GETTING STARTED

Hierarchical Linear Modeling: A Step by Step Guide

Utilize R for your mixed model analysis

Kay Chansiri Jan 4, 2021 · 15 min read

In most cases, data tends to be clustered. Hierarchical Linear Modeling (HLM) enables you to explore and understand your data and decreases Type I error rates. This tutorial uses R to demonstrate the basic steps of HLM in social science research.

Credit: https://pixabay.com/users/geralt-9301/ from Pixabay

Before beginning the analysis, let's briefly talk about what is HLM first.

HLM (AKA multilevel modeling) analyzes data that is clustered in an organized pattern(s), such as universities in states, non-white males in tech companies, and clinics in hospitals. HLM is an ordinary least square (OLS) that requires all assumptions met (check out my tutorial for OLS assumption and data screening) except the independence of errors assumption. The assumption is likely violated as HLM allows data across clusters to be correlated.

Predictors in HLM can be categorized into random and fixed effects. Random effects refer to variables that are not the main focus of a study but may impact the dependent variable and therefore needed to be included in the model. Fixed effects, on the other hand, are key predictors of the study. For example, a psychologist wants to predict the impact of adverse childhood trauma on one's tendency to develop borderline personality disorder (BPD) in adulthood. Participants are from collectivist and individualistic cultures, and both cultures likely define parents' behaviors differently. People in individualistic cultures, such as those in America or the U.K., probably consider parents' spanking abusive, whereas collectivist individuals, such as Asians and Africans, may consider spanking as a way to enhance a child's discipline. Thus, participants from different cultures may be variously impacted by the same behavior from their parents during childhood and may develop BPD symptoms at a different level. According to the example, childhood trauma is treated as the fixed effects based on personality literature and the researcher's interest as a psychologist. Cultures could be treated as random effects as the variable potentially impact borderline personality development but not the main focus of the study. It is noteworthy that random effects should be categorical, whereas fixed effects could be dummy variables (a categorical variable with two levels) or continuous variables.

Some of you may think, *why don't we use a single level regression model and control for potential random effects (e.g., cultures according to the mentioned example)?* Doing so may introduce wrong standard error estimates as residuals (i.e., observations in the same group) tend to be correlated. For instance, people from the same culture may view a behavior in the same way. A single-level model's error term represents clustered data errors across levels, limiting us from knowing how much effects that the key

predictor (e.g., childhood trauma) has on one's tendency to develop BPD after controlling for cultures in which participants are nested.

Still confused? Let's look at the equations below:

$yi = \beta 0 + \beta 1xi + ei$ **A single regression model — (1)**

$yij = \beta 0 + uj + eij$ **A variance components model — (2)**

$yij = \beta 0 + \beta 1xij + uj + eij$ **A mixed model (with random intercepts) — (3)**

i is the number of observation (e.g., participant #1, #2, #3..).

j is the category of culture that each observation belongs to (i.e., j = 1 is collectivism, and j = 0 is individualism).

$\beta 1$ is adverse childhood trauma

y is BPD tendency.

u is variance in y that is not explained by cultures, controlling for other predictors.

e is variance in y that is not explained by childhood trauma controlling for other predictors.

According to equation 1, the error term (*ei*) indicates an unexplained variance of the outcome that is not accounted for by the key independent variable (e.g., childhood trauma). Equation 2 shows two error terms, including the error term of the random effects (*uj*) (i.e., cultures) and the error term of the fixed effects nested in the random effects (*eij*) (childhood trauma scores in different cultures). Equation 3 represents a mixed model that integrates equations 1 and 2, accounting for more accurate error estimates relative to the single-level regression model in equation 1.

Now that you have some foundation of HLM let's see what you need before the analysis.

- **Data in a long format:** Data is typically structured in a wide format (i.e., each column represents one variable, and each row depicts one observation). You need to convert data into a long format (i.e., a case's data is distributed across rows. One column describes variable types, and another column contains values of those variables). Check out this tutorial for how to reshape data from a wide to long format.

- **R packages**: nlme for linear and non-linear model testing

```
install.packages("nlme")
library(nlme)
```

Case Study

A fictional data set is used for this tutorial. We will look at whether one's narcissism predicts their intimate relationship satisfaction, assuming that narcissistic symptoms (e.g., self absorb, lying, a lack of empathy) vary across times in which different life events occur. Thus, fixed effects are narcissistic personality disorder symptoms (NPD). The outcome variable is one's intimate relationship satisfaction (Satisfaction). The random effects are Time with three levels coded as 1 (before marriage), 2 (1 year after marriage), and 3 (5 years after marriage).

