5492507
```

5. Compute intraclass correlation at participant level, $\rho_2 = \tau_{00}^{(2)}/(\tau_{00}^{(3)} + \tau_{00}^{(2)} + \sigma^2)$:

```
icc2 <- tau2/(tau3 + tau2 + sigma2)

icc2
```

```
[1] 0.1735019
```

Readers may try to run the null model of the positive affect from the `posaffect` data set. Then, compare the results when the `day` variable and the `l2id` variable are used for L2 ID. Note that the correct model should have the `l2id` variable as L2 ID. You will see that two models provide totally different values.

Model 23: Three-Level Linear Trajectory

In this model, the change of math achievement scores (`mathach`) across grade (`grade`) is modeled. Measurements are nested in students (`caseid`) and students are nested in schools (`schoolid`). This model is similar to Model 15 but the `schoolid` variable is accounted for. First, the `grade` variable is centered at Grade 7 to make the intercepts meaningful:

```
long$gradec <- long$grade - 7
```

The intercepts and slopes (linear change) are random across students and schools. The three-level linear growth model would be

L1	$Y_{ijk} = \pi_{0jk} + \pi_{1jk}(t_{ijk} - 7) + r_{ijk}$	$r_{ijk} \sim N(0, \sigma^2)$
L2	$\pi_{0jk} = \beta_{00k} + e_{0jk}$ $\pi_{1jk} = \beta_{10k} + e_{1jk}$	$\begin{bmatrix} e_{0jk} \\ e_{1jk} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00}^{(2)} & \\ \tau_{10}^{(2)} & \tau_{11}^{(2)} \end{bmatrix}\right)$
L3	$\beta_{00k} = \gamma_{000} + u_{00k}$ $\beta_{10k} = \gamma_{100} + u_{10k}$	$\begin{bmatrix} u_{00k} \\ u_{10k} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00}^{(3)} & \\ \tau_{10}^{(3)} & \tau_{11}^{(3)} \end{bmatrix}\right)$

These notations should represent

- Y_{ijk} = The math achievement score of Measurement *i* in Student *j* in School *k*
- t_{ijk} = The grade that the Measurement *i* in Student *j* in School *k* was observed
- π_{0jk} = The math achievement score of Student *j* in School *k* at Grade 7
- π_{1jk} = The expected change in math achievement score when grade increases by 1 for Student *j* in School *k*, which is the rate of change for Student *j* in School *k*
- β_{00k} = The average of math achievement scores in Grade 7 across students within School *k*
- β_{10k} = The average rate of change in math achievement scores across students within School *k*
- γ_{000} = The average of math achievement scores in Grade 7 across schools
- γ_{100} = The average rate of change in math achievement scores across schools
- r_{ijk} = The difference between the actual math achievement score of Measurement *i* in Student *j* in School *k* and the expected score of Student *j* in School *k* at a given grade level
- e_{0jk} = The deviation of the actual math achievement score of Student *j* in School *k* at Grade 7 from the average math achievement score at Grade 7 across students within School *k*
- e_{1jk} = The deviation of the rate of change of Student *j* in School *k* from the average rate of change across students within School *k*
- u_{00k} = The deviation of the actual math achievement score of School *k* at Grade 7 from the average math achievement score at Grade 7 across schools
- u_{10k} = The deviation of the rate of change of School *k* from the average rate of change across schools
- σ^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade
- $\tau_{00}^{(2)}$ = The variance of math achievement scores at Grade 7 within the student level (partialled out the school variances)
- $\tau_{11}^{(2)}$ = The variance of the rate of change in math achievement score within the student level (partialled out the school variances)
- $\tau_{10}^{(2)}$ = The covariance between the math achievement score at Grade 7 (initial status) and the rate of change within the student level

- $\rho_{10}^{(2)} = \tau_{10}^{(2)} / \sqrt{\tau_{00}^{(2)} \tau_{11}^{(2)}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)
- $\tau_{00}^{(3)}$ = The variance of math achievement scores at Grade 7 across schools
- $\tau_{11}^{(3)}$ = The variance of the rate of change in math achievement score across schools
- $\tau_{10}^{(3)}$ = The covariance between the math achievement score at Grade 7 (initial status) and the rate of change at the school level
- $\rho_{10}^{(3)} = \tau_{10}^{(3)} / \sqrt{\tau_{00}^{(3)} \tau_{11}^{(3)}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)

Next, the model with linear trajectory can be run by the `lmer` function:

```
m23 <- lmer(mathach ~ 1 + gradec + (1 + gradec|caseid) + (1 + gradec|schoolid), data=long,
REML=FALSE)

summary(m23)
```

```
Linear mixed model fit by maximum likelihood
Formula: mathach ~ 1 + gradec + (1 + gradec | caseid) + (1 + gradec |      schoolid)
   Data: long
    AIC    BIC logLik deviance REMLdev
 129817 129888 -64899   129799  129801
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 caseid   (Intercept) 72.67928 8.52521
          gradec       2.11129 1.45303  0.336
 schoolid (Intercept) 25.96207 5.09530
          gradec       0.52503 0.72459  -0.198
 Residual             20.36538 4.51280
Number of obs: 19041, groups: caseid, 5858; schoolid, 95

Fixed effects:
            Estimate Std. Error t value
(Intercept) 51.06746    0.55463   92.07
gradec       3.33156    0.08828   37.74

Correlation of Fixed Effects:
       (Intr)
gradec -0.258
```

The `gradec` variable is included in both parentheses to represent the random effects at both student and school levels. The mapping from the formula and reduced-form equation would be

mathach ~ 1 + gradec + (1 + gradec|caseid) + (1 + gradec| schoolid)
$Y_{ij} = \gamma_{000}(1) + \gamma_{100}(t_{ij} - 7) + u_{00k}(1) + u_{10k}(t_{ij} - 7) + e_{0jk}(1) + e_{1jk}(t_{ij} - 7) + r_{ijk}$ Fixed Effect + Random Effect

The linear trajectory was significant, $z = 37.74$, $p < .001$, such that, when the grade increases by 1, the math achievement score increases by 3.33 points on average. Interestingly, the correlations between the initial status (math achievement at Grade 7) and rate of change were positive (.34) at student level but negative (-.20) at school level. Within a school, if students had higher initial status, students tended to have higher growth. Across schools, however, schools with a higher initial status tended to have slower growth.

Readers are encouraged to compare the result with Model 15. Notice the change in the parameter estimate and *t* value of the fixed effects and the variances of random effects at the student level.

Model 24: Time-Invariant Covariate in Three-Level Model

In this model, the linear change of math achievement scores (`mathach`) across grade (`grade`) is predicted by gender (time-invariant covariate), which is similar to Model 17. However, the school level is accounted for here. In addition, the gender differences in initial status (math achievement score at Grade 7) and the gender differences in linear trajectory are random across schools. The three-level growth curve model with L2 covariate would be

L1 $$Y_{ijk} = \pi_{0jk} + \pi_{1jk}(t_{ijk} - 7) + r_{ijk} \qquad r_{ijk} \sim N(0, \sigma^2)$$

L2 $$\pi_{0jk} = \beta_{00k} + \beta_{01k}X_{jk} + e_{0jk}$$
$$\pi_{1jk} = \beta_{10k} + \beta_{11k}X_{jk} + e_{1jk}$$
$$\begin{bmatrix} e_{0jk} \\ e_{1jk} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00}^{(2)} & \\ \tau_{10}^{(2)} & \tau_{11}^{(2)} \end{bmatrix}\right)$$

L3 $$\beta_{00k} = \gamma_{000} + u_{00k}$$
$$\beta_{10k} = \gamma_{100} + u_{10k}$$
$$\beta_{01k} = \gamma_{010} + u_{01k}$$
$$\beta_{11k} = \gamma_{110} + u_{11k}$$
$$\begin{bmatrix} u_{00k} \\ u_{10k} \\ u_{01k} \\ u_{11k} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00}^{(3)} & & & \\ \tau_{10}^{(3)} & \tau_{11}^{(3)} & & \\ \tau_{20}^{(3)} & \tau_{21}^{(3)} & \tau_{22}^{(3)} & \\ \tau_{30}^{(3)} & \tau_{31}^{(3)} & \tau_{32}^{(3)} & \tau_{33}^{(3)} \end{bmatrix}\right)$$

These notations should represent (the blue lines indicate that the meanings changed from Model 23)

- Y_{ijk} = The math achievement score of Measurement *i* in Student *j* in School *k*
- t_{ijk} = The grade that the Measurement *i* in Student *j* in School *k* was observed
- X_{jk} = The gender of Student *j* in School *k* (1 = Male, 0 = Female)
- π_{0jk} = The math achievement score of Student *j* in School *k* at Grade 7
- π_{1jk} = The expected change in math achievement score when grade increases by 1 for Student *j* in School *k*, which is the rate of change for Student *j* in School *k*
- β_{00k} = The average of math achievement scores in Grade 7 across all female students within School *k*
- β_{10k} = The average rate of change in math achievement scores across all female students within School *k*
- β_{01k} = The gender difference in the average of math achievement scores in Grade 7 within School *k*
- β_{11k} = The gender difference in the rate of change within School *k*
- γ_{000} = The average of math achievement scores in Grade 7 across all female students across schools
- γ_{100} = The average rate of change in math achievement scores across all female students across schools
- γ_{010} = The average gender difference in math achievement scores in Grade 7 across schools
- γ_{110} = The average gender difference in the rate of change across schools
- r_{ijk} = The difference between the actual math achievement score of Measurement *i* in Student *j* in School *k* and the expected score of Student *j* in School *k* at a given grade level
- e_{0jk} = The deviation of the actual math achievement score of Student *j* in School *k* at Grade 7 from the average math achievement score at Grade 7 across students with the same gender within School *k*
- e_{1jk} = The deviation of the rate of change of Student *j* in School *k* from the average rate of change across students with the same gender within School *k*

- u_{00k} = The deviation of the actual female math achievement score of School k at Grade 7 from the average female math achievement score at Grade 7 across schools
- u_{10k} = The deviation of the female rate of change of School k from the average female rate of change across schools
- u_{01k} = The deviation of the gender difference in math achievement score of School k in Grade 7 from the average gender defferences of math achievement score in Grade 7 across schools
- u_{11k} = The deviation of the gender difference in rate of change of School k from the average gender difference in rate of change across schools
- σ^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade
- $\tau_{00}^{(2)}$ = The residual variance of math achievement scores at Grade 7 (initial status) within the student level (controlling for schools and gender)
- $\tau_{11}^{(2)}$ = The residual variance of the rate of change in math achievement score within the student level (controlling for schools and gender)
- $\tau_{10}^{(2)}$ = The residual covariance between the math achievement score at Grade 7 (initial status) and the rate of change within the student level (controlling for schools and gender)
- $\rho_{10}^{(2)} = \tau_{10}^{(2)} / \sqrt{\tau_{00}^{(2)} \tau_{11}^{(2)}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)
- $\tau_{00}^{(3)}$ = The variance of female math achievement scores at Grade 7 across schools
- $\tau_{11}^{(3)}$ = The variance of the female rate of change in math achievement score across schools
- $\tau_{22}^{(3)}$ = The variance of the gender difference in math achievement scores at Grade 7 across schools
- $\tau_{33}^{(3)}$ = The variance of the gender difference in rate of change across schools
- $\tau_{10}^{(3)}$ = The covariance between female math achievement score at Grade 7 and female rate of change at the school level
- $\tau_{20}^{(3)}$ = The covariance between female math achievement score at Grade 7 and the gender difference in math achievement scores at Grade 7 at the school level
- $\tau_{21}^{(3)}$ = The covariance between female rate of change and the gender difference in math achievement scores at Grade 7 at the school level
- $\tau_{30}^{(3)}$ = The covariance between female math achievement score at Grade 7 and the gender difference in rate of change at the school level
- $\tau_{31}^{(3)}$ = The covariance between female rate of change and the gender difference in rate of change at the school level
- $\tau_{32}^{(3)}$ = The covariance between the gender difference in math achievement scores at Grade 7 and the gender difference in rate of change at the school level
- $\rho_{st}^{(3)} = \tau_{st}^{(3)} / \sqrt{\tau_{ss}^{(3)} \tau_{tt}^{(3)}}$ (where $s, t = 0$, 1, 2, or 3 and $s \neq t$) = The covariance mentioned above in the correlation scale (from -1 to 1)

Next, the model with linear trajectory can be run by the `lmer` function:

