10

MULTILEVEL MODELING FOR CONTINUOUS RESPONSE VARIABLES

OBJECTIVES OF THIS CHAPTER

This chapter introduces multilevel modeling for continuous response variables. It starts with an introduction to multilevel modeling, model-building strategies, model fit statistics, centering, and data structure followed by a description of the research questions and data. Then several models, from the unconditional (null) model to the random-intercept model and random-coefficient model to the contextual models, are illustrated using R with step-by-step instructions. R commands are explained, and the output is interpreted for each model in detail. The chapter also illustrates how the results are displayed in publication-quality tables using the R command and reported in text. It focuses on model fitting with R, as well as on interpreting and presenting the results. After reading this chapter, you should be able to:

- Determine when multilevel modeling for continuous variables is used.
- Formulate multilevel models.
- Conduct multilevel modeling analysis for continuous response variables using R.
- Interpret the output.
- Compute and interpret the intraclass correlation coefficient (ICC).
- Be familiar with model fitting strategies.
- Compare models using the likelihood ratio test.
- Present results in publication-quality tables using R.
- Write the results for publication.

10.1 MULTILEVEL MODELING: AN INTRODUCTION

In previous chapters, we have focused on single-level analytic techniques for categorical response variables. Multilevel modeling has been widely used in education, social, and behavioral sciences in recent years, and researchers are increasingly interested in applying this technique to analyze multilevel data in their research. This chapter presents multilevel modeling for continuous outcome variables when the data structure has more than one level.

10.1.1 Multilevel Data Structure

Multilevel data, nested data, or hierarchical structured data have a data format in which observations at lower levels are nested within a higher level. For example, in businesses, employees are nested within companies; in educational research, students are nested in schools; in medical science, patients are nested within hospitals; in political sciences, voters are nested within districts; and in sociology, families are nested within communities. Observations in the same group could be more homogeneous than those across different groups, and thus, the assumption of independence is violated. Another type of multilevel data structure occurs in longitudinal studies in which there are repeated measures for each subject. In this case, measures for multiple time points are nested within a subject. This type of analysis is known as the multilevel analysis for change (Singer & Willett, 2003). The focus of this text is the cross-sectional data structure.

What can multilevel modeling do? There are several advantages to using multilevel modeling. First, in multilevel modeling, variables at higher levels can be included in the model to estimate their relationships with the outcome variable. Second, we can examine whether an effect or slope of a variable at a lower level is allowed to vary among higher level variables. Third, we can also examine whether higher level variables moderate the relationships between lower level variables and the outcome variable.

10.1.2 Intraclass Correlation

With a multilevel data structure, the observations within a group or cluster may violate the assumption of independency. In other words, the observations within the same group or cluster may be more homogeneous than those in other groups or clusters. To justify why multilevel modeling is warranted, we also need to examine how much variance of the outcome variable is accounted for by groups or clusters. The intraclass correlation coefficient (ICC) is used as an index to measure the proportion of variance in the outcome variable explained by groups or clusters (Hox, 2010; Raudenbush & Bryk, 2002; Snijders & Bosker, 2012). It is the ratio of the between-group variance to the total variance. Its range is from 0 to 1. When it is close to 0, it means that using multilevel modeling might not be a good strategy for data analysis. A larger ICC provides strong evidence that this technique is needed.

10.1.3 Overview of a Basic Two-Level Model

Let us look at a basic two-level model with one predictor in each level. In the following example, researchers are interested in estimating the math achievement scores from a student-level variable, math self-efficacy, and a school-level variable (whether a school is public). The level 1 predictor variable is math self-efficacy (`gceffic`). Both the intercept and the slope of `gceffic` are allowed to vary randomly across schools. The level 2 predictor is whether a school is public or private (`public`). Following the convention of model specification by Raudenbush and Bryk (2002), a two-level model can be expressed as:

$$\text{Level 1}: Y_{ij} = \beta_{0j} + \beta_{1j}\texttt{gceffic}_{ij} + r_{ij}$$
$$\text{Level 2}: \beta_{0j} = \gamma_{00} + \gamma_{01}\texttt{public}_j + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + \gamma_{11}\texttt{public}_j + u_{1j}$$

where Y_{ij} represents the math achievement score for the ith student in the jth school, β_{0j} is the level 1 intercept, the average math achievement score in the jth school, β_{1j} is the level 1 slope for `gceffic` in the jth school, and $\texttt{gceffic}_{ij}$ represents the value of math self-efficacy of the ith student in the jth school. r_{ij} is the random error, which is the deviation of the individual's math score from the average math score in the school.

The γ_{00} is the overall intercept of the outcome variable across schools. It is the predicted mean math achievement score controlling for the effect of the level 2 predictor (i.e., when the level 2 predictor variables are held constant at 0). γ_{01} represents the effect of the level 2 variable `public` on the intercept. γ_{10} represents the mean of the level 1 slope when the level 2 predictors are held constant at 0, and γ_{11} represents the effect of the level 2 predictor `public`. γ_{11} is the cross-level interaction between `gceffic` and `public`, which moderates the effect of math self-efficacy on math achievement scores. u_{0j} and u_{1j} are the random effects associated with the level 1 intercept and the slope of `gceffic` across schools, respectively. In other words, the level 1 intercept and the slope of `gceffic` are allowed to vary randomly across schools so their respective variances (i.e., between-group variance and slope variance) need to be estimated.

Fix Effects Versus Random Effects

In multilevel modeling, fixed effects are the regression coefficients that estimate the relationships between the predictor variables and the outcome variable from the entire population (West et al., 2014), whereas random effects are the randomly varying parameters across higher level units. For example, the random intercept (u_{0j}) in the previous example is a random deviation from the overall intercept, and the random coefficient (u_{1j}) is a random deviation from the overall fixed effect, the slope of `gceffic`. Variance and covariance components are estimated for random effects. For example, in the two-level model above, the variance and covariance components include the between-group variance for the intercept, the variance for the random coefficient or slope, and the covariance between the intercept and the coefficient. Sometimes, random effects are estimated in terms of standard deviations since they are just the square root of variances.

What if an ordinary least-squares (OLS) regression instead of multilevel modeling is used? In other words, what will happen if random effects are not estimated? When a single-level regression analysis is conducted to analyze multilevel data, the precision of parameter estimates is compromised (Heck & Thomas, 2015). Heck and Thomas (2015) pointed out that multilevel modeling has four advantages over the OLS regression method. These advantages include incorporating regression equations at different levels into a single statistical model, more accurate estimates of standard errors, flexibility in specifying various models, and the capability of estimating different types of response variables.

By using multilevel modeling, we can estimate the influence of both the student-level and school-level predictors on the outcome variable. We can also investigate whether there are cross-level interactions between variables at different levels. In addition, we can estimate random effects by allowing the intercept and slopes of lower level predictors to vary randomly at higher levels. The variance and covariance components of the random effects can also be determined. For example, the estimated error variance for r_{ij} is the within-group variance, the estimated variance for u_{0j} is the intercept variance, which is the between-group variance, and the estimated variance for u_{1j} is the slope variance.