Pre-Analysis Steps

Step 1: Import data

```
#Set working directory
setwd("insert your file location:")
#import data
library(foreign)
data<-read.spss("HLM.sav(your data name)," use.value.label = TRUE,
to.data.frame = TRUE)
```

Step 2: Data cleaning

This tutorial assumes that your data has been cleaned. Check out my data preparation tutorial if you would like to learn more about cleaning your data. For my current data set, all of the assumptions, except the indepe**n**dence of errors, are met, consistent with the HLM requirement.

HLM Analysis Steps

Step 1:An intercept only model.

An intercept only model is the simplest form of HLM and recommended as the first step before adding any other predictive terms. This type of model testing allows us to understand whether the outcome variable scores (i.e., relationship satisfaction in this

tutorial) are significantly different from zero (i.e., participants have indicated certain relationship satisfaction levels) without considering other predictors. For an OLS model, an intercept is also known as the constant, which in an intercept only model is the mean of the outcome variable, as shown in the below equation:

$$y_i = \beta_0$$

Image by author

We will use the gls function (i.e., generalized least squares) to fit a linear model. The gls function enables errors to be correlated and to have heterogeneous variances, which are likely the case for clustered data. I will identify my intercept only model as 'model1.'

```
model1=gls(Satisfaction~1, data = data, method = "ML," na.action =
"na.omit")
summary(model1)
```

Here are the results:

```
Generalized least squares fit by maximum likelihood
  Model: Satisfaction ~ 1
  Data: data
      AIC      BIC    logLik
  6543.89 6555.678 -3269.945

Coefficients:
               Value  Std.Error t-value p-value
(Intercept) 5.087982 0.01582679 321.479       0

Standardized residuals:
       Min         Q1        Med         Q3        Max
-4.9894040 -0.5142181  0.0960345  0.7644064  1.1131222

Residual standard error: 0.8193328
Degrees of freedom: 2681 total; 2680 residual
```

The p-value is significant, indicating that participants' relationship satisfaction is significantly different from zero.

Step 2: A random intercept model.

This step added my random effects (i.e., Time) to see whether the predictor increases a significant variance explained in my dependent variable relative to the previous intercept only model (Model 1).

Statistically speaking, if you still remember the earlier equations, the intercept for the overall regression of an intercept only model is still $\beta 0$. However, for each group of random effects(i.e., each point of Time after marriage), the intercept is $\beta 0 + uj$ (when uj represents errors of the dependent variable that are not explained by Time).

To test the random intercept model, I will use the lme function as an alternative approach in addition to the mentioned gls function. Like gls, the lme function is used to test a linear mixed-effects model, allowing nested random effects and the correlations among within-group errors. Both lme and gls enable the maximum likelihood application.

Before including Time as random effects, make sure that the variable is categorical:

```
is.factor(data$Time)
[1] FALSE
```

The output says 'false,' so I need to convert Time into a categorical variable.

```
data$Time = as.factor(data$Time)

#Check again whether Time is categorical

is.factor(data$Time)

[1] TRUE
```

Modeling the random intercept:

```
model2 = lme(Satisfaction~1, data = data, method = "ML", na.action =
"na.omit", random = ~1|Time)

summary(model2)
```

The results:

```
Linear mixed-effects model fit by maximum likelihood
 Data: data
       AIC      BIC    logLik
  6533.549 6551.231 -3263.775

Random effects:
 Formula: ~1 | Time
        (Intercept)  Residual
StdDev:  0.06596515 0.8165719

Fixed effects: Satisfaction ~ 1
               Value  Std.Error   DF  t-value p-value
(Intercept) 5.092783 0.04124424 2678 123.4786       0

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-5.0747125 -0.4169725  0.1953434  0.6985522  1.2158700

Number of Observations: 2681
Number of Groups: 3
```

Now, you may wonder how I could know whether my random effects (i.e., Time) are significant. There are a couple of ways to look at this.

1. Compare the AIC of the intercept only model (Model1) and AIC of the random intercept model (Model 2). **AIC = 2k — 2(log-likelihood),** when k is the number of variables in the model including the intercept), and the log-likelihood is a model fit measure, which can be obtained from statistical output. Check out this useful information from Satisticshowto.