```
m24 <- lmer(mathach ~ 1 + gradec + gender + gradec*gender + (1 + gradec|caseid) + (1 + gradec +
gender + gradec*gender|schoolid), data=long, REML=FALSE)

summary(m24)
```

```
Linear mixed model fit by maximum likelihood
Formula: mathach ~ 1 + gradec + gender + gradec * gender + (1 + gradec |      caseid) + (1 + gradec + gender +
gradec * gender | schoolid)
   Data: long
    AIC    BIC logLik deviance REMLdev
 129774 129915 -64869   129738  129744
Random effects:
 Groups   Name              Variance  Std.Dev. Corr
 caseid   (Intercept)       71.494814 8.45546
          gradec             2.071278 1.43919  0.349
 schoolid (Intercept)       20.438957 4.52095
          gradec             0.358130 0.59844  -0.088
          gendermale         5.208332 2.28218   0.416 -0.611
          gradec:gendermale  0.082743 0.28765  -0.007  0.991 -0.505
 Residual                   20.361267 4.51235
Number of obs: 19041, groups: caseid, 5858; schoolid, 95

Fixed effects:
                  Estimate Std. Error t value
(Intercept)       51.47943    0.51408  100.14
gradec             3.19418    0.08216   38.88
gendermale        -0.85434    0.37284   -2.29
gradec:gendermale  0.29289    0.07662    3.82

Correlation of Fixed Effects:
            (Intr) gradec gndrml
gradec      -0.171
gendermale   0.046 -0.241
grdc:gndrml  0.037 -0.032 -0.345
```

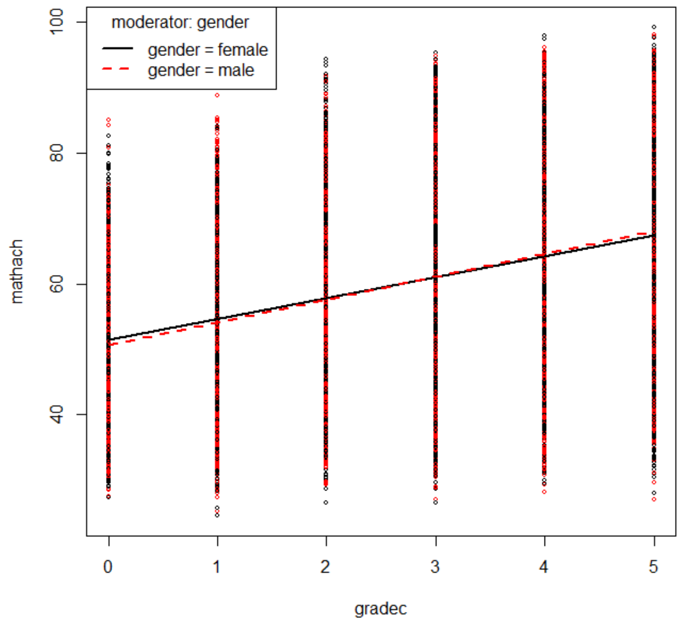
The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + gradec + gender + gradec*gender
             + (1 + gradec|caseid)
             + (1 + gradec + gender + gradec*gender|schoolid)
```

$Y_{ij} = \gamma_{000}(1) + \gamma_{100}(t_{ij} - 7) + \gamma_{010}X_{jk} + \gamma_{110}(t_{ij} - 7)X_{jk}$	Fixed Effect
$+e_{0jk}(1) + e_{1jk}(t_{ij} - 7)$ $+u_{00k}(1) + u_{10k}(t_{ij} - 7) + u_{01k}X_{ij} + u_{11k}(t_{ij} - 7)X_{ij} + r_{ijk}$	Random Effect

The gender difference in the linear trajectory was significant, $z = 3.82$, $p < .001$. Again, the `plotSlopes.mlm` and `testSlopes.mlm` function of the `rockchalkMultilevel` package can be used to visualize the interaction:

```
library(rockchalkMultilevel)

simpleSlope24 <- plotSlopes.mlm(m24, "gradec", "gender")
```




```
testSlopes.mlm(simpleSlope24)
```

```
These are the straight-line "simple slopes" of the variable gradec
 for the selected moderator values.
                "gender"    slope Std. Error  z value      Pr(>|z|)
female            gradec 3.194183 0.08216338 38.87600  0.000000e+00
male   gradec:gendermale 3.487068 0.11055343 31.54193 2.313728e-218
```

The linear trajectories for both males and females were significant; however, the magnitude of gender difference on the linear trajectories was trivial according to the graph.

We can simultaneously test whether the gender difference in initial status and linear trajectory were random across schools. The reference model with fixed effect of gender differences on initial status and linear trajectory was built and compared with the current model:

```
m24a <- lmer(mathach ~ 1 + gradec + gender + gradec*gender + (1 + gradec|caseid) + (1 +
gradec|schoolid), data=long, REML=FALSE)

anova(m24a, m24)
```

```
Data: long
Models:
m24a: mathach ~ 1 + gradec + gender + gradec * gender + (1 + gradec |
m24a:     caseid) + (1 + gradec | schoolid)
m24: mathach ~ 1 + gradec + gender + gradec * gender + (1 + gradec |
m24:     caseid) + (1 + gradec + gender + gradec * gender | schoolid)
     Df    AIC    BIC logLik  Chisq Chi Df Pr(>Chisq)
m24a 11 129793 129879 -64885
m24  18 129774 129915 -64869 32.803      7  2.882e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The random effects of gender differences on initial status and linear trajectory across schools were significant, $\chi^2(7) = 32.80$, $p < .001$. Thus, although the magnitude of gender difference in linear trajectories was trivial on average, the gender difference was varied across schools. That is, some schools had higher rates of linear change for males and other schools had higher rates of linear change for females.

Model 25: Level-3 Time-Invariant Covariate

In this model, the linear change of math achievement scores (`mathach`) across grade (`grade`) is predicted by school's cohort size (Level 3 time-invariant covariate). The school's cohort size (`cohortsize`) was centered at its grand mean. The three-level growth curve model with L3 covariate would be

L1 $$Y_{ijk} = \pi_{0jk} + \pi_{1jk}(t_{ijk} - 7) + r_{ijk} \qquad r_{ijk} \sim N(0, \sigma^2)$$

L2 $$\pi_{0jk} = \beta_{00k} + e_{0jk}$$ $$\pi_{1jk} = \beta_{10k} + e_{1jk} \qquad \begin{bmatrix} e_{0jk} \\ e_{1jk} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00}^{(2)} & \\ \tau_{10}^{(2)} & \tau_{11}^{(2)} \end{bmatrix}\right)$$

L3 $$\beta_{00k} = \gamma_{000} + \gamma_{001}(W_k - \overline{W}_{...}) + u_{00k}$$ $$\beta_{10k} = \gamma_{100} + \gamma_{101}(W_k - \overline{W}_{...}) + u_{10k} \qquad \begin{bmatrix} u_{00k} \\ u_{10k} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00}^{(3)} & \\ \tau_{10}^{(3)} & \tau_{11}^{(3)} \end{bmatrix}\right)$$

These notations should represent (the blue lines indicate that the meanings changed from Model 23)

- Y_{ijk} = The math achievement score of Measurement *i* in Student *j* in School *k*
- t_{ijk} = The grade that the Measurement *i* in Student *j* in School *k* was observed
- W_k = The cohort size in School *k*
- π_{0jk} = The math achievement score of Student *j* in School *k* at Grade 7
- π_{1jk} = The expected change in math achievement score when grade increases by 1 for Student *j* in School *k*, which is the rate of change for Student *j* in School *k*
- β_{00k} = The average of math achievement scores in Grade 7 across students within School *k*
- β_{10k} = The average rate of change in math achievement scores across students within School *k*
- γ_{000} = The expected school math achievement scores in Grade 7 when the cohort size equals the average cohort size across schools
- γ_{100} = The expected school rate of change in math achievement scores when the cohort size equals the average cohort across schools
- γ_{001} = The expected change in school math achievement score in Grade 7 when cohort size increases by 1
- γ_{101} = The expected change in school rate of change when cohort size increases by 1
- r_{ijk} = The difference between the actual math achievement score of Measurement *i* in Student *j* in School *k* and the expected score of Student *j* in School *k* at a given grade level
- e_{0jk} = The deviation of the actual math achievement score of Student *j* in School *k* at Grade 7 from the average math achievement score at Grade 7 across students within School *k*
- e_{1jk} = The deviation of the rate of change of Student *j* in School *k* from the average rate of change across students within School *k*
- u_{00k} = The deviation of the actual math achievement score of School *k* at Grade 7 from the predicted school math achievement score at Grade 7 (given cohort size)
- u_{10k} = The deviation of the rate of change of School *k* from the predicted school rate of change (given cohort size)

- σ^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade
- $\tau_{00}^{(2)}$ = The variance of math achievement scores at Grade 7 within the student level (partialled out the school variances)
- $\tau_{11}^{(2)}$ = The variance of the rate of change in math achievement score within the student level (partialled out the school variances)
- $\tau_{10}^{(2)}$ = The covariance between the math achievement score at Grade 7 (initial status) and the rate of change within the student level
- $\rho_{10}^{(2)} = \tau_{10}^{(2)} \big/ \sqrt{\tau_{00}^{(2)}\tau_{11}^{(2)}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)
- $\tau_{00}^{(3)}$ = The residual variance of math achievement scores at Grade 7 across schools controlling for cohort size
- $\tau_{11}^{(3)}$ = The residual variance of the rate of change in math achievement score across schools controlling for cohort size
- $\tau_{10}^{(3)}$ = The residual covariance between the math achievement score at Grade 7 (initial status) and the rate of change at the school level controlling for cohort size
- $\rho_{10}^{(3)} = \tau_{10}^{(3)} \big/ \sqrt{\tau_{00}^{(3)}\tau_{11}^{(3)}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)

Before analyzing data, the `cohortsize` variable needs to be centered at its grand mean:

```
long$cohortsizeC <- long$cohortsize - mean(long$cohortsize, na.rm = TRUE)
```

Next, the model with linear trajectory can be run by the `lmer` function:

```
m25 <- lmer(mathach ~ 1 + gradec + cohortsizeC + gradec*cohortsizeC + (1 + gradec|caseid) + (1 +
gradec|schoolid), data=long, REML=FALSE)

summary(m25)
```

```
Linear mixed model fit by maximum likelihood
Formula: mathach ~ 1 + gradec + cohortsizeC + gradec * cohortsizeC + (1 +      gradec | caseid) + (1 + gradec |
schoolid)
   Data: long
   AIC   BIC logLik deviance REMLdev
 69023 69103 -34501    69001   69028
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 caseid   (Intercept) 69.97730 8.36524
          gradec       2.00823 1.41712  0.400
 schoolid (Intercept) 26.55133 5.15280
          gradec       0.62167 0.78846  -0.303
 Residual             20.79327 4.55996
Number of obs: 10121, groups: caseid, 3085; schoolid, 64

Fixed effects:
                     Estimate Std. Error t value
(Intercept)         50.8903548  0.7057707   72.11
gradec               3.3798438  0.1216957   27.77
cohortsizeC         -0.0026315  0.0012201   -2.16
gradec:cohortsizeC  0.0004707  0.0003407    1.38

Correlation of Fixed Effects:
            (Intr) gradec chrtsC
gradec      -0.356
cohortsizeC  0.015 -0.007
grdc:chrtsC  0.000  0.016 -0.360
```

The mapping from the formula and reduced-form equation would be

```
langPOST ~ 1 + gradec + cohortsizeC + gradec*cohortsizeC
           + (1 + gradec|caseid)
           + (1 + gradec|schoolid)
```

$$Y_{ij} = \gamma_{000}(1) + \gamma_{100}(t_{ij} - 7) + \gamma_{010}X_{jk} + \gamma_{110}(t_{ij} - 7)X_{jk}$$

Fixed Effect

$$+e_{0jk}(1) + e_{1jk}(t_{ij} - 7)$$
$$+u_{00k}(1) + u_{10k}(t_{ij} - 7) + r_{ijk}$$

Random Effect

The linear growth was not significantly moderated by cohort size, $z = 1.38$, $p = .17$. Thus, the interaction will not be probed. The schools with higher cohort size, however, significantly had lower average math achievement scores, $z = -2.16$, $p = .031$. Note that this model can be compared with Model 23. Readers are encouraged to implement and interpret the deviance test between this model and Model 23.

Multivariate Model

In this part, multiple dependent variables are predicted by independent variables simultaneously. Rather than running models for each dependent variable separately, researchers can investigate the covariance between dependent variables at each level. Furthermore, the statistical power is higher especially for correlated dependent variables. The R code for multivariate models is much more complicated, however.

Restructuring Data for Multivariate Models

Multivariate model requires a special format of data structure. In this section, we will use the `long` dataset again:

```
long <- read.csv("mathgrowth.csv", header = TRUE, na.strings="-999999")
```

In this data set, rows represent L1 units. Before starting restructure data for multivariate models, users should transform predictors (e.g., centering or changing to factor format) for their future analyses at this point. For this data set, the `grade` variable will be centered at Grade 7 and the `gender` variable will be changed into the factor format for later uses:

```
long$gradec <- long$grade - 7

long$gender <- factor(long$gender, labels=c("female", "male"))
```

As another requirement, the data set must have a variable representing L1 ID and all rows must have distinct L1 ID values. The easiest way to create such ID variable is to create a variable of row index:

```
long <- data.frame(long, obs = 1:nrow(long))
```

The `obs` variable is attached to the `long` data set, which is simply a sequence from one to the total number of rows.

In this section, we will use two dependent variables: parent encouragement in studying math (`parentpush`) and peer encouragement in studying math (`peerpush`). These two variables are listed in two separate columns. For example,

L1ID	L2ID	DV1	DV2

1	1	1	5
2	1	4	7
3	2	2	5
4	2	3	9
5	3	1	4
6	3	3	6

This data set must be transformed such that one column represents both dependent variables. In other words, two (or more) dependent variables are stacked into one dependent variable. Each row represents data from one dependent variable of one L1 unit. That is, each L1 unit has multiple rows in the data set to represent each dependent variable. We can say that dependent variables are listed as another level lower than Level 1. Then, one variable are created to indicate which rows represent the first and the second dependent variables. For example, the previous table should be transformed as

L1ID	L2ID	DVS	IND
1	1	1	DV1
2	1	4	DV1
3	2	2	DV1
4	2	3	DV1
5	3	1	DV1
6	3	3	DV1
1	1	1	DV2
2	1	4	DV2
3	2	2	DV2
4	2	3	DV2
5	3	1	DV2
6	3	3	DV2

where DVS represents the stacked dependent variable values and IND represents the dependent variable each row represents. To transform the data in such format, the `melt` function in the `reshape2` package will be used.