10.1.4 Model-Building Strategies

Although researchers may have their own strategies to build multilevel models, a common practice illustrated by Raudenbush and Bryk (2002), Snijders and Bosker (2012), and other publications (Garson, 2013, 2020; Heck et al., 2010; Heck & Thomas, 2015; Kreft & de Leeuw, 1998; Luke, 2004; West et al., 2014) is to start from a basic model and work up to more complex models. Specifically, this strategy starts with the unconditional means model with no level 1 or level 2 predictors (null model). This model is equivalent to the one-way random-effects analysis of variance (ANOVA) model. This model serves as the baseline model for future model comparisons. The unconditional means model estimates the overall average of the outcome variable across all subjects and the between-group and within-group variances. The variance between groups or clusters estimated from this model can be used to calculate the ICC so that we can decide whether multilevel modeling is needed. The between-group and within-group variances can also be used to compute the proportion of variance explained after the level 1 and level 2 predictors are added to the model. Next, we can add level 1 predictors and build a random-intercept model and a random-coefficient model. In the random-intercept model, only intercepts are allowed to vary freely in higher level clusters and the level 1 slopes are fixed. In the random-coefficient model, both intercepts and coefficients of the level 1 predictors are allowed to vary across higher level clusters. Finally, we add level 2 predictors to the level 2 model so the random-coefficient model includes both level 1 and level 2 predictors. This model is referred to as the contextual model. If the model has more than two levels, then higher level predictors can be added.

Although the earlier simple-to-complex model building strategy is commonly followed by researchers, you can decide whether all the steps need to be followed for your own research.

10.1.5 Model Fit Statistics

As with logistic regression models, several measures of goodness-of-fit statistics, such as the likelihood ratio test, the Akaike information criterion (AIC), and Bayesian information criterion (BIC) statistics, can be applied to multilevel modeling for continuous outcome variables. The following discussion is a brief review of these tests (see Chapter 3 for a more detailed description).

Likelihood Ratio Test

The likelihood ratio test can be used to compare nested models. Models are nested when one model, the reduced model, is a special case of the other one, the full model. For example, more constraints can be put on parameters in one model than the other. A simple case is that one model (model 1) contains predictor variables X_1 and X_2, and the second model (model 2) contains an extra variable X_3. We conclude that model 1 is nested within model 2 since predictors in the former are the subset of the latter. In multilevel modeling, an unconditional model is nested within a random-intercept model, which is then nested within a random-coefficient model and finally a contextual model with both level 1 and level 2 predictor variables.

The likelihood ratio test statistic is expressed as the difference in -2LL between nested models, where LL stands for the log likelihood value for the fitted model with either the full maximum likelihood (ML) estimation or the restricted maximum likelihood (REML) estimation. Since deviance equals -2LL, the likelihood ratio test is also referred to as the difference in deviance, which follows a chi-square distribution, with the degrees of freedom of the distribution equaling the difference in the number of parameters between two nested models. The difference in deviance is often expressed as a generic form: $G =$ Deviance for the reduced model $-$ Deviance for the full model or $D_{\text{Reduced}} - D_{\text{Full}}$, where the reduced model has fewer variables and is nested within the full model. As with logistic regression models in previous chapters, we use the likelihood ratio test to compare nested models from a simple model with one predictor to more complex models with multiple predictors. In multilevel modeling, we can also use the same test to compare a series of nested models from the unconditional means model to the random-intercept model to more complex models, such as the contextual models with level 1 and level 2 predictor variables. A significant likelihood ratio chi-square test statistic indicates that a more complex model fits the data better than a simpler, nested model.

Information Criteria Indices: AIC and BIC

The AIC and the BIC statistics can be used to compare non-nested models. Both AIC and BIC statistics can be applied to multilevel modeling. The AIC statistic adjusts the deviance by the number of parameters. It is expressed as $-2\text{LL} + 2k$ or deviance $+ 2k$, where k is the number of parameters. The BIC statistic is defined as $\text{BIC} = -2\text{LL} + \ln(n) \times k = D_m + \ln(n) \times k$, where n is the sample size and k is the number of parameters. Smaller AIC and BIC statistics indicate a better fit of the model.

10.1.6 Centering

The purpose of centering is to make the results more interpretative. It is often used when a predictor variable does not have a meaningful value of zero. By subtracting the mean of a predictor variable from each value, we obtain a meaningful zero for the predictor variable. Predictors at both levels of the model can be centered. Two types of centering are often used in multilevel modeling. One is grand-mean centering, and the other is group-mean centering. For the grand-mean centering, we subtract the grand mean of the predictor variable from each value of the sample. For example, when we use grand-mean centering of the math efficacy (`efficacy`), we compute the overall mean of this variable and then subtract it from each score of efficacy. For the group-mean centering, we subtract the group mean, which is the mean of each group or cluster where individuals are nested from each score. For example, to group-mean center the predictor variable `efficacy`, we first compute the group mean for each school where a student belongs and then subtract the mean for each school (i.e., group mean) from each score of efficacy.

The choice of grand-mean centering and group-mean centering is complicated, and this topic has been widely discussed in the literature (Enders & Tofighi, 2007; Garson, 2020; Hofmann & Gavin, 1988; Hox, 2010; Kreft et al., 1995; Luke, 2004; Ma et al., 2008; McCoach, 2010; Paccagnella, 2006). The advantage of grand-mean centering is that the subsequent multilevel models with this centering are mathematically equivalent to the models using raw scores without centering. It also makes the computation faster and reduces convergence problems (Hox, 2010). On the other hand, group-mean centering produces a model that is mathematically different from the raw score model. Hox (2010) suggested using group-mean centering with caution for novice users. Enders and Tofighi (2007) suggested that researchers use group-mean centering when level 1 variables and the interactions among them are the research interests, whereas grand-mean centering is a good choice if level 2 variables are the focus after controlling for level 1 variables. Therefore, the decision of using centering methods should be based on research questions or theories.

10.1.7 Sample Size

In multilevel modeling, the sample size needs to be considered at different levels. Theoretically, we would like to have a large sample size for all levels to obtain unbiased estimates for fixed and random effects. Factors such as the complexity of the model, the intraclass correlation, cross-level interactions, and power considerations impact sample size determination. Although there is no definite number to define a sufficient sample size in the literature, researchers have suggested several rules-of-thumb for a two-level model. Kreft and de Leeuw (1998) recommended a sample size of more than 20 groups for cross-level interactions. Maas and Hox (2005) conducted simulations and the results suggested that a sample size of 50 or more groups is needed to obtain unbiased estimates of the standard errors at level 2. They also found that the standard errors were underestimated at level 2 with a sample size of 30 groups. Hox (2010) suggested a 50/20 rule for cross-level interactions, that is, 50 level 2 groups with 20 level 1 subjects. In addition, a 100/10 rule (i.e., 100 level 2 groups with 10 level 1 subjects) was suggested when the focus was on random effects. For a more detailed review of other simulation studies, refer to Garson (2020). Sometimes, when the group number at level 2 is small, multilevel modeling can still be a useful tool for analyzing nested data, but the results should be interpreted with caution.