From my model 1's and 2's outputs, you will see that model 1's AIC = 6543.89, and Model 2's AIC = 6533.549. Generally, the two **AIC** values that differ more than 2 indicate a significant difference in model fitting. The lower the AIC value is, the better fit a model. You can see that including Time as random effects in Model 2 improves my Model 1 (6543.89 -6533.549 > 2).

2. In addition to AIC, we can compare the intercept only model and the random intercept using the ANOVA function.

```
anova(model1, model2)
```

Here are the results:

```
Model df      AIC      BIC    logLik   Test  L.Ratio p-value
model1     1  2 6543.890 6555.678 -3269.945
model2     2  3 6533.549 6551.231 -3263.775 1 vs 2 12.34079   4e-04
```

The p-value, 4e-04, is equal to 4 x 10^-4, indicating that the results are highly significant. Adding the random intercept thus significantly improves the intercept only model.

In addition to the gls and lme functions from the package nlme, we can use lmer from package lme4. In general, both lme and lmer are effective functions for mixed data analysis with some differences to be considered:

1. lmer does not analyze some correlation structures that lme does.
2. nlme is a larger toolkit and their codes about mixed models are easier to understand.
3. nlme can be used to define cross random effects easier and quicker than lme.
4. Models fitted by the nlme packages (e.g., lme and gls function) and the lme4 package (e.g., the lmer function) assume that the sampling variances are known.

To put it simply, I would say for a simple HLM analysis, both lme4 and nlme should provide close parameter values. You may check out this page for comparisons of the packages.

If you want to try lme4, you need to install merTools first:

```
install.packages("merTools")
library(lme4)
library(merTools)
```

Let's run our random intercept model using lmer from lme4

```
model2.1<-lmer(Satisfaction~1+(1|Time), REML = FALSE, data = data)
summary(model2.1)
```

Results:

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: Satisfaction ~ 1 + (1 | Time)
   Data: data

AIC      BIC   logLik deviance df.resid
  6533.5   6551.2  -3263.8   6527.5     2678

Scaled residuals:
    Min      1Q  Median      3Q     Max
-5.0747 -0.4170  0.1953  0.6986  1.2159

Random effects:
 Groups   Name        Variance Std.Dev.
 Time     (Intercept) 0.004351 0.06597
 Residual             0.666790 0.81657
Number of obs: 2681, groups:  Time, 3

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.09278    0.04124   123.5
```

You can see that the parameters of model 2 (lme4) and model 2.1 (nlme) are quite close.

We can also run an ICC (AKA Intraclass Correlation Coefficient) to see the correlation of observations within groups (i.e., relationship satisfaction within each Time point in my case). The ICC index can range from 0 to 1, with more values indicate higher homogeneity within groups (Gelman & Hill, 2007).

```
ICC(outcome = “Satisfaction”, group = “Time”, data = data)
[1] 0.01019326
```

You can see that my ICC value is approximately .01, indicating that the relationship satisfaction of participants nested within a point of Time is quite different from each other.

Before moving to the next HLM analysis step, I want to make sure that my fixed effects regression coefficient is accurate. To do so, I will request a 95% confidence interval (CI) using confint.

If you are not familiar with a CI, the term refers to a range of values that may include the true population parameter with a certain range of percent confidence (mostly 95%). The formula is

Image by author

x bar is the sample mean

z is confidence level value

n is sample size

s is sample SD

A CI, let's say at 95%, contains two endpoints. We may set a lower 1% limit, meaning that the probability that the true population parameter is below the 1% limit of our data scores is only 1%. We may also set an upper 96% limit, meaning that the probability that the true population parameter is beyond the 96% limit of our data scores is only 4%. The upper and lower limits together indicate that an interval or the probability that we will find the true population parameter out of the range that we set (1% — 96%) is 5% (1% + 4%). So we have a 95% confidence interval that the true

parameter will be in the upper and lower limit range of our sample. If you want to learn more about CI and its relation to t-distribution, check out this link.

Now, *confidence levels* are different from *a confidence interval.* If we re-run a study several times and estimate a parameter of interest with a 95% CI, we will get different 95% CI values each Time due to errors in our data that could be caused by several factors, such as participants' factors, measurement errors, our moods during each analysis. However, 95% of those different CI values will cover the true parameter value, and this concept is *confidence levels*. If we set a lower limit of our confidence levels at 1%, it means that out of many experiments that we conduct repeatedly, the true parameter value will be lower than this 1% limit in only 1% of those many experiments. If we set an upper 96% limit, the probability that we will find the true parameter value higher than the upper limit is 4% of several experiments that we repeatedly conduct.

As humans like symmetrical things, people often set a 95% CI as a lower 2.5% limit and an upper 97.5% limit. The true population parameter value will be below the interval in 2.5% of repeated studies and above it in another 2.5% of those studies. Thus, the confidence levels will cover the true parameter in 95% of all conducted studies.

Let's get back to our example. If I want to know the confidence levels of model 2.1, I will use the following code.