```
library(reshape2)

dvvars <- c("parentpush", "peerpush")

othervars <- setdiff(colnames(long), dvvars)

long2 <- melt(long, id.vars = othervars, measure.vars = dvvars)
```

The `dvvars` object is the names of dependent variables. The `othervars` object is the names of other variables, which is created by the difference between all variable names and dependent variable names (by the `setdiff` function). In the `melt` function, the first argument is a target data set. The names of dependent variables are put in the `measure.vars` argument. The names of other variables are put in the `id.vars` argument. The resulting data can be viewed:

```
head(long2, 10)
```

```
  caseid schoolid grade mathach likemath gender race gradec obs   variable value
1      1      101     7   70.05        1   male    3      0   1 parentpush     2
2      1      101     8   69.23        1   male    3      1   2 parentpush     2
3      1      101     9   71.07        1   male    3      2   3 parentpush     2
4      1      101    10   78.52        1   male    3      3   4 parentpush     2
```

```
5       1      101    11   81.66        1  male   3      4   5 parentpush     2
6       1      101    12   78.77        1  male   3      5   6 parentpush     2
7       2      101     7   59.36        3  male   3      0   7 parentpush     2
8       2      101     8   65.20        3  male   3      1   8 parentpush     0
9       2      101     9   68.92        3  male   3      2   9 parentpush     1
10      2      101    10   72.06        3  male   3      3  10 parentpush     2
```

Readers can check the number of rows in the old and new data sets. The number of rows should be doubled in the new data set. Note that the `variable` variable is the name of dependent variables in each row and the `value` variable is the stacked dependent variables. Finally, two dummy variables are created to represent the indicators of each dependent variable:

```
long2$constparent <- as.numeric(long2$variable == "parentpush")

long2$constpeer <- as.numeric(long2$variable == "peerpush")
```

Model 26: Multivariate Null Model

Parent encouragement in studying math (`parentpush`) and peer encouragement in studying math (`peerpush`) will be used as dependent variables. We will run multivariate null model to find the variances and covariance of both variables at measurement and student levels. The multivariate null model would be

L1

$$Y_{Aij} = \beta_{0j} + e_{Aij}$$
$$Y_{Bij} = \beta_{2j} + e_{Bij}$$
$$\begin{bmatrix} e_{Aij} \\ e_{Bij} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_A^2 & \\ \sigma_{BA} & \sigma_B^2 \end{bmatrix}\right)$$

L2

$$\beta_{0j} = \gamma_{00} + u_{0j}$$
$$\beta_{2j} = \gamma_{20} + u_{2j}$$
$$\begin{bmatrix} u_{0j} \\ u_{2j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & \\ \tau_{20} & \tau_{22} \end{bmatrix}\right)$$

These notations represent

- Y_{Aij} = The parent encouragement in studying math of Measurement *i* in Student *j*
- Y_{Bij} = The peer encouragement in studying math of Measurement *i* in Student *j*
- β_{0j} = The average of parent encouragement of Student *j*
- β_{2j} = The average of peer encouragement of Student *j*
- γ_{00} = The average of parent encouragement across students
- γ_{20} = The average of peer encouragement across students
- e_{Aij} = The deviation of the parent encouragement of Measurement *i* in Student *j* from the Student *j* average
- e_{Bij} = The deviation of the peer encouragement of Measurement *i* in Student *j* from the Student *j* average
- u_{0j} = The deviation of the parent encouragement of Student *j* from the grand mean
- u_{2j} = The deviation of the peer encouragement of Student *j* from the grand mean
- σ_A^2 = The parent encouragement variance within participants (L1 variance)
- σ_B^2 = The peer encouragement variance within participants (L1 variance)
- σ_{BA} = The covariance between parent encouragement and peer encouragement within participants, which is the covariance between both scores across measurements within a student
- $\rho_{BA}^{(1)} = \sigma_{BA}/\sqrt{\sigma_A^2\sigma_B^2}$ = The covariance mentioned above in the correlation scale (from -1 to 1)
- τ_{00} = The parent encouragement variance across participants (L2 variance)
- τ_{22} = The peer encouragement variance across participants (L2 variance)

- τ_{20} = The covariance between parent encouragement and peer encouragement across participants
- $\rho_{20}^{(2)} = \tau_{20}/\sqrt{\tau_{00}\tau_{22}}$ = The covariance mentioned above in the correlation scale (from -1 to 1)

The formulas listed above with two dependent variables can be condensed into a formula with one dependent variable with the help of dummy variables:

L1 $$\tilde{Y}_{ij} = \delta_A(\beta_{0j}) + \delta_B(\beta_{2j}) + \tilde{e}_{ij} \qquad \tilde{e}_{ij} \sim N(0, \tilde{\sigma}^2)$$

where

- $\tilde{Y}_{ij}$ = The stacked values of dependent variables
- δ_A = A dummy variable where 1 represents rows from parent encouragement and 0 represents rows from peer encouragement
- δ_B = A dummy variable where 1 represents rows from peer encouragement and 0 represents rows from parent encouragement
- $\tilde{e}_{ij}$ = The stacked L1 residuals
- $\tilde{\sigma}^2$ = The variance of stacked L1 residuals, which the error structure is based on different dependent variables

The reduced form of this formula would be

$$Y_{ij} = \delta_A\gamma_{00} + \delta_B\gamma_{20} + u_{0j}\delta_A + u_{2j}\delta_B + e_{ij}$$

Notice that this model does not have the intercept term (no fixed effect multiplied by just 1). All regression coefficients are multiplied by dummy variables.

Because L1 residuals have different variances for each dependent variable, the `lme4` package cannot be used. The `lme` function from the `nlme` package is used instead:

```
m26 <- lme(value ~ 0 + constparent + constpeer, data = long2,

        random = ~0 + constparent + constpeer | caseid,

        correlation = corSymm(form = ~ 1 | caseid/obs),

        weights = varIdent(form = ~1 | variable), method = "ML", na.action = "na.omit")

summary(m26)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: long2
       AIC      BIC    logLik
  116863.4 116932.4 -58423.69

Random effects:
 Formula: ~0 + constparent + constpeer | caseid
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
constparent 0.5409066 cnstpr
constpeer   0.4687826 0.23
Residual    0.8717789

Correlation Structure: General
 Formula: ~1 | caseid/obs
 Parameter estimate(s):
 Correlation:
  1
2 0.245
Variance function:
```

```
 Structure: Different standard deviations per stratum
 Formula: ~1 | variable
 Parameter estimates:
parentpush   peerpush
  1.000000   1.098731
Fixed effects: value ~ 0 + constparent + constpeer
                Value   Std.Error    DF   t-value p-value
constparent 1.4054332 0.009593400 35335 146.50000       0
constpeer   0.8088264 0.009252692 35335  87.41526       0
 Correlation:
          cnstprn
constpeer 0.227

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.29442504 -0.67953407 -0.02575005  0.50227314  3.42114375

Number of Observations: 41280
Number of Groups: 5944
```

The first argument is the formula of fixed effects. Because no intercept is used, `0` is used instead of `1`. In the `random` argument, the coefficients of two dummy variables, which are β_{0j} and β_{2j}, are varied across students. Again, `0` is used because the model does not have intercepts. The `weight` argument is specified such that the variances are equal within the same value of dependent variables (the `variable` variable). Different dependent variables can have different L1 residual variances.

Specifying the `correlation` argument is tricky. This `caseid/obs` expression means all L1 ID (measurement ID) within L2 ID (student ID). Thus, `corSymm(form = ~ 1 | caseid/obs)` means the correlation structure is symmetric (`corSymm`) within any L1 units (`~ 1 | caseid/obs`). Because there are two rows representing different dependent variables within a L1 unit, the correlation represents the relationship between two dependent variables within L1 units.[21]

The mapping from the formula and reduced-form equation would be

```
formula = value ~ 0 + constparent + constpeer
random = ~ 0 + constparent + constpeer | caseid
```

$$Y_{ij} = \underbrace{\delta_A\gamma_{00} + \delta_B\gamma_{20}}_{\text{Fixed Effect}} + \underbrace{u_{0j}\delta_A + u_{2j}\delta_B + e_{ij}}_{\text{Random Effect}}$$

From this output, the correlation between two variables at measurement and student levels were .245 (in the `Correlation Structure: General` section) and .230 (in the `Random effects` section), respectively. The L1 residual standard deviation of peer encouragement was 1.099 times higher than the L1 residual standard deviation of parent encouragement (in the `Variance function` section).

[21] The alternative code for this model is

```
m26 <- lme(value ~ 0 + variable, data = long2,

        random = ~0 + variable | caseid,

        correlation = corSymm(form = ~ 1 | caseid/obs),

        weights = varIdent(form = ~1 | variable), method = "ML", na.action = "na.omit")

summary(m26)
```

Because the `variable` variable is in a factor format, `0 + variable` will create two dummy variables for each group (which are similar to `constparent` and `constpeer`).

Researchers may wish to calculate intraclass correlations (ICC) for both dependent variables. Here are the steps to calculate ICC:

1. Extract the variances of random effects and residual variance by the `VarCorr` function:

```
allvar <- VarCorr(m26)

allvar
```

```
caseid = pdLogChol(0 + constparent + constpeer)
            Variance  StdDev    Corr
constparent 0.2925799 0.5409066 cnstprn
constpeer   0.2197571 0.4687826 0.23
Residual    0.7599985 0.8717789
```

2. Get the L1 residual standard deviation of the first variable:

```
dv1sd <- as.numeric(allvar[3, 2])
```

   The `as.numeric` function is needed to transform the text format to the number format.

3. Compute the L1 residual standard deviation of all variables:

```
l1sd <- c(dv1sd, dv1sd * coef(m26$modelStruct$varStruct, unconstrained = FALSE))
```

   The code used to find the ratio of standard deviations between the first and the rest of dependent variables are similar to the scripts provided in Model 19.

4. Compute the L1 residual variance of all variables:

```
l1var <- l1sd^2
```

5. Get the L2 variances of all variables:

```
l2var <- as.numeric(allvar[1:2, 1])
```

6. Compute ICC of all variables:

```
icc <- l2var / (l1var + l2var)

icc
```

```
          peerpush
0.2779650 0.1932381
```

   The intraclass correlations of parent and peer encouragement are .28 and .19, respectively.

The average of parent and peer encouragement across students were 1.41 and 0.81, respectively. We can compare whether these averages are different. There are two options for this comparison: multi-parameter contrast and deviance test. The multi-parameter contrast can be implemented by the `glht` function from the `multcomp` package:

```
library(multcomp)

ctr <- glht(m26, "constparent - constpeer = 0")

summary(ctr)
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: lme.formula(fixed = value ~ 0 + constparent + constpeer, data = long2,
```

```
    random = ~0 + constparent + constpeer | caseid, correlation = corSymm(form = ~1 |
        caseid/obs), weights = varIdent(form = ~1 | variable),
    method = "ML", na.action = "na.omit")

Linear Hypotheses:
                             Estimate Std. Error z value Pr(>|z|)
constparent - constpeer == 0  0.59661    0.01172   50.91   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

The average of parent encouragement was significantly higher than peer encouragement by 0.60 points, $z = 50.9$, $p < .001$.