10.1.8 Data Structure for Model Fitting

In R, the data structure for multilevel modeling is a single dataset containing variables at different levels. For example, a two-level model needs a single dataset with both student-level and school-level variables and the former variables are nested with the latter. If the original student-level and school-level variables are saved in two separate datasets, then they need to be merged into one dataset in a format where students are nested within schools. This stacked data format requires that each school have multiple records, one for each student. For example, when 50 students are selected from a school, in the dataset, 50 students with different IDs (with each one having a row) are nested within the same school ID. Such a dataset needs to be created before model fitting.

10.2 MULTILEVEL MODELING FOR CONTINUOUS OUTCOME VARIABLES

10.2.1 Research Example and Research Questions

In the following example, researchers are interested in examining the relationships between high-school students' mathematics achievement and mathematics self-efficacy, school type, and school climate using the Educational Longitudinal Study of 2002 (ELS: 2002) data. The student-level predictor variable is students' mathematics self-efficacy, and the two school-level predictor variables are school type and school climate. The following research questions will be addressed:

1. Can high-school students' mathematics scores be predicted by students' mathematics self-efficacy?
2. Do school characteristics, such as school type and school climate, impact math achievement?
3. Do mathematics scores vary across schools?
4. Does the relationship between mathematics self-efficacy and mathematics achievement vary across schools?
5. Are there any interaction effects between the two school-level variables (i.e., school type and school climate) and math self-efficacy? In other words, does school type or school climate moderate or influence the relationship between mathematics self-efficacy and mathematics achievement? Put it another way: Does the effect of mathematics self-efficacy on mathematics achievement vary across school type and school climate?

10.2.2 Description of the Data and Sample

The ELS:2002 base-year data are used for the following analyses. The variables are listed as follows:

- mathach: mathematics item response theory (IRT) estimated scores of high-school students
- gceffic: math self-efficacy (grand-mean centered)
- public: school type (1= public, 0 = private and others)
- csclimat: school climate (grand-mean centered).

10.3 MULTILEVEL MODELING FOR CONTINUOUS RESPONSE VARIABLES WITH R

10.3.1 The lme() Function in the nlme Package

The lme() function in the nlme package (Pinheiro et al., 2021) is used for multilevel models with continuous response variables. Since nlme is a user-written package, you need to install it first by typing install.packages("nlme") and then load the package by typing library(nlme).

The basic syntax for the model formula of lme() includes two components. The first component or argument specifies the dependent variable and the predictor variable(s) for the fixed effects component, which are separated by the tilde (~). When there are multiple predictor variables in the formula, they are connected by plus (+) symbols. This is the fixed effects part of the model, which looks the same as the model formula for any linear regression in lm(). The predictor variables from different levels are specified here, but the command itself does not tell the specific levels within which the variables belong to. Next, the random argument specifies the random effects of the model and the grouping variable which are separated by a vertical line (|). A predictor variable or a list of predictor variables that have random coefficients is specified first, followed by a vertical line (|), and then, an identifier variable at a higher level as the grouping variable. Sometimes the grouping variable may be omitted. In addition to the model formula, several optional arguments, such as the data argument for the data frame, method = "ML" for the maximum likelihood estimation, and na.action = "na.omit" for removing missing data, can be specified. For example, the command lme(y ~ x, random = ~ 1| schid, method = "ML") tells R to fit a multilevel model to estimate a continuous outcome variable y with a predictor variable x, and random = ~ 1| schid specifies the random intercepts varying across schools with 1 as the intercept and schid as the identifier variable. The method = "ML" argument requests the full maximum likelihood estimation rather than the default restricted maximum likelihood (method = "REML"). For more details on how to use this function, type help(lme) in the command prompt after loading the nlme package.

10.3.2 Unconditional Means Model (Model 1: Null Model)

The unconditional means model or the null model is known as the one-way random-effects ANOVA. Neither level 1 nor level 2 predictor variables are included in the model. This model can be expressed as follows:

$$\text{Level 1}: \; Y_{ij} = \beta_{0j} + r_{ij}$$
$$\text{Level 2}: \; \beta_{0j} = \gamma_{00} + u_{0j}$$

The m1 <- lme(mathach ~ 1, random = ~1|SCH_ID, na.action = "na.omit", method = "ML", data = chp10) command is used to fit the unconditional model (model 1). In the lme() function, the fixed effects part of the model is specified first. Because this is the unconditional model without any predictor variables, the continuous outcome variable mathach and the intercept 1 are specified as mathach ~1. The random effects part of the model is then specified after the fixed part as random = ~1 which is separated from the grouping variable SCH_ID by a vertical line (|). SCH_ID is the grouping variable or identifier variable. Since no random coefficients for any predictor variables are specified in this model, only 1 is specified as the intercept. The method = "ML" argument requests the maximum likelihood estimation method. The na.action = "na.omit" argument removes the missing data if there are any. The fitted model is named m1 and the output is displayed by the summary(m1) command.

```
> library(foreign)
> chp10<-read.dta("C:/CDA/els2002.dta")
> chp10  <-  chp10[!is.na(chp10$mathach)&!is.na(chp10$efficacy)&!is.na(chp10$
public)&!is.na(chp10$sclimate), ]
> chp10$csclimat <- chp10$sclimate-mean(chp10$sclimate, na.rm=TRUE)
> chp10$gceffic <- chp10$efficacy-mean(chp10$efficacy, na.rm=TRUE)
> attach(chp10)
> #install.packages("nlme")
> library(nlme)
> # Null model with lme()
> m1 <- lme(mathach ~1, random =~1|SCH_ID, na.action="na.omit", method="ML",
data=chp10)
> summary(m1)
Linear mixed-effects model fit by maximum likelihood
Data: chp10
      AIC       BIC      logLik
 75354.68  75376.27   -37674.34

Random effects:
 Formula: ~1| SCH_ID
        (Intercept)    Residual
StdDev:      5.6338    10.45764
```

```
Fixed effects: mathach ~ 1
                 Value  Std.Error   DF  t-value  p-value
(Intercept)   39.12354  0.2537879 9249 154.1584        0

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.08523311 -0.71736082  0.02771707  0.74094481  2.99690652

Number of Observations: 9866
Number of Groups: 617
```

Interpreting the Output

The R output for the `lme()` function includes the model fit statistics such as the AIC, the BIC, and the log-likelihood value, the random effects, the fixed effects, the standardized within-group residuals, and the number of observations and groups.

First, the statement "`Linear mixed-effects model fit by maximum likelihood`" tells us the linear mixed model is fitted with the ML estimation rather than the REML estimation method. The AIC and BIC statistics are 75,354.68 and 75,376.27, respectively. The log likelihood value is −37,674.34. These fit statistics will be used for model comparisons in the following sections.