```
confint(model2.1)
```

Results:

```
2.5 %    97.5 %
.sig01      0.0267156 0.2084871
.sigma      0.7951820 0.8389382
(Intercept) 4.9785129 5.2081911
```

The results indicate that if I re-rerun my study several times, 95% of the times, the intercept coefficient (i.e., the true mean of relationship satisfaction in population considering the random effects of Time) would be somewhere between 4.98–5.21 approximately.

Step 3: Fixed effects in the random intercept model

As I am mainly interested in the NPD's fixed effects, I will include the predictor in my random intercept model (model 2 or model 2.1). I still let the intercept vary, meaning that each point of Time may have different intercepts of relationship satisfaction scores. To generate fixed effects in the random intercept model, I will use lme() from the nlme package.

```
model3 = lme(Satisfaction~ NPD, data = data, method = "ML",
na.action = "na.omit", random = ~1|Time)
summary(model3)
```

Results:

```
Linear mixed-effects model fit by maximum likelihood
 Data: data
      AIC      BIC   logLik
  6468.46 6492.036 -3230.23

Random effects:
 Formula: ~1 | Time
        (Intercept)  Residual
StdDev:  0.07411888 0.8063175

Fixed effects: Satisfaction ~ NPD
               Value  Std.Error   DF  t-value p-value
(Intercept) 4.672165 0.06842444 2677 68.28210       0
NPD         0.122980 0.01491822 2677  8.24362       0
 Correlation:
    (Intr)
NPD -0.746

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-5.0666244 -0.4724214  0.1792983  0.7452213  1.6161859

Number of Observations: 2681
Number of Groups: 3
```

The fixed effects are significant. Let's compare whether the random intercept model with fixed effects (Model 3) is better than the random intercept model (Model 2).

```
anova(model3, model2)
```

Results:

```
Model df      AIC      BIC    logLik   Test L.Ratio p-value
model3     1  4 6468.460 6492.036 -3230.230
model2     2  3 6533.549 6551.231 -3263.775 1 vs 2 67.0889  <.0001
```

The results show a significant difference across the two models, indicating that adding fixed effects significantly improved the random intercept model.

An alternative for model fitting in Step 3 is to use the lmer function:

```
model3.1 <-lmer(Satisfaction~1+NPD+(1| Time), REML = FALSE, data =
data)
summary(model3.1)
```

Results:

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: Satisfaction ~ 1 + NPD + (1 | Time)
   Data: data

AIC      BIC   logLik deviance df.resid
  6468.5   6492.0  -3230.2   6460.5     2677

Scaled residuals:
    Min      1Q  Median      3Q     Max
-5.0666 -0.4724  0.1793  0.7452  1.6162

Random effects:
 Groups   Name        Variance Std.Dev.
 Time     (Intercept) 0.005494 0.07412
 Residual             0.650148 0.80632
Number of obs: 2681, groups:  Time, 3

Fixed effects:
            Estimate Std. Error t value
(Intercept)  4.67216    0.06840  68.308
NPD          0.12298    0.01491   8.247

Correlation of Fixed Effects:
    (Intr)
NPD -0.746
```

You see that the parameter estimates are quite close across the lme and lmer functions.

Step 4: Adding a random slope term.