Another way is to compare the levels of parent and peer encouragements by deviance test of nested models. We need the reference model that the fixed effects of parent and peer encouragement are equal, $\gamma_{00} = \gamma_{20}$. If we create this model, the reduced-form formula would be

$$Y_{ij} = \delta_A\gamma_{00} + \delta_B\gamma_{00} + u_{0j}\delta_A + u_{2j}\delta_B + e_{ij} = (\delta_A + \delta_B)\gamma_{00} + u_{0j}\delta_A + u_{2j}\delta_B + e_{ij}$$

Because $\delta_A + \delta_B = 1$ in all cases (the row is either parent or peer encouragement), our reference model with equal fixed effects of both dependent variables would be

$$Y_{ij} = \gamma_{00} + u_{0j}\delta_A + u_{2j}\delta_B + e_{ij}$$

which can be translated into the R script:

```
m26a <- lme(value ~ 1, data = long2,

        random = ~0 + constparent + constpeer | caseid,

        correlation = corSymm(form = ~ 1 | caseid/obs),

        weights = varIdent(form = ~1 | variable), method = "ML", na.action = "na.omit")
```

The formula of the reference model has only the intercept (`1`) because the intercepts of both dependent variables are equal (γ_{00}). This model can be compared with the original model by the deviance test:

```
anova(m26, m26a)
```

```
     Model df      AIC      BIC    logLik   Test  L.Ratio p-value
m26      1  8 116863.4 116932.4 -58423.69
m26a     2  7 118994.7 119055.1 -59490.35 1 vs 2 2133.333  <.0001
```

Similar to the multi-parameter contrast, the difference between the averages of parent and peer encouragement was significant, $\chi^2(1) = 2133.33$, $p < .001$.

The multi-parameter contrast can be used to compare fixed effects but not to compare random effects. The deviance test can be used to compare whether the random effects of both dependent variables are identical, $u_{0j} = u_{2j}$. The reduced-form formula can be transformed as

$$\begin{aligned} Y_{ij} &= \delta_A\gamma_{00} + \delta_B\gamma_{20} + u_{0j}\delta_A + u_{0j}\delta_B + e_{ij} = \delta_A\gamma_{00} + \delta_B\gamma_{20} + u_{0j}(\delta_A + \delta_B) + e_{ij} \\ &= \delta_A\gamma_{00} + \delta_B\gamma_{20} + u_{0j} + e_{ij} \end{aligned}$$

where $\delta_A + \delta_B = 1$. The R scipt for this reference model would be

```
m26b <- lme(value ~ 0 + constparent + constpeer, data = long2,
```

```
        random = ~1 | caseid,

        correlation = corSymm(form = ~ 1 | caseid/obs),

        weights = varIdent(form = ~1 | variable), method = "ML", na.action = "na.omit")
```

You can see that the `random` argument is defined by the intercept, `1`, which represents u_{0j} only. This model can be compared with the original model by the deviance test:

```
anova(m26, m26b)
```

```
     Model df      AIC      BIC    logLik   Test  L.Ratio p-value
m26      1  8 116863.4 116932.4 -58423.69
m26b     2  6 118241.1 118292.9 -59114.56 1 vs 2 1381.745  <.0001
```

The random effects of parent and peer encouragements are significantly different, $\chi^2(2) = 1381.75$, $p < .001$. I think that this comparison is nonsense. The reference model says that random effect values are identical in each model. For example, in Student 1, if the random effect of parent encouragement, u_{0j}, is 2, the random effect of peer encouragement, u_{2j}, is 2 too. Because these values are based on totally different dependent variables, constraining two random effects of two different dependent varaibles to be equal is nonsense. I will not mention this test further in this section.

Model 27: Multivariate Linear Growth Model

The linear trajectories of both parent encouragement (`parentpush`) and peer encouragement (`peerpush`) across grades are modeled. The `grade` variable is centered at Grade 7, which has been done during the restructuring process. The multivariate linear growth model would be

L1
$$Y_{Aij} = \beta_{0j} + \beta_{1j}(t_{ij} - 7) + e_{Aij}$$
$$Y_{Bij} = \beta_{2j} + \beta_{3j}(t_{ij} - 7) + e_{Bij}$$
$$\begin{bmatrix} e_{Aij} \\ e_{Bij} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_A^2 & \\ \sigma_{BA} & \sigma_B^2 \end{bmatrix}\right)$$

L2
$$\beta_{0j} = \gamma_{00} + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + u_{1j}$$
$$\beta_{2j} = \gamma_{20} + u_{2j}$$
$$\beta_{3j} = \gamma_{30} + u_{3j}$$
$$\begin{bmatrix} u_{0j} \\ u_{1j} \\ u_{2j} \\ u_{3j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & & & \\ \tau_{10} & \tau_{11} & & \\ \tau_{20} & \tau_{21} & \tau_{22} & \\ \tau_{30} & \tau_{31} & \tau_{32} & \tau_{33} \end{bmatrix}\right)$$

These notations represent (the blue lines indicate that the meanings changed from Model 26)

- Y_{Aij} = The parent encouragement in studying math of Measurement *i* in Student *j*
- Y_{Bij} = The peer encouragement in studying math of Measurement *i* in Student *j*
- t_{ij} = The grade that the Measurement *i* in Student *j* was observed
- β_{0j} = The expected parent encouragement of Student *j* at Grade 7
- β_{1j} = The expected change in parent encouragement when grade increases by 1 for Student *j*, which is the rate of change in parent encouragement for Student *j*
- β_{2j} = The expected peer encouragement of Student *j* at Grade 7
- β_{3j} = The expected change in peer encouragement when grade increases by 1 for Student *j*, which is the rate of change in peer encouragement for Student *j*
- γ_{00} = The average of parent encouragement in Grade 7 across students
- γ_{10} = The average rate of change in parent encouragement across students
- γ_{20} = The average of peer encouragement in Grade 7 across students
- γ_{30} = The average rate of change in peer encouragement across students

- e_{Aij} = The difference between the actual parent encouragement score of Measurement *i* in Student *j* and the expected score of Student *j* at a given grade level
- e_{Bij} = The difference between the actual peer encouragement score of Measurement *i* in Student *j* and the expected score of Student *j* at a given grade level
- u_{0j} = The deviation of the actual parent encouragement score of Student *j* at Grade 7 from the average math achievement score at Grade 7 across students
- u_{1j} = The deviation of the rate of change in parent encouragement of Student *j* from the average rate of change across students
- u_{2j} = The deviation of the actual peer encouragement score of Student *j* at Grade 7 from the average math achievement score at Grade 7 across students
- u_{3j} = The deviation of the rate of change in peer encouragement of Student *j* from the average rate of change across students
- σ_A^2 = The parent encouragement score residual variance within the measurement level (L1 residual variance) controlling for grade
- σ_B^2 = The peer encouragement score residual variance within the measurement level (L1 residual variance) controlling for grade
- σ_{BA} = The residual covariance between parent and peer encouragement within participants controlling for grade
- $\rho_{BA}^{(1)} = \sigma_{BA}/\sqrt{\sigma_A^2\sigma_B^2}$ = The covariance mentioned above in the correlation scale (from -1 to 1)
- τ_{00} = The variance of parent encouragement scores at Grade 7 across students
- τ_{11} = The variance of the rate of change in parent encouragement score across students
- τ_{22} = The variance of peer encouragement scores at Grade 7 across students
- τ_{33} = The variance of the rate of change in peer encouragement score across students
- τ_{10} = The covariance between the parent encouragement score at Grade 7 and the rate of change in parent encouragement
- τ_{20} = The covariance between the parent and peer encouragement scores at Grade 7
- τ_{30} = The covariance between the parent encouragement score at Grade 7 and the rate of change in peer encouragement
- τ_{21} = The covariance between the rate of change in parent encouragement and the peer encouragement score at Grade 7
- τ_{31} = The covariance between the rate of change in parent encouragement and the rate of change in peer encouragement
- τ_{32} = The covariance between the peer encouragement score at Grade 7 and the rate of change in peer encouragement
- $\rho_{st} = \tau_{st}/\sqrt{\tau_{ss}\tau_{tt}}$ (where $s, t = 0$, 1, 2, or 3 and $s \neq t$) = The covariance mentioned above in the correlation scale (from -1 to 1)

The formulas listed above with two dependent variables can be condensed into a formula with one dependent variable with the help of dummy variables:

L1 $$\tilde{Y}_{ij} = \delta_A\left(\beta_{0j} + \beta_{1j}(t_{ij} - 7)\right) + \delta_B\left(\beta_{2j} + \beta_{3j}(t_{ij} - 7)\right) + \tilde{e}_{ij} \qquad \tilde{e}_{ij} \sim N(0, \tilde{\sigma}^2)$$

where notations are defined in Model 26.

The model can be analyzed by the `lme` function from the `nlme` package:

```
m27 <- lme(value ~ 0 + constparent + constparent:gradec + constpeer + constpeer:gradec,

        data = long2,

        random = ~0 + constparent + constparent:gradec + constpeer + constpeer:gradec| caseid,

        correlation = corSymm(form = ~ 1 | caseid/obs),

        weights = varIdent(form = ~1 | variable), method = "ML", na.action = "na.omit",

        control = lmeControl(maxIter = 500, msMaxIter = 500, niterEM = 100, msMaxEval = 400))

summary(m27)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: long2
       AIC      BIC    logLik
  112866.7 113013.4 -56416.35

Random effects:
 Formula: ~0 + constparent + constparent:gradec + constpeer + constpeer:gradec | caseid
 Structure: General positive-definite, Log-Cholesky parametrization
                   StdDev    Corr
constparent        0.7069079 cnstprn constpr cnstp:
constpeer          0.6833550  0.376
constparent:gradec 0.1086271 -0.528  -0.271
gradec:constpeer   0.1329360 -0.187  -0.701   0.265
Residual           0.7594011

Correlation Structure: General
 Formula: ~1 | caseid/obs
 Parameter estimate(s):
 Correlation:
  1
2 0.12
Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | variable
 Parameter estimates:
parentpush   peerpush
  1.000000   1.168075
Fixed effects: value ~ 0 + constparent + constparent:gradec + constpeer + constpeer:gradec
                        Value   Std.Error    DF   t-value p-value
constparent         2.0305157 0.015211684 35333 133.48395       0
constpeer           1.2778746 0.016440875 35333  77.72546       0
constparent:gradec -0.2220791 0.004016936 35333 -55.28569       0
gradec:constpeer   -0.1653800 0.004564085 35333 -36.23508       0
 Correlation:
                   cnstprn constpr cnstp:
constpeer           0.236
constparent:gradec -0.762  -0.165
gradec:constpeer   -0.150  -0.825   0.167

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-2.7390889 -0.6244660 -0.1355851  0.5876161  3.8055511

Number of Observations: 41280
Number of Groups: 5944
```

The script is similar to Model 26.[22] The differences are in the `formula` and the `random` arguments. The `gradec` variable is added to those formulas via the `constparent:gradec` and

[22] The alternative script is

```
m27 <- lme(value ~ 0 + variable + variable:gradec, data = long2,

        random = ~0 + variable + variable:gradec | caseid,

        correlation = corSymm(form = ~ 1 | caseid/obs),
```

`constpeer:gradec` terms, which are corresponding to $\delta_A\gamma_{10}(t_{ijk}-7)$ and $\delta_B\gamma_{30}(t_{ijk}-7)$. The `:` operator is used to create interaction. Note that the `*` operator cannot be used here because the lower-order terms must not be included.[23] The `control` argument is specified here because the default number of iterations is not enough to reach the convergent result. Please see further details in the help page (by typing `?lmeControl` in the R console).