Then, the random effects section contains the model formula of the random effects and the standard deviations of the intercept and residual. The column (labeled `Intercept`) reports the standard deviation at level 2 (i.e., schools) and the column (labeled `Residual`) reports the within-school standard deviation. The between-school standard deviation is 5.634 and the within-school standard deviation is 10.458. To request the variance components of the model, we need to either square the standard deviations or use the `VarCorr(m1)` command. The between-school variance (τ_{00}) is 31.740, and the within-school variance (σ^2) is 109.362.

```
> VarCorr(m1)
SCH_ID = pdLogChol(1)
              Variance   StdDev
(Intercept)    31.7397  5.63380
Residual      109.3623 10.45764
```

Next, the fixed effects section contains the model formula of the fixed effects and the estimate for the intercept, its standard error, the degrees of freedom, the t statistic, and the associated p value. Since no predictors are included in the model, this section only displays the estimate for the intercept. The intercept γ_{00} (labeled `Intercept`) is 39.124, which is significant (p = `.000`). This means that the average math achievement score across all schools is 39.124.

Since currently the `confint()` function does not work with `lme()` function, we use the `intervals(m1)` command to extract the 95% confidence intervals of the estimates. The results are displayed as follows.

```
> confint(m1)
Error in confint.lme(m1) :
  not (yet) implemented. Contributions are welcome; use intervals() instead (for
now)
> intervals(m1)
Approximate 95% confidence intervals
 Fixed effects:
                  lower       est.      upper
(Intercept)    38.62609   39.12354    39.621

 attr(,"label")
 [1] "Fixed effects:"
 Random Effects:
  Level: SCH_ID
                    lower      est.      upper
sd((Intercept))  5.246673    5.6338   6.049491

 Within-group standard error:
    lower        est.      upper
 10.30785    10.45764   10.60962
```

The ICC is defined as the proportion of total variance in the outcome variable ($\sigma^2 + \tau_{00}$) explained by the between-group variance (τ_{00}). It is expressed as ICC $= \tau_{00}/(\sigma^2 + \tau_{00})$.

From the earlier output, we compute

$$\text{ICC} = 31.740/(31.740 + 109.362) = .225,$$

which indicates that 22.5% of the total variance is accounted for by schools in level 2.

Finally, the output shows the minimum, first quartile, median, third quartile, and maximum values of the standardized within-group residuals. These residuals are normally distributed. In addition, the number of observations and the number of groups are provided. A total of 9,866 observations in level 1 is nested in 617 groups (i.e., schools) in level 2.

10.3.3 Random-Intercept Model (Model 2)

Next, we include the predictor variable `gceffic` (math self-efficacy) to the level 1 equation, with all other parts of the level 1 equation the same. The model is referred to as the random-intercept model since the intercept is allowed to vary across schools. This model can be expressed as follows:

$$\begin{aligned}
\text{Level 1}: Y_{ij} &= \beta_{0j} + \beta_{1j}\texttt{gceffic}_{ij} + r_{ij} \\
\text{Level 2}: \beta_{0j} &= \gamma_{00} + u_{0j} \\
\beta_{1j} &= \gamma_{10}
\end{aligned}$$

In the level 1 equation, the predictor variable is math self-efficacy (gceffic) and the outcome variable is math achievement (mathach). The predictor variable is grand-mean centered. The level 2 equations express the random intercepts (β_{0j}) and the fixed slopes (β_{1j}).

The command m2 <- lme(mathach ~ gceffic, random = ~1|SCH_ID, na.action = "na.omit", method = "ML", data = chp10) is used to fit the random-intercept model (model 2) after a predictor variable gceffic is added to the fixed part of the model. The fitted model is named m2. Just as in the unconditional model, no random coefficients are specified in the random part of the model. The following output is displayed by the summary(m2) command.

```
> # Random-intercept model
> m2 <- lme(mathach ~ gceffic, random = ~1|SCH_ID, na.action="na.omit", method="ML",
data=chp10)
> summary(m2)
Linear mixed-effects model fit by maximum likelihood
 Data: chp10
      AIC       BIC     logLik
73960.62  73989.41  -36976.31

Random effects:
 Formula: ~1 | SCH_ID
         (Intercept)    Residual
StdDev:     5.243221    9.743885

Fixed effects: mathach ~ gceffic
                Value   Std.Error   DF    t-value    p-value
(Intercept)  39.12132   0.2362576  9248   165.5875         0
gceffic       4.67671   0.1207781  9248    38.7215         0
Correlation:
        (Intr)
gceffic 0

Standardized Within-Group Residuals:
        Min           Q1          Med           Q3          Max
-3.96543934  -0.66900059   0.05549942   0.71325893   3.30894729

Number of Observations: 9866
Number of Groups: 617
```

Interpreting the Output

In the fixed-effects section, the estimated intercept is 39.121, and the coefficient for gceffic is 4.677. Both estimates are significant ($p < .001$). The intercept can be interpreted as follows: The average math achievement score is 39.121 for students with a value of math self-efficacy at 0. The coefficient for gceffic is 4.677, $t = 38.722$,

$p < .001$, which indicates that for a one-unit increase in math self-efficacy, there is an increase of 4.677 points in math achievement scores.

We also use the `intervals(m2)` command to extract the 95% confidence intervals of the parameter estimates. The results are displayed as follows.

```
> intervals(m2)
Approximate 95% confidence intervals

 Fixed effects:
                 lower       est.      upper
(Intercept)   38.65825  39.121317  39.584387
gceffic        4.43998   4.676707   4.913435
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: SCH_ID
                    lower      est.     upper
sd((Intercept))  4.882778  5.243221  5.630272

 Within-group standard error:
   lower      est.     upper
9.604314  9.743885  9.885484
```

To request the variance components of the model, we need to either square the standard deviations or use the `VarCorr(m2)` command. The between-school variance (τ_{00}) is 27.491, and the within-school variance (σ^2) is 94.943.

```
> VarCorr(m2)
SCH_ID = pdLogChol(1)
            Variance  StdDev
(Intercept) 27.49137  5.243221
Residual    94.94330  9.743885
```

After the level 1 predictor is entered in the model, the variance for the random intercept has decreased to 27.491, compared with the original 31.740 in the unconditional model.

Likelihood Ratio Test Comparing the Unconditional Model and the Random Intercept Model

```
> # Model comparison
> anova(m1, m2)
   Model df      AIC      BIC    logLik   Test  L.Ratio p-value
m1     1  3 75354.68 75376.27 -37674.34
m2     2  4 73960.62 73989.41 -36976.31 1 vs 2 1396.056  <.0001
```

The likelihood ratio test is used to compare the unconditional model (model 1) and the random-intercept model (model 2). The `anova(m1, m2)` command compares the log-likelihood statistics of these two models using the likelihood ratio test. The likelihood ratio test $\chi^2_{(1)} = 1{,}396.056$, $p < .001$, which indicates that the random-intercept model fits the data better than the unconditional model.