In HLM, adding random slopes allow regression lines across groups of random effects to vary in terms of slope coefficients. In my case, the slopes between one's NPD and the outcome (relationship satisfaction) across different levels of Time could vary as people's NPD symptoms may be weakened or strengthened across Time points, depending on their life events. To test the assumption, I will nest NPD traits in Time and allow the slopes of NPD and relationship satisfaction to vary across different Time levels.

```
model4= lme(Satisfaction~ NPD, data = data, method = "ML", na.action
= "na.omit", random = ~NPD|Time, control = lmeControl(msMaxIter =
200))
summary(model4)
```

Results:

```
Linear mixed-effects model fit by maximum likelihood
 Data: data
      AIC      BIC   logLik
  6472.46 6507.823 -3230.23

Random effects:
 Formula: ~NPD | Time
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev      Corr
(Intercept) 0.072374062 (Intr)
NPD         0.002428596 0.131
Residual    0.806315723

Fixed effects: Satisfaction ~ NPD
               Value  Std.Error   DF  t-value p-value
(Intercept) 4.672058 0.06779927 2677 68.91016       0
NPD         0.123021 0.01498469 2677  8.20977       0
 Correlation:
    (Intr)
NPD -0.742

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-5.0663508 -0.4722466  0.1806865  0.7456579  1.6137660
```

```
Number of Observations: 2681
Number of Groups: 3
```

The output suggests that the variation in the intercept of Time is fitted with a larger SD of 0.0724. The variation in NPD slopes in predicting relationship satisfaction is fitted with a smaller SD of 0.0024. The results indicate that participants' relationship satisfaction likely differs across levels of Time more than the severity of NPD symptoms within each point of Time.

A weak positive correlation (Corr; r=0.131) between the intercept of Time and the NPD slope means that a more positive value of the intercept is slightly related to a more positive value of the slope. If participants' intercepts increase by one unit of SD, the slopes will only increase by 0.131 SDs. In other words, the intercept of relationship satisfaction obviously differs across Time, whereas a variation in the slope of the correlation between NPD and relationship satisfaction is subtler. Thus, it is highly likely that Model 4 (adding the random slope term) does not significantly improve Model 3(the random intercept model). Let's test the assumption.

```
anova(model3, model4)
```

Results:

```
Model df      AIC      BIC   logLik   Test     L.Ratio p-value
model3     1  4 6468.46 6492.036 -3230.23
model4     2  6 6472.46 6507.823 -3230.23 1 vs 2 0.000787942  0.9996
```

As expected, adding the random slope term does not significantly improve the random intercept model and increased the AIC value (i.e., worse fit). To exclude the random slope term or not depends on several factors, such as theories that inform your data, whether excluding or including the random slope makes the models converge, and whether you would like to get a parsimonious or maximal model. It all depends on your decision and field of study. This article provides additional detail about random effects that are worth reading.

Additional steps:

If you have an interaction term, you may test whether adding the term improves your model. I will test whether adding borderline personality disorder traits (BPD), which are highly comorbid with NPD, as a moderator will improve my random intercept model (model 3). I choose to ignore the random slope model (model4) as the term does not improve the model, and studies argue that NPD traits may not change across Time points.

```
model3withBPD<-lme(Satisfaction~NPD+BPD+BPD*NPD,
data = data, method = "ML", na.action = "na.omit", random = ~1|Time)
summary(model3withBPD)
```

Results:

```
Linear mixed-effects model fit by maximum likelihood
 Data: data
       AIC      BIC    logLik
  6425.735 6461.098 -3206.867

Random effects:
 Formula: ~1 | Time
        (Intercept)  Residual
StdDev:  0.07982052 0.7992555

Fixed effects: Satisfaction ~ NPD + BPD + BPD * NPD
                Value  Std.Error   DF  t-value p-value
(Intercept)  4.443310 0.09474416 2675 46.89799       0
NPD          0.153825 0.02988573 2675  5.14709       0
BPD          0.017154 0.00251750 2675  6.81408       0
NPD:BPD     -0.003436 0.00058873 2675 -5.83621       0
 Correlation:
        (Intr) NPD    BPD
NPD     -0.807
BPD     -0.417  0.251
NPD:BPD  0.600 -0.578 -0.907

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-5.2024359 -0.4590723  0.1866308  0.7317000  1.8891006

Number of Observations: 2681
Number of Groups: 3
```

The interaction term is significant. We will see whether adding the interaction improves Model 3:

```
anova(model3, model3withBPD)
```

As expected, adding the interaction term significantly improves my random intercept only model:

```
Model df      AIC      BIC    logLik   Test  L.Ratio p-value
model3            1  4 6468.460 6492.036 -3230.230
model3withBPD     2  6 6425.735 6461.098 -3206.867 1 vs 2 46.72568
<.0001
```

I hope by now, you have got a sense of how to conduct simple HLM. Please stay tuned for more complex HLM analysis in the future.

For the full codes used in this tutorial, please see below:

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. Take a look.

Get this newsletter

Data Science Cluster Analysis Modeling Editors Pick Getting Started