The mapping from the formula and reduced-form equation would be

```
formula = value ~ 0 + constparent + constparent:gradec
          + constpeer + constpeer:gradec
random = ~ 0 + constparent + constparent:gradec
          + constpeer + constpeer:gradec | caseid
```

$$Y_{ij} = \delta_A\gamma_{00} + \delta_A\gamma_{10}(t_{ij}-7) + \delta_B\gamma_{20} + \delta_B\gamma_{30}(t_{ij}-7) \quad \text{Fixed Effect}$$

$$+u_{0j}\delta_A + u_{1j}\delta_A(t_{ij}-7) + u_{2j}\delta_B + u_{3j}\delta_B(t_{ij}-7) + e_{ij} \quad \text{Random Effect}$$

The parent scores were significantly declined across grades, z = -55.29, p < .001, as well as peer encouragement, z = -36.26, p < .001. I will illustrate two contrasts for comparing the levels of parent and peer encouragement at Grade 7 and comparing the rates of decline of two types of encouragement. As the first option, the `glht` function from the `multcomp` package can be used:

```
library(multcomp)

ctr <- glht(m27, c("constparent - constpeer = 0", "constparent:gradec - gradec:constpeer = 0"))

summary(ctr)
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: lme.formula(fixed = value ~ 0 + constparent + constparent:gradec +
    constpeer + constpeer:gradec, data = long2, random = ~0 +
    constparent + constparent:gradec + constpeer + constpeer:gradec |
    caseid, correlation = corSymm(form = ~1 | caseid/obs), weights = varIdent(form = ~1 |
    variable), method = "ML", na.action = "na.omit", control = lmeControl(maxIter = 500,
    msMaxIter = 500, niterEM = 100, msMaxEval = 400))

Linear Hypotheses:
                                         Estimate Std. Error z value Pr(>|z|)
constparent - constpeer == 0              0.752641   0.019588   38.42   <1e-10 ***
constparent:gradec - gradec:constpeer == 0 -0.056699   0.005553  -10.21   <1e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

Note that the contrasts must be written based on the names of fixed effect from the `summary` function (e.g., `constparent:gradec` and `gradec:constpeer`). After adjusting for the familywise error rate, both contrasts were significant. The parent encouragement was significantly higher than peer encouragement at Grade 7, z = 38.42, p < .001. The rate of decline of parent encouragement was significantly stronger than peer encouragement, z = -10.21, p < .001.

As the second option, we can create appropriate reference models and use deviance test to test target parameters. As the first comparison, we will compare the levels of parent and peer encouragement at

```
        weights = varIdent(form = ~1 | variable), method = "ML", na.action = "na.omit",

        control = lmeControl(maxIter = 500, msMaxIter = 500, niterEM = 100, msMaxEval = 400))
```

[23] The `constparent*gradec` term means `1 + constparent + gradec + constparent:gradec`, which includes lower-order terms

Grade 7 such that the reference model would have $\gamma_{00} = \gamma_{20}$. If we create this model, the $\delta_A\gamma_{10} + \delta_B\gamma_{20}$ term in the reduced-form formula can be replaced by $\delta_A\gamma_{00} + \delta_B\gamma_{00} = (\delta_A + \delta_B)\gamma_{00} = \gamma_{00}$. The reference model can be translated into the R script:

```
m27a <- lme(value ~ 1 + constparent:gradec + constpeer:gradec, data = long2,

        random = ~0 + constparent + constparent:gradec + constpeer + constpeer:gradec| caseid,

        correlation = corSymm(form = ~ 1 | caseid/obs),

        weights = varIdent(form = ~1 | variable), method = "ML", na.action = "na.omit",

        control = lmeControl(maxIter = 500, msMaxIter = 500, niterEM = 100, msMaxEval = 400))
```

The formula of the reference model has the intercept (`1`) instead of `constparent` and `constpeer` because the intercepts of both dependent variables are equal (γ_{00}). This model can be compared with the original model by the deviance test:

```
anova(m27, m27a)
```

```
     Model df      AIC      BIC    logLik   Test  L.Ratio p-value
m27      1 17 112866.7 113013.4 -56416.35                        
m27a     2 16 114143.7 114281.7 -57055.84 1 vs 2 1278.992  <.0001
```

Similar to the multi-parameter contrast, the difference between the averages of parent and peer encouragement at Grade 7 was significant, $\chi^2(1) = 1278.99$, $p < .001$. Note that this deviance test does not account for the familywise error rate.

As the second comparison, we will compare comparing the rates of decline of two types of encouragement such that the reference model would have $\gamma_{10} = \gamma_{30}$. If we create this model, the $\delta_A\gamma_{10}(t_{ij} - 7) + \delta_B\gamma_{30}(t_{ij} - 7)$ term in the reduced-form formula can be replaced by $\delta_A\gamma_{10}(t_{ij} - 7) + \delta_B\gamma_{30}(t_{ij} - 7) = (\delta_A + \delta_B)\gamma_{10}(t_{ij} - 7) = \gamma_{10}(t_{ij} - 7)$. The reference model can be translated into the R script:

```
m27b <- lme(value ~ 0 + gradec + constparent + constpeer, data = long2,

        random = ~0 + constparent + constparent:gradec + constpeer + constpeer:gradec| caseid,

        correlation = corSymm(form = ~ 1 | caseid/obs),

        weights = varIdent(form = ~1 | variable), method = "ML", na.action = "na.omit",

        control = lmeControl(maxIter = 500, msMaxIter = 500, niterEM = 100, msMaxEval = 400))
```

The formula of the reference model has the fixed effect of grade (`gradec`) instead of the products of grade with the dummy variables (`constparent:gradec` and `constpeer:gradec`) because the effects of grade of both dependent variables are equal (γ_{10}). This model can be compared with the original model by the deviance test:

```
anova(m27, m27b)
```

```
     Model df      AIC      BIC    logLik   Test  L.Ratio p-value
m27      1 17 112866.7 113013.4 -56416.35                        
m27b     2 16 112965.2 113103.2 -56466.60 1 vs 2 100.5027  <.0001
```

Similar to the multi-parameter contrast, the difference between the rates of decline of parent and peer encouragement was significant, $\chi^2(1) = 100.50$, $p < .001$. Again, this deviance test does not account for the familywise error rate.

Model 28: Multivariate Linear Growth Model with Time-Invariant Covariate

The linear trajectories of both parent encouragement (`parentpush`) and peer encouragement (`peerpush`) across grades are predicted by gender. The `gender` variable needs to be in the factor format, which has been done during the restructuring process. The multivariate linear growth model with time-invariant covariate would be

L1
$$Y_{Aij} = \beta_{0j} + \beta_{1j}(t_{ij} - 7) + e_{Aij}$$
$$Y_{Bij} = \beta_{2j} + \beta_{3j}(t_{ij} - 7) + e_{Bij}$$
$$\begin{bmatrix} e_{Aij} \\ e_{Bij} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_A^2 & \\ \sigma_{BA} & \sigma_B^2 \end{bmatrix}\right)$$

L2
$$\beta_{0j} = \gamma_{00} + \gamma_{01}W_j + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + \gamma_{11}W_j + u_{1j}$$
$$\beta_{2j} = \gamma_{20} + \gamma_{21}W_j + u_{2j}$$
$$\beta_{3j} = \gamma_{30} + \gamma_{31}W_j + u_{3j}$$
$$\begin{bmatrix} u_{0j} \\ u_{1j} \\ u_{2j} \\ u_{3j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & & & \\ \tau_{10} & \tau_{11} & & \\ \tau_{20} & \tau_{21} & \tau_{22} & \\ \tau_{30} & \tau_{31} & \tau_{32} & \tau_{33} \end{bmatrix}\right)$$

These notations represent (the blue lines indicate that the meanings are changed from Model 27)

- Y_{Aij} = The parent encouragement in studying math of Measurement *i* in Student *j*
- Y_{Bij} = The peer encouragement in studying math of Measurement *i* in Student *j*
- t_{ij} = The grade that the Measurement *i* in Student *j* was observed
- β_{0j} = The expected parent encouragement of Student *j* at Grade 7
- β_{1j} = The expected change in parent encouragement when grade increases by 1 for Student *j*, which is the rate of change in parent encouragement for Student *j*
- β_{2j} = The expected peer encouragement of Student *j* at Grade 7
- β_{3j} = The expected change in peer encouragement when grade increases by 1 for Student *j*, which is the rate of change in peer encouragement for Student *j*
- γ_{00} = The average of parent encouragement in Grade 7 across female students
- γ_{10} = The average rate of change in parent encouragement across female students
- γ_{20} = The average of peer encouragement in Grade 7 across female students
- γ_{30} = The average rate of change in peer encouragement across female students
- γ_{01} = The gender difference in parent encouragement at Grade 7
- γ_{11} = The gender difference in the rate of change in parent encouragement
- γ_{21} = The gender difference in peer encouragement at Grade 7
- γ_{31} = The gender difference in the rate of change in peer encouragement
- e_{Aij} = The difference between the actual parent encouragement score of Measurement *i* in Student *j* and the expected score of Student *j* at a given grade level
- e_{Bij} = The difference between the actual peer encouragement score of Measurement *i* in Student *j* and the expected score of Student *j* at a given grade level
- u_{0j} = The deviation of the actual parent encourangement score of Student *j* at Grade 7 from the average math achievement score at Grade 7 across students with the same gender
- u_{1j} = The deviation of the rate of change in parent encourangement of Student *j* from the average rate of change across students with the same gender

- u_{2j} = The deviation of the actual peer encourangement score of Student *j* at Grade 7 from the average math achievement score at Grade 7 across students with the same gender
- u_{3j} = The deviation of the rate of change in peer encourangement of Student *j* from the average rate of change across students with the same gender
- σ_A^2 = The parent encourangement score residual variance within the measurement level (L1 residual variance) controlling for grade
- σ_B^2 = The peer encourangement score residual variance within the measurement level (L1 residual variance) controlling for grade
- σ_{BA} = The residual covariance between parent encouragement and peer encouragement within participants controlling for grade
- $\rho_{BA}^{(1)} = \sigma_{BA}/\sqrt{\sigma_A^2\sigma_B^2}$ = The covariance mentioned above in the correlation scale (from -1 to 1)
- τ_{00} = The residual variance of parent encouragement scores at Grade 7 across students controlling for gender
- τ_{11} = The residual variance of the rate of change in parent encouragement score across students controlling for gender
- τ_{22} = The residual variance of peer encouragement scores at Grade 7 across students controlling for gender
- τ_{33} = The residual variance of the rate of change in peer encouragement score across students controlling for gender
- τ_{10} = The residual covariance between the parent encouragement score at Grade 7 and the rate of change in parent encouragement controlling for gender
- τ_{20} = The residual covariance between the parent encouragement score at Grade 7 and the peer encouragement score at Grade 7 controlling for gender
- τ_{30} = The residual covariance between the parent encouragement score at Grade 7 and the rate of change in peer encouragement controlling for gender
- τ_{21} = The residual covariance between the rate of change in parent encouragement and the peer encouragement score at Grade 7 controlling for gender
- τ_{31} = The residual covariance between the rate of change in parent encouragement and the rate of change in peer encouragement controlling for gender
- τ_{32} = The residual covariance between the peer encouragement score at Grade 7 and the rate of change in peer encouragement controlling for gender
- $\rho_{st} = \tau_{st}/\sqrt{\tau_{ss}\tau_{tt}}$ (where s, t = 0, 1, 2, or 3 and $s \neq t$) = The covariance mentioned above in the correlation scale (from -1 to 1)

The formulas listed above with two dependent variables can be condensed into a formula with one dependent variable with the help of dummy variables:

L1 $$\tilde{Y}_{ij} = \delta_A\left(\beta_{0j} + \beta_{1j}(t_{ijk} - 7)\right) + \delta_B\left(\beta_{2j} + \beta_{3j}(t_{ijk} - 7)\right) + \tilde{e}_{ij} \qquad \tilde{e}_{ij} \sim N(0, \tilde{\sigma}^2)$$

where notations are defined in Model 26.

The model can be analyzed by the `lme` function from the `nlme` package:

```
m28 <- lme(value ~ 0 + variable + variable:gradec + variable:gender + variable:gender:gradec,
```

```
        data = long2,

        random = ~0 + variable + variable:gradec| caseid,

        correlation = corSymm(form = ~ 1 | caseid/obs),

        weights = varIdent(form = ~1 | variable), method = "ML", na.action = "na.omit",

        control = lmeControl(maxIter = 500, msMaxIter = 500, niterEM = 100, msMaxEval = 400))

summary(m28)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: long2
       AIC      BIC    logLik
  112856.6 113037.7 -56407.28

Random effects:
 Formula: ~0 + variable + variable:gradec | caseid
 Structure: General positive-definite, Log-Cholesky parametrization
                          StdDev    Corr
variableparentpush        0.7068118 vrblprn vrblprp vrblp:
variablepeerpush          0.6822875  0.377
variableparentpush:gradec 0.1086279 -0.529  -0.274
variablepeerpush:gradec   0.1329276 -0.188  -0.701   0.267
Residual                  0.7594470