10.3.4 Random-Coefficient Model: Random-Intercept and Slope Model With Level 1 Variable (Model 3)

In addition to the random intercept, level 1 slopes (i.e., the coefficients of the level 1 predictors) can also be specified to be random. In other words, a predictor may have a random slope across clusters. For example, we may allow the effect of math self-efficacy on math achievement to vary across different schools. This model can be expressed as follows:

$$\text{Level 1}: Y_{ij} = \beta_{0j} + \beta_{1j}\texttt{gceffic}_{ij} + r_{ij}$$
$$\text{Level 2}: \beta_{0j} = \gamma_{00} + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + u_{1j}$$

The level 1 equation for the random-coefficient model is the same as that for the random-intercept model. The math self-efficacy (`gceffic`) is still the only predictor variable, and math achievement is the outcome variable. Unlike the random-intercept model, in the random-coefficient model, the level 2 equations specify that both the intercept and the coefficient at level 1 are random across schools.

The command `m3 <- lme(mathach ~ gceffic, random = ~gceffic| SCH_ID, na.action = "na.omit", method = "ML", data = chp10)` is used to fit the random-coefficient model (model 3) after a predictor variable `gceffic` is added to the random part of the model. The fitted model is named `m3`. The `summary(m3)` command displays the output of the fitted model.

```
> # Random-coefficient model
> m3 <- lme(mathach ~ gceffic, random = ~gceffic|SCH_ID, na.action="na.omit",
method="ML", data=chp10)
> summary(m3)
Linear mixed-effects model fit by maximum likelihood
 Data: chp10

      AIC        BIC      logLik
 73953.39   73996.57    -36970.7

Random effects:
 Formula: ~gceffic | SCH_ID
 Structure: General positive-definite, Log-Cholesky parametrization
             StdDev     Corr
(Intercept)  5.237083   (Intr)
gceffic      1.003340    0.389
Residual     9.708897
```

```
Fixed effects: mathach ~ gceffic
                 Value  Std.Error   DF   t-value  p-value
(Intercept)   39.09802  0.2360843 9248 165.61046        0
gceffic        4.64856  0.1281118 9248  36.28516        0
 Correlation:
        (Intr)
gceffic  0.115

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.88739725 -0.66933019  0.05610879  0.71053443  3.31730425

Number of Observations: 9866
Number of Groups: 617
```

The fixed effects in the output look similar to those in the random-intercept model (model 2). We use the `intervals(m3, which = "fixed")` command to extract the 95% confidence intervals of the parameter estimates of the fixed effects. Without the `which = "fixed"` argument, an error message will occur since the confidence intervals cannot be computed for the variance and covariance components. The results are displayed as follows.

```
> intervals(m3)
Error in intervals.lme(m3) :
  cannot get confidence intervals on var-cov components: Non-positive definite
approximate variance-covariance
Consider 'which = "fixed"'
> intervals(m3, which="fixed")
Approximate 95% confidence intervals

 Fixed effects:
                lower      est.     upper
(Intercept) 38.635295 39.098025 39.560755
gceffic      4.397457  4.648559  4.899661
attr(,"label")
[1] "Fixed effects:"
```

To request the variance and covariance components of the model, we use the `VarCorr(m3)` command. The results are displayed as follows.

```
> VarCorr(m3)
SCH_ID = pdLogChol(gceffic)
            Variance  StdDev   Corr
(Intercept) 27.427040 5.237083 (Intr)
gceffic      1.006691 1.003340 0.389
Residual    94.262689 9.708897

> .389*1.003*5.237
[1] 2.043305
```

Interpreting the Output

The VarCorr(m3) command reports the unstructured variance–covariance components. The between-school variance (τ_{00}) is 27.427 and the within-school variance (σ^2) is 94.263.

The variance for the random coefficient of gceffic or the slope variance (labeled var(gceffic)) is 1.007. The output reports the correlation between gceffic and the intercept rather than the covariance between them. The correlation coefficient for gceffic and the intercept is .389. We can compute the covariance between gceffic and the intercept with the correlation between them and their standard deviations. The covariance equals the product of the correlation and the two standard deviations, so we compute .389*1.003*5.237 = 2.043.

The within-school variance or the level 1 residual variance (labeled var(Residual)) is 94.263, which is similar to that of the random-intercept model.

Model Comparisons Using the Likelihood Ratio Test

```
> anova(m2, m3)
    Model  df      AIC       BIC     logLik   Test   L.Ratio  p-value
m2      1   4  73960.62  73989.41  -36976.31
m3      2   6  73953.39  73996.57  -36970.70  1 vs 2  11.23146   0.0036
```

The likelihood ratio test is used to compare the random-intercept model (model 2) and the random-coefficient model with a level 1 predictor (model 3). The anova(m2, m3) command compares the log likelihood statistics of these two models. The likelihood ratio test $\chi^2_{(2)} = 11.231$, $p < .01$, which indicates that the random-coefficient model has a significantly better fit than the random-intercept model. Therefore, allowing a random coefficient in the model is justified.

10.3.5 Contextual Model With Level 1 and Level 2 Variables (Model 4)

The contextual model is a special case of the random-coefficient model when both level 1 and level 2 predictor variables are included. The two level 2 equations specify the intercept and slope from the level 1 equation to be random. In this example, two school-level variables, public and csclimat, are added to the level 2 equation. This model can be expressed as follows:

$$\text{Level 1}: Y_{ij} = \beta_{0j} + \beta_{1j}\texttt{gceffic}_{ij} + r_{ij}$$
$$\text{Level 2}: \beta_{0j} = \gamma_{00} + \gamma_{01}\texttt{public}_j + \gamma_{02}\texttt{csclimat}_j + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + u_{1j}$$

With the level 1 equation the same as that for the previous random-coefficient model (model 3), the two school-level predictor variables are added to the equation for the random intercept at level 2. On the other hand, the equation for the random slope β_{1j} contains no predictor variables.

The command `m4 <- lme(mathach ~ gceffic + public + csclimat, random = ~gceffic|SCH_ID, na.action = "na.omit", method = "ML", data = chp10)` is used to fit the contextual model (model 4) after the two level 2 predictor variables `public` and `csclimat` are added to the model.

```
> # Contextual model with predictor variables in both levels
> m4 <- lme(mathach ~ gceffic + public + csclimat, random = ~gceffic|SCH_ID,
na.action = "na.omit", method = "ML", data = chp10)
> summary(m4)
Linear mixed-effects model fit by maximum likelihood
 Data: chp10
      AIC       BIC    logLik
 73806.56  73864.14 -36895.28

Random effects:
 Formula: ~gceffic | SCH_ID
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 4.447067  (Intr)
gceffic     1.037690  0.565
Residual    9.713777

Fixed effects: mathach ~ gceffic + public + csclimat
                Value Std.Error   DF  t-value p-value
(Intercept)  41.39652 0.4556953 9248 90.84255       0
gceffic       4.62698 0.1282449 9248 36.07922       0
publicpublic -2.89695 0.5248300  614 -5.51979       0
csclimat      3.02747 0.3249720  614  9.31610       0
 Correlation:
             (Intr) gceffc pblcpb
gceffic       0.065
publicpublic -0.890  0.014
csclimat     -0.304 -0.008  0.350

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.88077064 -0.67411813  0.05771525  0.70333885  3.41379144

Number of Observations: 9866
Number of Groups: 617
```

Interpreting the Output

The fixed-effects table displays the intercept and coefficients for both the level 1 and level 2 predictor variables. The coefficient for gceffic is 4.627, $t = 36.079$, $p < .001$, which indicates that students with higher mathematics self-efficacy tend to have higher mathematics achievement when holding other predictors constant. The effects of two school-level predictor variables are significant. The coefficient for public is −2.897, $t = -5.520$, $p < .001$, which indicates that students' mathematics scores in public schools tend to be lower than those in private schools. The coefficient for csclimat is 3.027, $t = 9.316$, $p < .001$, which indicates that schools with a better social climate tend to have higher mathematics scores.