Correlation Structure: General
 Formula: ~1 | caseid/obs
 Parameter estimate(s):
 Correlation:
  1
2 0.119
Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | variable
 Parameter estimates:
parentpush   peerpush
  1.000000   1.167889
Fixed effects: value ~ 0 + variable + variable:gradec + variable:gender + variable:gender:gradec
                                     Value  Std.Error    DF   t-value p-value
variableparentpush                 2.0176147 0.02185064 35329  92.33662  0.0000
variablepeerpush                   1.2264624 0.02363324 35329  51.89564  0.0000
variableparentpush:gradec         -0.2260322 0.00571405 35329 -39.55727  0.0000
variablepeerpush:gradec           -0.1590402 0.00652426 35329 -24.37673  0.0000
variableparentpush:gendermale      0.0248452 0.03043982 35329   0.81621  0.4144
variablepeerpush:gendermale        0.0993509 0.03287836 35329   3.02177  0.0025
variableparentpush:gradec:gendermale  0.0082202 0.00803640 35329   1.02287  0.3064
variablepeerpush:gradec:gendermale   -0.0118730 0.00912959 35329  -1.30050  0.1934
 Correlation:
                                     vrblprn vrblprp vrblntpsh:gr vrblpsh:gr vrblprpsh:gn vrblpsh:gn vrblprn::
variablepeerpush                     0.236
variableparentpush:gradec           -0.768  -0.167
variablepeerpush:gradec             -0.152  -0.829   0.168
variableparentpush:gendermale       -0.718  -0.169   0.551        0.109
variablepeerpush:gendermale         -0.169  -0.719   0.120        0.596      0.236
variableparentpush:gradec:gendermale  0.546   0.119  -0.711       -0.120     -0.762       -0.166
variablepeerpush:gradec:gendermale    0.108   0.593  -0.120       -0.715     -0.151       -0.825      0.168

Standardized Within-Group Residuals:
       Min         Q1        Med          Q3         Max
-2.7563962 -0.6234359 -0.1362312  0.5886060  3.7951317

Number of Observations: 41280
Number of Groups: 5944
```

Note that the script is slightly different in this example such that the `variable` variable is used instead of the `constparent` and `constpeer` variables. This `variable` variable is in the factor format. If a factor is used in a formula without intercept (e.g., `0 + variable`), the dummy variables representing each group are created (e.g., two dummy variables are created for a two-category factor). Thus, the code is equivalent to specifying the `constparent` and `constpeer` variables separately.[24] The mapping from the formula and reduced-form equation would be

```
formula = value ~ 0 + variable + variable:gradec
          + variable:gradec:gender + variable:gender:gradec
random = ~ 0 + variable + variable:gradec | caseid

formula = value ~ 0 + constparent + constparent:gradec
          + constparent:gradec:gender + constparent:gender:gradec
          + constpeer + constpeer:gradec
          + constpeer:gradec:gender + constpeer:gender:gradec
random = ~ 0 + constparent + constparent:gradec
          + constpeer + constpeer:gradec | caseid
```

$$Y_{ij} = \delta_A\gamma_{00} + \delta_A\gamma_{01}W_j + \delta_A\gamma_{10}(t_{ijk}-7) + \delta_A\gamma_{11}W_j(t_{ijk}-7) + \delta_B\gamma_{20} + \delta_B\gamma_{21}W_j + \delta_B\gamma_{30}(t_{ijk}-7) + \delta_B\gamma_{31}W_j(t_{ijk}-7)$$ Fixed Effect

$$+u_{0j}\delta_A + u_{1j}\delta_A(t_{ijk}-7) + u_{2j}\delta_B + u_{3j}\delta_B(t_{ijk}-7) + e_{ij}$$ Random Effect

The gender differences on the rates of change were not significant for parent encouragement, $z = 1.02$, $p = .31$, and for peer encouragement, $z = -1.30$, $p = .19$. The gender difference on the initial statuses of parent encouragement was not significant, z = .82, $p = .41$. However, male students had a significantly higher peer encouragement than female students by 0.1 point, $z = 3.02$, $p = .003$. If the interaction was significant, the `rockchalkMultilevel` package could not be used here. The package has not support multivariate model yet. Users are encouraged to use the online applet or centering to probe interactions.

Multiple Group Analysis

If a predictor is categorical variable, we can transform the categorical variable as dummy variables and put it as a predictor. By this method, both L1 and L2 variances are assumed to be equal across groups. We have discussed this approach in Model 2. This section will discuss about an alternative method that allows different groups to have different residual variances at both levels.[25] This method is similar to the trick used in analyzing multivariate model. Note that this approach is applicable for the categorical variable at the highest level (e.g., L2 for two-level MLM) only.

Model 29: Multiple-Group Null Model

The math achievement scores are predicted by gender. Gender can be used as a predictor directly (see Model 2). In this example, gender will be treated as multiple groups. L1 and L2 variances will be varied across groups. The multiple-group null model would be

L1

$$W_j = \begin{cases}0\\1\end{cases} \quad then \quad Y_{ij} = \begin{cases}\beta_{0j} + e_{Fij}\\ \beta_{2j} + e_{Mij}\end{cases} \qquad \begin{bmatrix}e_{Fij}\\ e_{Mij}\end{bmatrix} \sim N\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\sigma_F^2 & \\ 0 & \sigma_M^2\end{bmatrix}\right)$$

L2

$$\beta_{0j} = \gamma_{00} + u_{0j} \qquad \begin{bmatrix}u_{0j}\\ u_{2j}\end{bmatrix} \sim N\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\tau_{00} & \\ 0 & \tau_{22}\end{bmatrix}\right)$$
$$\beta_{2j} = \gamma_{20} + u_{2j}$$

---

[24] I have tried specifying the model by the `constparent` and `constpeer` variables. The computer was out of memory. I believe that the `lme` function handles the memory more efficiently when a variable with the factor format is specified than when dummy variables are used.

[25] The approach discussed in this section is analogous to multiple-group approach in SEM. The dummy variable approach is analogous to using a dummy variable as a covariate in SEM.

These notations represent

- Y_{ij} = The math achievement score of Measurement i in Student j
- W_j = The gender of Student j (1 = Male, 0 = Female)
- β_{0j} = The average of math achievement scores of Student j (who is female)
- β_{2j} = The average of math achievement scores of Student j (who is male)
- γ_{00} = The average of math achievement scores across female students
- γ_{20} = The average of math achievement scores across male students
- e_{Fij} = The deviation of the math achievement scores of Measurement i in Student j from the Student j (who is female) average
- e_{Mij} = The deviation of the math achievement scores of Measurement i in Student j from the Student j (who is male) average
- u_{0j} = The deviation of the math achievement scores of Student j from the average across female students
- u_{2j} = The deviation of the math achievement scores of Student j from the average across male students
- σ_A^2 = The math achievement scores variance within female students (L1 variance)
- σ_B^2 = The math achievement scores variance within male students (L1 variance)
- τ_{00} = The math achievement scores variance across female students (L2 variance)
- τ_{22} = The math achievement scores variance across male students (L2 variance)

Note that the covariances across genders are not defined because they are independent cases. That is, $\sigma_{MF} = 0$ and $\tau_{20} = 0$.

The formulas listed above with two dependent variables can be condensed into a formula with one dependent variable with the help of dummy variables:

L1 $$Y_{ij} = \delta_F(\beta_{0j}) + \delta_M(\beta_{2j}) + e_{ij} \qquad e_{ij} \sim N(0, \sigma^2\ )$$

where

- $\delta_F = 1 - W_j$ = A dummy variable where 1 represents females and 0 represents males
- $\delta_M = W_j$ = A dummy variable where 1 represents males and 0 represents females
- e_{ij} = The L1 residuals regardless of gender
- σ^2 = The variance within students regardless of gender, which the error structure is based on gender

The reduced form of this formula would be

$$Y_{ij} = \delta_F\gamma_{00} + \delta_M\gamma_{20} + u_{0j}\delta_F + u_{2j}\delta_M + e_{ij}$$

Notice that this model does not have the intercept term (no fixed effect multiplied by just 1). All regression coefficients are multiplied by dummy variables.

There are several data processing steps before fitting the model. First, the target data set is loaded. The `grade` variable needs to be centered at Grade 7.

```
long <- read.csv("mathgrowth.csv", header = TRUE, na.strings="-999999")

long$gradec <- long$grade - 7
```

The dummy variables representing each gender are also needed.

```
long$female <- as.numeric(long$gender == 1)

long$male <- as.numeric(long$gender == 2)
```

Because L1 residual have different variances for different genders, the `lme4` package cannot be used. The `lme` function from the `nlme` package is used instead:

```
m29 <- lme(mathach ~ 0 + female + male, data = long,

        random = list(caseid=pdBlocked(list(pdSymm(form = ~0 + female),

        pdSymm(form = ~0 + male)))),

        weights = varIdent(form = ~1 | gender), method = "ML", na.action = "na.omit")

summary(m29)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: long
       AIC      BIC    logLik
  142859.1 142906.2 -71423.54

Random effects:
 Composite Structure: Blocked

 Block 1: female
 Formula: ~0 + female | caseid
          female
StdDev: 10.75097

 Block 2: male
 Formula: ~0 + male | caseid
            male Residual
StdDev: 12.92212 7.916055

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | gender
 Parameter estimates:
        2         1
1.0000000 0.8835357
Fixed effects: mathach ~ 0 + female + male
          Value Std.Error   DF  t-value p-value
female 60.07720 0.2161749 5856 277.9101       0
male   59.69612 0.2539496 5856 235.0708       0
 Correlation:
     female
male 0

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-4.99440433 -0.54381884  0.01474496  0.56246627  3.88701374

Number of Observations: 19041
Number of Groups: 5858
```

The formula is similar to Model 26 such that intercepts are fixed to 0 and two dummy variables (female and male) are included as predictors to represent γ_{00} and γ_{20}. In this model, the `random` argument is quite hard to specify. Here is the target matrix of the covariances of random effects:

$$\begin{bmatrix} \tau_{00} & \\ 0 & \tau_{22} \end{bmatrix}$$

Let's think that the covariance matrix consists of two blocks: specification for **females** and **males**. Any covariances across blocks are fixed to 0 (male scores cannot be related with female scores). In the **blue** block, the random effect can be specified as `pdSymm(form = ~0 + female)`, where `pdSymm` is to create a symmetric matrix of covariances among random components. In the **red** block, the random effect can be specified as `pdSymm(form = ~0 + male)`. Then, these blocks are put into a list, `list(pdSymm(form = ~0 + female), pdSymm(form = ~0 + male))`. The list is put in the `pdBlocked` function to indicate that we have a blocked matrix, `pdBlocked(list(pdSymm(form = ~0 + female), pdSymm(form = ~0 + male)))`. Because we need specify the variable that we use to represent L2 units, the list with a name of L2 ID variable (`caseid`) is used to crop the blocked matrix.

The `correlation` argument is not specified here because there is no correlation between scores from males and females (since there are from separate persons), $\sigma_{MF} = 0$. The `weights` argument is specified such that the L1 residual variances depend on gender, `varIdent(form = ~1 | gender)`. The mapping from the formula and reduced-form equation would be

```
formula = value ~ 0 + female + male
random = list(caseid = pdBlocked(list(
            pdSymm(form = ~0 + female),
            pdSymm(form = ~0 + male)
          )))
```

$$\boldsymbol{Y_{ij}} = \underbrace{\boldsymbol{\delta_F}\gamma_{00} + \boldsymbol{\delta_M}\gamma_{20}}_{\text{Fixed Effect}} + \underbrace{\boldsymbol{u_{0j}}\boldsymbol{\delta_F} + \boldsymbol{u_{2j}}\boldsymbol{\delta_M}}_{\text{Random Effect}} + e_{ij}$$

Readers can see that the L2 standard deviations of math achievements across female and male students are 10.75 and 12.92, respectively. The L1 standard deviations of math achievement for female and male groups are 6.99 (0.88 × 7.92) and 7.92, respectively. From this output, researchers may wish to analyze residual intraclass correlations (accounting for gender):

1. Extract the variances of random effects and residual variance by the `VarCorr` function:

```
allvar <- VarCorr(m29)

allvar
```

```
caseid = pdSymm(0 + female), pdSymm(0 + male)
         Variance  StdDev
female   115.58343 10.750974
male     166.98114 12.922118
Residual  62.66393  7.916055
```

2. Get the L1 residual standard deviation.

```
malesd <- as.numeric(allvar[3, 2])
```

The `as.numeric` function is needed to transform the text format to the number format. This residual standard deviation is among male participants. If you see the output from `summary(m29)`, the residual variance is listed under `Block 2: male`.

3. Compute the L1 residual standard deviation of all groups:

```
l1sd <- c(malesd * coef(m29$modelStruct$varStruct, unconstrained = FALSE), malesd)
```

The code used to find the ratio of standard deviations between the first and the rest of dependent variables are similar to the codes in Model 19. I put the `malesd` as the last element according to the order of random effects from the result of the `VarCorr` function (`female` and `male`).

4. Compute the L1 residual variances of all variables:

```
l1var <- l1sd^2
```

5. Get the variances at L2 of all variables (the order will be `female` and `male`):

```
l2var <- as.numeric(allvar[1:2, 1])
```

6. Compute residual intraclass correlations of all groups:

```
icc <- l2var / (l1var + l2var)

icc
```

```
        1
0.7026301 0.7271271
```

The residual intraclass correlations of math achievement for females and males are .70 and .73, respectively.

Furthermore, we can compare whether the averages of math achievement between female and male students are different. The comparison can be done by the `glht` function in the `multcomp` package:

```
library(multcomp)

ctr <- glht(m29, "female - male = 0")

summary(ctr)
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: lme.formula(fixed = mathach ~ 0 + female + male, data = long,
    random = list(caseid = pdBlocked(list(pdSymm(form = ~0 +
        female), pdSymm(form = ~0 + male)))), weights = varIdent(form = ~1 |
        gender), method = "ML", na.action = "na.omit")

Linear Hypotheses:
                   Estimate Std. Error z value Pr(>|z|)
female - male == 0   0.3811     0.3335   1.143    0.253
(Adjusted p values reported -- single-step method)
```

On average, male and female students were not significantly different in math achievement, $z = 1.14$, $p = .25$. Readers are encouraged to run the model where gender is used as a predictor (similar to Model 2). Readers may compare the results of gender differences, residual variances at both levels, and residual intraclass correlations between the current model and the model with gender as a predictor.

Model 30: Multiple-Group Model of Linear Trajectories

The linear trajectory of math achievement is predicted by gender. This model is similar to Model 17. However, we will use the multiple-group framework to analyze this data. The multiple-group model of linear trajectories would be

L1 $$W_j = \begin{cases} 0 \\ 1 \end{cases} \quad then \quad Y_{ij} = \begin{cases} \beta_{0j} + \beta_{1j}\left(t_{ij} - 7\right) + e_{Fij} \\ \beta_{2j} + \beta_{3j}\left(t_{ij} - 7\right) + e_{Mij} \end{cases} \qquad \begin{bmatrix} e_{Fij} \\ e_{Mij} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_F^2 & \\ 0 & \sigma_M^2 \end{bmatrix}\right)$$

L2

$$\beta_{0j} = \gamma_{00} + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + u_{1j}$$
$$\beta_{2j} = \gamma_{20} + u_{2j}$$
$$\beta_{3j} = \gamma_{30} + u_{3j}$$

$$\begin{bmatrix} u_{0j} \\ u_{1j} \\ u_{2j} \\ u_{3j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & & & \\ \tau_{10} & \tau_{11} & & \\ 0 & 0 & \tau_{22} & \\ 0 & 0 & \tau_{32} & \tau_{33} \end{bmatrix}\right)$$

These notations represent (the blue lines indicate that the meanings changed from Model 29)

- Y_{ij} = The math achievement score of Measurement i in Student j
- W_j = The gender of Student j (1 = Male, 0 = Female)
- t_{ij} = The grade that the Measurement i in Student j was observed
- β_{0j} = The expected math achievement scores of Student j (who is female) at Grade 7
- β_{1j} = The expected change in math achievement when grade increases by 1 for Student j (who is female), which is the rate of change in math achievement for female Student j
- β_{2j} = The expected math achievement scores of Student j (who is male) at Grade 7
- β_{3j} = The expected change in math achievement when grade increases by 1 for Student j (who is male), which is the rate of change in math achievement for male Student j
- γ_{00} = The average of math achievement in Grade 7 across female students
- γ_{10} = The average rate of change in math achievement across female students
- γ_{20} = The average of math achievement in Grade 7 across male students
- γ_{30} = The average rate of change in math achievement across male students
- e_{Fij} = The difference between the actual math achievement score of Measurement i in Student j and the expected score of Student j (who is female) at a given grade level
- e_{Mij} = The difference between the actual math achievement score of Measurement i in Student j and the expected score of Student j (who is male) at a given grade level
- u_{0j} = The deviation of the actual math achievement score of Student j at Grade 7 from the average math achievement score at Grade 7 across female students
- u_{1j} = The deviation of the rate of change in math achievement of Student j from the average rate of change across female students
- u_{2j} = The deviation of the actual math achievement score of Student j at Grade 7 from the average math achievement score at Grade 7 across male students
- u_{3j} = The deviation of the rate of change in math achievement of Student j from the average rate of change across male students
- σ_F^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade for females
- σ_M^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade for males
- τ_{00} = The variance of math achievement scores at Grade 7 across female students
- τ_{11} = The variance of the rate of change in math achievement score across female students
- τ_{22} = The variance of math achievement scores at Grade 7 across male students
- τ_{33} = The variance of the rate of change in math achievement score across male students
- τ_{10} = The covariance between the math achievement score at Grade 7 and the rate of change in math achievement score for females

- τ_{32} = The covariance between the math achievement score at Grade 7 and the rate of change in math achievement score for males
- $\rho_{st} = \tau_{st}/\sqrt{\tau_{ss}\tau_{tt}}$ (where s, t = 0, 1, 2, or 3 and $s \neq t$) = The covariance mentioned above in the correlation scale (from -1 to 1)

The formulas listed above with two dependent variables can be condensed into a formula with one dependent variable with the help of dummy variables:

L1 $$\tilde{Y}_{ij} = \delta_F\left(\beta_{0j} + \beta_{1j}\left(t_{ijk} - 7\right)\right) + \delta_M\left(\beta_{2j} + \beta_{3j}\left(t_{ijk} - 7\right)\right) + \tilde{e}_{ij} \qquad \tilde{e}_{ij} \sim N(0, \tilde{\sigma}^2)$$

where the notations were defined in Model 29.

The model can be analyzed by the `lme` function from the `nlme` package:

```
m30 <- lme(mathach ~ 0 + female + female:gradec + male + male:gradec, data = long,

        random=list(caseid=pdBlocked(list(pdSymm(form = ~0 + female + female:gradec),

        pdSymm(form = ~0 + male + male:gradec)))),

        weights = varIdent(form = ~1 | gender), method = "ML", na.action = "na.omit",

        control = lmeControl(maxIter = 500, msMaxIter = 500, niterEM = 100, msMaxEval = 400))

summary(m30)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: long
       AIC      BIC    logLik
  130484.7 130578.9 -65230.34

Random effects:
 Composite Structure: Blocked

 Block 1: female, female:gradec
 Formula: ~0 + female + female:gradec | caseid
 Structure: General positive-definite
              StdDev   Corr
female        8.825821 female
female:gradec 1.450298 0.276

 Block 2: male, male:gradec
 Formula: ~0 + male + male:gradec | caseid
 Structure: General positive-definite
            StdDev    Corr
male        10.089393 male
male:gradec  1.623012 0.363
Residual     4.946354

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | gender
 Parameter estimates:
        2         1
1.0000000 0.8279436
Fixed effects: mathach ~ 0 + female + female:gradec + male + male:gradec
                 Value  Std.Error    DF   t-value p-value
female        51.32906 0.19871755  5856 258.30161       0
male          50.30066 0.22367698  5856 224.88081       0
female:gradec  3.22685 0.04606491 13182  70.05015       0
gradec:male    3.55499 0.05320433 13182  66.81774       0
 Correlation:
              female male   fml:gr
male           0.000
female:gradec -0.262  0.000
gradec:male    0.000 -0.233  0.000

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
```

```
-5.47211081 -0.47768057  0.01952874  0.50429261  3.84441936

Number of Observations: 19041
Number of Groups: 5858
```

The script is similar to Model 29. The differences are in the `formula` and the `random` arguments. The `gradec` variable is added to those formulas by the `female:gradec` and `male:gradec` terms, corresponding to $\delta_F\gamma_{10}$ and $\delta_M\gamma_{30}$. The `:` operator is to create interaction. The `*` operator cannot be used here because the lower-order terms must not be included.

Again, the `random` argument is quite hard to specify. Here is the target matrix of the covariances of random effects:

$$\begin{bmatrix} \tau_{00} & & & \\ \tau_{10} & \tau_{11} & & \\ 0 & 0 & \tau_{22} & \\ 0 & 0 & \tau_{32} & \tau_{33} \end{bmatrix}$$

Let's think that the covariance matrix consists of two blocks: specifications for **females** and **males**, which is similar to Model 29. In the **blue** block, the random effect is specified as `pdSymm(form = ~0 + female + female:gradec)`, which is translated to the variance of `female` (τ_{00}), the variance of `female:gradec` (τ_{11}), and covariance between `female` and `female:gradec` (τ_{10}). In the **red** block, the random effect is specified as `pdSymm(form = ~0 + male + male:gradec)`. These blocks are combined with a similar method to Model 29.

The `control` argument is specified here because the default number of iterations is not enough to get a convergent result. Please see the details in the help page of the `lmeControl` function (type `?lmeControl`).

The mapping from the formula and reduced-form equation would be

```
formula = value ~ 0 + female + female:gradec
            + male + male:gradec
random = list(caseid = pdBlocked(list(
            pdSymm(form = ~0 + female + female: gradec),
            pdSymm(form = ~0 + male + male: gradec)
          )))
```

$Y_{ij} = \delta_F\gamma_{00} + \delta_F\gamma_{10}(t_{ijk} - 7) + \delta_M\gamma_{20} + \delta_M\gamma_{30}(t_{ijk} - 7)$	Fixed Effect
$+u_{0j}\delta_F + u_{1j}\delta_F(t_{ijk} - 7) + u_{2j}\delta_M + u_{3j}\delta_M(t_{ijk} - 7) + e_{ij}$	Random Effect

The average rates of change in math achievement were significant for males, $z = 70.05$, $p < .001$, and females, $z = 66.82$, $p < .001$.

The initial status and the rate of growth can be compared across gender. The `glht` function from the `multcomp` package can be used:

```
library(multcomp)

ctr <- glht(m30, c("female - male = 0", "female:gradec - gradec:male = 0"))

summary(ctr)
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: lme.formula(fixed = mathach ~ 0 + female + female:gradec + male +
    male:gradec, data = long, random = list(caseid = pdBlocked(list(pdSymm(form = ~0 +
    female + female:gradec), pdSymm(form = ~0 + male + male:gradec)))),
    weights = varIdent(form = ~1 | gender), method = "ML", na.action = "na.omit",
    control = lmeControl(maxIter = 500, msMaxIter = 500, niterEM = 100,
        msMaxEval = 400))

Linear Hypotheses:
                                 Estimate Std. Error z value Pr(>|z|)
female - male == 0                1.02840    0.29917   3.438  0.00117 **
female:gradec - gradec:male == 0 -0.32814    0.07037  -4.663 6.23e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

Female students had a significant higher initial status than male students, $z = 3.44$, $p = .001$, but female students had a significant lower rate of growth than male students, $z = -4.66$, $p < .001$. In the output of this model, the initial statuses and the rates of change of each gender were already provided so probing interaction is not needed. Readers are encouraged to compare the result of this model with the result from Model 17.

Model 31: Multiple-Group Model of Linear Trajectories with Time-Invariant Covariate

The linear trajectory of math achievement is predicted by gender, which we will analyze by multiple-group framework in this section (similar to Model 30). In this example, attitude toward math, which is student-level predictor, is included in the model. Attitude toward math (with grand-mean centering) is used to predict both initial status and rate of change of math achievement. The multiple-group model of linear trajectories with time-invariance covariates would be

L1
$$W_j = \begin{cases} 0 \\ 1 \end{cases} \quad then \quad Y_{ij} = \begin{cases} \beta_{0j} + \beta_{1j}\left(t_{ij} - 7\right) + e_{Fij} \\ \beta_{2j} + \beta_{3j}\left(t_{ij} - 7\right) + e_{Mij} \end{cases} \qquad \begin{bmatrix} e_{Fij} \\ e_{Mij} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_F^2 & \\ 0 & \sigma_M^2 \end{bmatrix} \right)$$

L2
$$\begin{aligned} \beta_{0j} &= \gamma_{00} + \gamma_{01}\left(Z_j - \bar{Z}\right) + u_{0j} \\ \beta_{1j} &= \gamma_{10} + \gamma_{11}\left(Z_j - \bar{Z}\right) + u_{1j} \\ \beta_{2j} &= \gamma_{20} + \gamma_{21}\left(Z_j - \bar{Z}\right) + u_{2j} \\ \beta_{3j} &= \gamma_{30} + \gamma_{31}\left(Z_j - \bar{Z}\right) + u_{3j} \end{aligned} \qquad \begin{bmatrix} u_{0j} \\ u_{1j} \\ u_{2j} \\ u_{3j} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau_{00} & & & \\ \tau_{10} & \tau_{11} & & \\ 0 & 0 & \tau_{22} & \\ 0 & 0 & \tau_{32} & \tau_{33} \end{bmatrix} \right)$$

These notations represent (the blue lines indicate that the meanings changed from Model 30)