We use the intervals(m4, which= "fixed") command to extract the 95% confidence intervals of the parameter estimates of the fixed effects. The results are displayed as follows.

```
> intervals(m4, which = "fixed")
Approximate 95% confidence intervals

 Fixed effects:
                   lower        est.       upper
(Intercept)    40.503438   41.396520   42.289602
gceffic         4.375639    4.626976    4.878313
publicpublic   -3.927424   -2.896953   -1.866483
csclimat        2.389408    3.027470    3.665532

attr(,"label")
[1] "Fixed effects:"
```

To request the variance and covariance components of the model, we use the VarCorr(m4) command. The results are displayed as follows.

```
> VarCorr(m4)
SCH_ID = pdLogChol(gceffic)
             Variance     StdDev     Corr
(Intercept)  19.776405   4.447067   (Intr)
gceffic       1.076801   1.037690   0.565
Residual     94.357456   9.713777

> .565*1.038*4.447
[1] 2.608032
```

The variance for the random coefficient of gceffic or the slope variance (labeled gceffic)) is 1.077, the between-group variance (labeled Intercept) is 19.776. Since the correlation coefficient for gceffic and the intercept is .565 and the standard deviations of gceffic and the intercept are 1.038 and 4.447, respectively, we can compute the covariance between gceffic and the intercept as follows: .565*1.038*4.447 = 2.608. Both the slope variance and the covariance in model 4 look similar to those in the random-coefficient model (model 3). After two school-level predictors are included in the model, the between-group variance decreases from 27.427 to 19.776. The percentage decrease is computed as (27.427 − 19.776)/27.427 = 27.9%. In other words, the school-level variables explain 27.9% of the between-group variance.

```
> (27.427-19.776)/27.427
[1] 0.2789587
```

The within-school variance or the level 1 residual variance (labeled Residual) is 94.357, which is similar to that of the random-coefficient model.

Model Comparisons Using the Likelihood Ratio Test

```
> anova(m3, m4)
   Model df      AIC      BIC    logLik   Test  L.Ratio p-value
m3     1  6 73953.39 73996.57 -36970.70
m4     2  8 73806.56 73864.14 -36895.28 1 vs 2 150.8293  <.0001
```

The anova(m3, m4) command compares the log-likelihood statistics of the contextual model with both level 1 and level 2 predictor variables (model 4: m4) and the random-intercept and slope model with a level 1 predictor (model 3: m3). A comparison between these two models yields the value of the likelihood ratio test $\chi^2_{(2)} = 150.829$, $p < .001$, which indicates that the contextual model fits the data better.

10.3.6 Contextual Model With Cross-Level Interactions (Model 5)

We can also add cross-level interactions to the model by including level 2 predictor variables in the equation for the random slope (β_{1j}). This model can be expressed as follows:

$$\begin{aligned}
\text{Level 1}: \; Y_{ij} &= \beta_{0j} + \beta_{1j}\texttt{gceffic}_{ij} + r_{ij} \\
\text{Level 2}: \; \beta_{0j} &= \gamma_{00} + \gamma_{01}\texttt{public}_j + \gamma_{02}\texttt{csclimat}_j + u_{0j} \\
\beta_{1j} &= \gamma_{10} + \gamma_{11}\texttt{public}_j + \gamma_{12}\texttt{csclimat}_j + u_{1j}
\end{aligned}$$

In R, we need to add the interaction term between public and gceffic and the interaction term between csclimat and gceffic in the model formula. The command m5 <- lme(mathach ~ gceffic + public + csclimat + public*gceffic + csclimat*gceffic, random = ~gceffic|SCH_ID, method = "ML", data = chp10) is used to fit the contextual model with cross-level interactions (model 5) after the cross-level interactions, `public*gceffic` and `csclimat*gceffic`, are added to the model. The fitted model is named `m5`. The `summary(m5)` command displays the resulting output.

```
> # Contextual model with cross-level interactions
> m5 <- lme(mathach ~ gceffic + public + csclimat + public*gceffic + csclimat*gceffic,
random = ~gceffic|SCH_ID, method="ML", data=chp10)
> summary(m5)
Linear mixed-effects model fit by maximum likelihood
 Data: chp10
      AIC       BIC    logLik
 73806.63   73878.6  -36893.31

Random effects:
 Formula: ~gceffic | SCH_ID
 Structure: General positive-definite, Log-Cholesky parametrization
              StdDev    Corr
(Intercept)  4.453850  (Intr)
gceffic      1.001821  0.583
Residual     9.713346

Fixed effects: mathach ~ gceffic + public + csclimat + public * gceffic + csclimat *
gceffic
                        Value  Std.Error   DF   t-value  p-value
(Intercept)          41.31079  0.4598111 9246  89.84297   0.0000
gceffic               4.22730  0.2792129 9246  15.14006   0.0000
publicpublic         -2.77430  0.5312182  614  -5.22252   0.0000
csclimat              3.00094  0.3297024  614   9.10198   0.0000
gceffic:publicpublic  0.52294  0.3237919 9246   1.61505   0.1063
gceffic:csclimat     -0.10002  0.2066567 9246  -0.48400   0.6284
 Correlation:
                     (Intr)  gceffc  pblcpb  csclmt  gcffc:p
gceffic               0.140
publicpublic         -0.892  -0.125
csclimat             -0.307  -0.058  0.351
gceffic:publicpublic -0.125  -0.889  0.148   0.060
gceffic:csclimat     -0.056  -0.333  0.058   0.162   0.368

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.90127367 -0.67502516  0.05711133  0.70307008  3.42698496

Number of Observations: 9866
Number of Groups: 617
```

Interpreting the Output

The fixed-effects section displays the intercept and the coefficients for the level 1 and level 2 predictor variables and two interaction terms. Let us take a look at the coefficients for the two interaction terms first since we are interested in the cross-level interactions. The coefficient for `gceffic:public` $(\gamma_{11}) = .523$, $t = 1.615$, $p > .05$, and the coefficient for `gceffic:csclimat` (γ_{12}) = −.100, t = −.484, p > .05, which means both interactions are not significant. What if there is a cross-level effect? How can it be interpreted? For example, if the interaction between `public` and `gceffic` is significant, then this effect can be interpreted as follows: The relationship between mathematics self-efficacy and mathematics achievement varies across school type.