- Y_{ij} = The math achievement score of Measurement *i* in Student *j*
- W_j = The gender of Student *j* (1 = Male, 0 = Female)
- Z_j = The attitude toward math of Student *j*
- t_{ij} = The grade that the Measurement *i* in Student *j* was observed
- β_{0j} = The expected math achievement scores of Student *j* (who is female) at Grade 7
- β_{1j} = The expected change in math achievement when grade increases by 1 for Student *j* (who is female), which is the rate of change in math achievement for female Student *j*
- β_{2j} = The expected math achievement scores of Student *j* (who is male) at Grade 7
- β_{3j} = The expected change in math achievement when grade increases by 1 for Student *j* (who is male), which is the rate of change in math achievement for male Student *j*
- γ_{00} = The average of math achievement in Grade 7 across female students given that the attitude toward math is equal to its grand mean

- γ_{01} = The change in the average of math achievement in Grade 7 for female students if the attitude toward math increases by 1
- γ_{10} = The average rate of change in math achievement across female students given that the attitude toward math is equal to its grand mean
- γ_{11} = The change in the average rate of change in math achievement for female students if the attitude toward math increases by 1
- γ_{20} = The average of math achievement in Grade 7 across male students given that the attitude toward math is equal to its grand mean
- γ_{21} = The change in the average of math achievement in Grade 7 for male students if the attitude toward math increases by 1
- γ_{30} = The average rate of change in math achievement across male students given that the attitude toward math is equal to its grand mean
- γ_{31} = The change in the average rate of change in math achievement for male students if the attitude toward math increases by 1
- e_{Fij} = The difference between the actual math achievement score of Measurement *i* in Student *j* and the expected score of Student *j* (who is female) at a given grade level
- e_{Mij} = The difference between the actual math achievement score of Measurement *i* in Student *j* and the expected score of Student *j* (who is male) at a given grade level
- u_{0j} = The deviation of the actual math achievement score of Student *j* at Grade 7 from the expected math achievement score at Grade 7 across female students (given the attitude toward math)
- u_{1j} = The deviation of the rate of change in math achievement of Student *j* from the expected rate of change across female students (given the attitude toward math)
- u_{2j} = The deviation of the actual math achievement score of Student *j* at Grade 7 from the expected math achievement score at Grade 7 across male students (given the attitude toward math)
- u_{3j} = The deviation of the rate of change in math achievement of Student *j* from the expected rate of change across male students (given the attitude toward math)
- σ_F^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade for females
- σ_M^2 = The math achievement score residual variance within the measurement level (L1 residual variance) controlling for grade for males
- τ_{00} = The residual variance of math achievement scores at Grade 7 across female students controlling for attitude toward math
- τ_{11} = The residual variance of the rate of change in math achievement score across female students controlling for attitude toward math
- τ_{22} = The residual variance of math achievement scores at Grade 7 across male students controlling for attitude toward math
- τ_{33} = The residual variance of the rate of change in math achievement score across male students controlling for attitude toward math
- τ_{10} = The residual covariance between the math achievement score at Grade 7 and the rate of change in math achievement score in females controlling for attitude toward math

- τ_{32} = The residual covariance between the math achievement score at Grade 7 and the rate of change in math achievement score in males controlling for attitude toward math
- $\rho_{st} = \tau_{st}/\sqrt{\tau_{ss}\tau_{tt}}$ (where $s, t = 0$, 1, 2, or 3 and $s \neq t$) = The covariance mentioned above in the correlation scale (from -1 to 1)

The formulas listed above with two dependent variables can be condensed into a formula with one dependent variable with the help of dummy variables:

L1 $$\tilde{Y}_{ij} = \delta_A\left(\beta_{0j} + \beta_{1j}(t_{ijk} - 7)\right) + \delta_B\left(\beta_{2j} + \beta_{3j}(t_{ijk} - 7)\right) + \tilde{e}_{ij} \qquad \tilde{e}_{ij} \sim N(0, \tilde{\sigma}^2)$$

where the notations were defined in Model 29.

The model can be analyzed by the `lme` function from the `nlme` package:

```
m31 <- lme(mathach ~ 0 + female + female:gradec + female:likemathC + female:gradec:likemathC

        + male + male:gradec + male:likemathC + male:gradec:likemathC, data = long,

        random=list(caseid=pdBlocked(list(pdSymm(form = ~0 + female + female:gradec),

        pdSymm(form = ~0 + male + male:gradec)))),

        weights = varIdent(form = ~1 | gender), method = "ML", na.action = "na.omit",

        control = lmeControl(maxIter = 500, msMaxIter = 500, niterEM = 100, msMaxEval = 400))

summary(m31)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: long
       AIC      BIC    logLik
  130245.8 130371.4 -65106.89

Random effects:
 Composite Structure: Blocked

 Block 1: female, female:gradec
 Formula: ~0 + female + female:gradec | caseid
 Structure: General positive-definite
              StdDev   Corr
female        8.761090 female
female:gradec 1.431502 0.262

 Block 2: male, male:gradec
 Formula: ~0 + male + male:gradec | caseid
 Structure: General positive-definite
            StdDev    Corr
male        10.008505 male
male:gradec  1.593101 0.347
Residual     4.948018

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | gender
 Parameter estimates:
       2        1
1.000000 0.828015
Fixed effects: mathach ~ 0 + female + female:gradec + female:likemathC + female:gradec:likemathC +      male +
male:gradec + male:likemathC + male:gradec:likemathC
                           Value  Std.Error    DF   t-value p-value
female                  51.40761 0.19809139  5843 259.51461       0
male                    50.22549 0.22351079  5843 224.71171       0
female:gradec            3.24632 0.04591228 13179  70.70696       0
female:likemathC         0.85452 0.15529674  5843   5.50248       0
gradec:male              3.53531 0.05311606 13179  66.55819       0
likemathC:male           1.01004 0.19166397  5843   5.26984       0
female:gradec:likemathC  0.15914 0.03596786 13179   4.42460       0
gradec:likemathC:male    0.25277 0.04620519 13179   5.47069       0
 Correlation:
```

```
                          female male   fml:gr fml:lC grdc:m lkmtC: fml::C
male                       0.000                                          
female:gradec             -0.272  0.000                                   
female:likemathC           0.049  0.000 -0.001                            
gradec:male                0.000 -0.248  0.000  0.000                     
likemathC:male             0.000 -0.087  0.000  0.000  0.034              
female:gradec:likemathC -0.001  0.000  0.057 -0.286  0.000  0.000       
gradec:likemathC:male      0.000  0.033  0.000  0.000 -0.100 -0.273  0.000

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max 
-5.46984210 -0.47714938  0.01898742  0.50460806  3.84992045 

Number of Observations: 19029
Number of Groups: 5847 
```

The script is similar to Model 30. The only difference is in the `formula` argument such that `likemathC` is added in the model. The `female:gradec:likemathC` and `male:gradec:likemathC` terms represent the interaction effects (rate of growth predicted by attitude toward math) and the `female:gradec` and `male:gradec` terms represent the direct effect to math achievement of each group (initial status predicted by attitude toward math). The `random` argument remains the same as one in Model 30 because the `likemathC` variable is the student-level predictor.

The mapping from the formula and reduced-form equation would be

```
formula = value ~ 0 + female + female:gradec + female:likemathC
                 + female:gradec:likemathC
                 + male + male:gradec + male:likemathC
                 + male:gradec:likemathC
random = list(caseid = pdBlocked(list(
                 pdSymm(form = ~0 + female + female: gradec),
                 pdSymm(form = ~0 + male + male: gradec)
               )))
```

$$Y_{ij} = \delta_F\gamma_{00} + \delta_F\gamma_{10}(t_{ijk} - 7) + \delta_F\gamma_{01}(Z_j - \bar{Z}) + \delta_F\gamma_{11}(t_{ijk} - 7)(Z_j - \bar{Z})$$ Fixed Effect

$$+\delta_M\gamma_{20} + \delta_M\gamma_{30}(t_{ijk} - 7) + \delta_M\gamma_{21}(Z_j - \bar{Z}) + \delta_M\gamma_{31}(t_{ijk} - 7)(Z_j - \bar{Z})$$

$$+u_{0j}\delta_F + u_{1j}\delta_F(t_{ijk} - 7) + u_{2j}\delta_M + u_{3j}\delta_M(t_{ijk} - 7) + e_{ij}$$ Random Effect

The attitude toward math significantly moderated the change in math achievement across grades in both females, $z = 4.42$, $p < .001$, and males, $z = 5.47$, $p < .001$. The rate of increases in math achievement was higher when the attitude toward math increased. The difference in this moderation effect (i.e., three-way interaction between gender, grade, and attitude toward math) can be tested by the multi-parameter contrast:

```
library(multcomp)

ctr <- glht(m31, "female:gradec:likemathC - gradec:likemathC:male = 0")

summary(ctr)
```

```
         Simultaneous Tests for General Linear Hypotheses

Fit: lme.formula(fixed = mathach ~ 0 + female + female:gradec + female:likemathC + 
    female:gradec:likemathC + male + male:gradec + male:likemathC + 
    male:gradec:likemathC, data = long, random = list(caseid = pdBlocked(list(pdSymm(form = ~0 + 
    female + female:gradec), pdSymm(form = ~0 + male + male:gradec)))), 
    weights = varIdent(form = ~1 | gender), method = "ML", na.action = "na.omit", 
    control = lmeControl(maxIter = 500, msMaxIter = 500, niterEM = 100, 
        msMaxEval = 400))

Linear Hypotheses:
                                                     Estimate Std. Error z value Pr(>|z|)
female:gradec:likemathC - gradec:likemathC:male == 0 -0.09363    0.05854  -1.599     0.11
```

```
(Adjusted p values reported -- single-step method)
```

The difference between the moderation effect of attitude toward math across genders was not significant, z = -1.60, p = .11.

Provide Feedback

This article is used for lab sections in the Multilevel Modeling class, University of Kansas. If you find any errors or give suggestions, please let me know at

Sunthud Pornprasertmanit
Department of Psychology and Center for Research Methods and Data Analysis
University of Kansas
Email: psunthud@ku.edu

References

Bates, D., Maechler, M., & Bolker, B. (2012). *lme4: Linear mixed-effects models using S4 classes* [Computer Software]. R package version 0.999999-0. Available at the Comprehensive R Archive Network.

Bauer, D. J., & Curran, P. J. (2005). Probing interactions in fixed and multilevel regression: Inferential and graphical techniques. *Multivariate Behavioral Research*, *40*, 373-400.

Cheung, M. W. L. (2009). Constructing approximate confidence intervals for parameters with structural equation models. *Structural Equation Modeling*, *16*, 267-294.

De Rosario-Martinez, H. (2012). *phia: Post-hoc interaction analysis* [Computer Software]. R package version 0.1-0. Available at the Comprehensive R Archive Network.

Enders, C. K., & Tofighi, D. (2007). Centering predictor variables in cross-sectional multilevel models: a new look at an old issue. *Psychological methods*, *12*, 121-138.

Graham, J. W. (2009). Missing data analysis: Making it work in the real world. *Annual review of psychology*, *60*, 549-576.

Hothorn, T., Bretz, F., & Westfall, P. (2008). Simultaneous inference in general parametric models. *Biometrical Journal, 50*, 346-363.

Howell, D. C. (2007). *Statistical methods for psychology* (6th ed.). Belmont, CA: Thomson Wadsworth.

Hox, J. J. (2010). *Multilevel analysis: Techniques and applications* (2nd ed.). New York: Routledge.

Johnson, P. E. (2012). *rockchalk: Regression estimation and presentation* [Computer Software]. R package version 1.6.2. Available at the Comprehensive R Archive Network.

Li, K.H., Meng, X.-L., Raghunathan, T.E. and Rubin, D.B. (1991). Significance levels from repeated p-values with multiply-imputed Data. *Statistica Sinica, 1*, 65-92.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., & R Development Core Team. (2013). *nlme: Linear and nonlinear mixed effects models* [Computer Software]. R package version 3.1-108. Available at the Comprehensive R Archive Network.

Pornprasertmanit, S. (2013). *rockchalkMultilevel: Multilevel regression estimation and presentation* [Computer Software]. R package version 0.0-1. Available at http://rweb.quant.ku.edu/kran

Preacher, K. J., Curran, P. J., & Bauer, D. J. (2006). Computational tools for probing interaction effects in multiple linear regression, multilevel modeling, and latent curve analysis. *Journal of Educational and Behavioral Statistics*, *31*, 437-448.

Snijders, T. A. A., & Bosker, R. J. (2011). *Multilevel analysis: An introduction to basic and advanced multilevel modeling* (2nd ed.). Thousand Oaks, CA: Sage.

Raudenbush, S. W., & Byrk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods* (2nd ed.). Thousand Oaks, CA: Sage.

Van Buuren, S., Groothuis-Oudshoorn, K. (2011). `mice`: Multivariate imputation by chained equations in `R`. *Journal of Statistical Software*, **45**(3), 1-67.

Vrieze, S. I. (2012). Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychological methods*, *17*, 228-243.