We use the `intervals(m5, which = "fixed")` command to extract the 95% confidence intervals of the parameter estimates of the fixed effects. The results are displayed as follows.

```
> intervals(m5, which="fixed")
Approximate 95% confidence intervals

 Fixed effects:
                        lower         est.        upper
(Intercept)        40.4097378   41.3107948   42.2118519
gceffic             3.6801490    4.2273014    4.7744538
publicpublic       -3.8172044   -2.7742967   -1.7313890
csclimat            2.3536604    3.0009446    3.6482288
gceffic:publicpublic -0.1115691  0.5229414    1.1574518
gceffic:csclimat   -0.5049920   -0.1000226    0.3049469

 attr(,"label")
[1] "Fixed effects:"
```

To request the variance and covariance components of the model, we use the `VarCorr(m5)` command. The results are displayed as follows.

```
> VarCorr(m5)
SCH_ID = pdLogChol(gceffic)
              Variance    StdDev    Corr
(Intercept)  19.836777  4.453850  (Intr)
gceffic       1.003644  1.001821  0.583
Residual     94.349097  9.713346

> .583*1.002*4.454
[1] 2.601875
```

The variance and covariance components for the random effects for model 5 look similar to those for model 4, so they will not be interpreted here.

Model Comparisons Using the Likelihood Ratio Test

```
> anova(m4, m5)
   Model  df      AIC      BIC    logLik   Test  L.Ratio  p-value
m4     1   8 73806.56 73864.14 -36895.28
m5     2  10 73806.63 73878.60 -36893.31 1 vs 2 3.933261   0.1399
```

After two interaction terms are added to the model, we again use the `anova(m4, m5)` command to compare the log-likelihood statistics of the contextual models with or without interactions (model 5 vs. model 4). The likelihood ratio test $\chi^2_{(2)} = 3.93$, $p = .140$, which indicates that the contextual model with cross-level interactions does not fit the data better than the contextual model without interactions. Therefore, we should remove the interaction terms from the model.

10.4 MULTILEVEL MODELING FOR CONTINUOUS RESPONSE VARIABLES WITH THE `lmer()` FUNCTION IN THE `lme4` PACKAGE

10.4.1 The `lmer()` Function in the `lme4` Package

The `lmer()` function in the `lme4` package (Bates et al., 2015) can also be used for multilevel models for continuous response variables. Since `lme4` is a user-written package, you need to install it first by typing `install.packages("lme4")` and then load the package by typing `library(lme4)`. The model formula syntax of `lmer()` is similar to that of `lme()` introduced in the last section, but they are slightly different in specifying the random effects. The model formula in `lmer()` also includes two components connected by a plus (+) sign. The first component specifies the dependent variable and the predictor variable(s), which are separated by the tilde (~). When there are more than one predictor variable in the formula, they are connected by plus (+) symbols. This is the fixed effects component of the model. Next, the random effects component of the model is specified within parentheses, which is different from the `random` argument specified in `lme()` for linear mixed effects models. Within parentheses, a predictor variable or a list of predictor variables that have random coefficients is specified first, followed by a vertical line (|), and then, an identifier variable as the grouping variable at a higher level. The `REML = FALSE` argument requests the full maximum likelihood estimation rather than the default restricted likelihood (`REML = TRUE`). For example, the command `lmer(y ~ x + (1|schid), REML = FALSE, data = data1)` tells R to fit a multilevel model to estimate a continuous outcome variable `y` on a predictor variable `x` with random intercepts varying across schools by specifying `(1|schid)`, where `1` indicates the random intercept on the left of the vertical line (|) and `schid` is the identifier variable on the right. The `REML = FALSE` argument requests the maximum likelihood estimation. For more details on how to use this function, type `help(lmer)` in the command prompt after loading the `lme4` package.

We use the `lmerTest` package (Kuznetsova et al., 2017) to request the associated *p* values for the *t* tests of the parameter estimates. Since `lmerTest` is a user-written package, you need to install it first by typing `install.packages("lmerTest")` and then load the package by typing `library(lmerTest)`.

In the following example, the `m4.b <- lmer(mathach ~ gceffic + public + csclimat + (gceffic|SCH_ID), data= chp10, REML = FALSE)` command tells R to fit the same contextual model (model 4) in the previous section after the two level 2 predictor variables `public` and `csclimat` are added to the model. The fitted model is named `m4.b`. The following output is displayed by the `summary(m4.b)` command.

```
> # Contextual model with lmer()
> library(lmerTest)
> library(lme4)
> m4.b <- lmer(mathach ~ gceffic + public + csclimat + (gceffic|SCH_ID),
data=chp10, REML = FALSE)
> summary(m4.b)
Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
  method [lmerModLmerTest]
Formula: mathach ~ gceffic + public + csclimat + (gceffic | SCH_ID)
  Data: chp10

     AIC       BIC      logLik    deviance   df.resid
 73806.6   73864.1    -36895.3     73790.6       9858

Scaled residuals:
     Min        1Q    Median        3Q       Max
 -3.8808   -0.6741    0.0577    0.7033    3.4138

Random effects:
 Groups        Name      Variance    Std.Dev.    Corr
 SCH_ID     (Intercept)    19.777      4.447
             gceffic        1.077      1.038      0.56
 Residual                  94.357      9.714
Number of obs: 9866, groups: SCH_ID, 617
Fixed effects:
                Estimate   Std. Error        df    t value    Pr(>|t|)
(Intercept)      41.3965       0.4556  581.5800     90.860    < 2e-16 ***
gceffic           4.6270       0.1282  536.0261     36.086    < 2e-16 ***
publicpublic     -2.8969       0.5247  573.6036     -5.521    5.12e-08 ***
csclimat          3.0275       0.3249  570.2441      9.318    < 2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
              (Intr)    gceffc    pblcpb
gceffic        0.065
publicpublc   -0.890     0.014
csclimat      -0.304    -0.008     0.350
```

The R output produced by the `lmer()` function is similar to that by the `lme()` function introduced in the previous section. It includes the model fit statistics, the scaled residuals, the random effects, the number of observations and the number of groups, the fixed effects, and the correlations of the fixed effects.

The random effects section provides the variances and standard deviations for within-group, between-group, and the random coefficient of gceffic. It also provides the correlation coefficient for gceffic and the intercept. The results are the same as those provided by the VarCorr(m4) command in the previous section.

The fixed effects section looks the same as that produced by the lme() function. We use the confint(m4.b) command rather than the intervals() function to extract the 95% confidence intervals of the parameter estimates. The results are displayed as follows.

```
> confint(m4.b)
Computing profile confidence intervals ...
                    2.5 %      97.5 %
.sig01          4.1147172    4.805720
.sig02          0.2734630    1.000000
.sig03          0.4181463    1.493107
.sigma          9.5715169    9.859538
(Intercept)    40.4992243   42.291286
gceffic         4.3740861    4.880153
publicpublic   -3.9287910   -1.863101
csclimat        2.3894054    3.665517
```

The 95% confidence intervals for both the random effects and the fixed effects are provided. Please note that the confidence intervals for the random effects component are based on the standard deviations rather than the variances. The 95% confidence intervals for the fixed effects component are the same as those provided by the intervals(m4, which = "fixed") command introduced in the previous section.

10.4.2 Interpreting the Predicted Values With the ggpredict() Function in the ggeffects Package

By using the ggpredict() function in the ggeffects package (Lüdecke, 2018b), we can compute the predicted values of the continuous response variable at specified values of the predictor variables. We first load the package with library (ggeffects) since it has been installed in previous chapters. The command m4.b.pub <- ggpredict(m4.b, terms = "public") tells R to compute the predicted values of the response variable using the ggpredict() function. The argument inside the function includes the estimated model, m4.b, and the terms = "public" option which specifies the predictor variable public. The other predictor variables are held either at their means or at the reference level. The output is assigned to an object named m4.b.pub. The as.data.frame() function is used to request the standard errors.

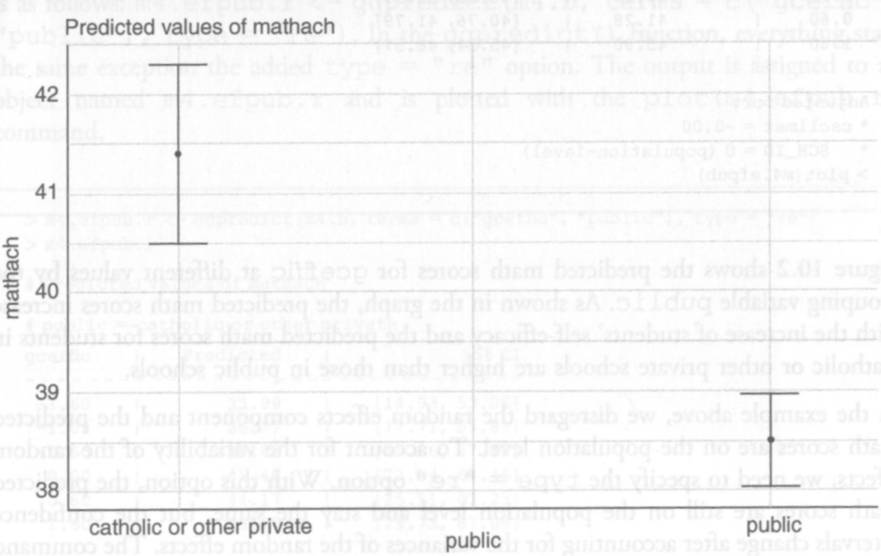
```
> library(ggeffects)
> m4.b.pub <- ggpredict(m4.b, terms = "public")
> m4.b.pub
```

```
# Predicted values of mathach
public                   |   Predicted   |         95% CI
-------------------------------------------------------
catholic or other private |     41.40    |   [40.50, 42.29]
public                   |     38.50    |   [38.03, 38.97]

Adjusted for:
* gceffic = 0.00
* csclimat = -0.00
* SCH_ID = 0 (population-level)
> as.data.frame(m4.b.pub)
                   x   predicted   std.error   conf.low   conf.high   group
1        catholic or   41.39651   0.4556121   40.50352    42.28949       1
       other private
2             public   38.49957   0.2395262   38.03010    38.96903       1
> plot(m4.b.pub)
Loading required namespace: ggplot2
```

The output displays the predicted math scores for both Catholic or other private schools and public schools and other predictor variables are fixed at their means. The last section under the title "`Adjusted for`" lists the means of the other three variables. For Catholic or other private schools, the predicted math scores = 41.397 and for public schools, the predicted math scores = 38.500.

The predicted math scores are plotted with `plot(m4.b.pub)`. Figure 10.1 shows the predicted math scores for both Catholic or other private schools and public schools.

FIGURE 10.1 ● Predicted Math Scores for `public` at 0 and 1 With Others Fixed at Their Means



As shown in the graph, the predicted math scores for students in Catholic or other private schools are higher than those in public schools.

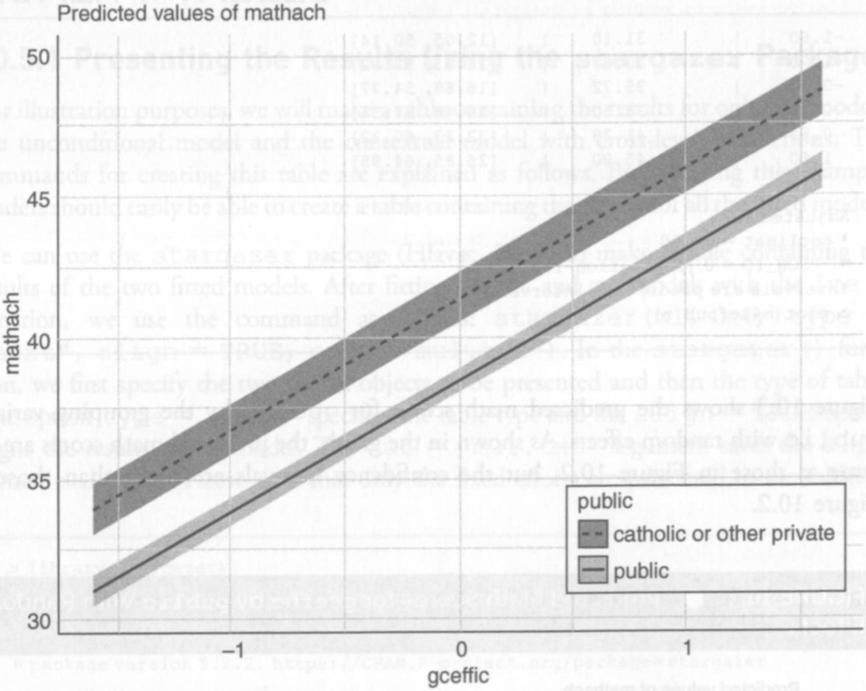
We can compute the predicted scores for a continuous variable at given values by different groups. In the following example, we compute the predicted math scores for `gceffic` at different values by the two groups in `public` when holding other variables at their means. The command is as follows: `m4.efpub <- ggpredict(m4.b, terms = c("gceffic", "public"))`. In the `ggpredict()` function, the `terms = c("gceffic", "public")` option specifies both `gceffic` and `public`, with the latter as the grouping variable. The output is assigned to an object named `m4.efpub` and is plotted with the `plot(m4.efpub)` command.

```
> m4.efpub <- ggpredict(m4.b, terms = c("gceffic", "public"))
> m4.efpub
# Predicted values of mathach
# public = catholic or other private

gceffic |  Predicted |        95% CI
-------------------------------------
  -1.60 |      33.99 | [33.04, 34.95]
  -1.00 |      36.77 | [35.86, 37.68]
  -0.60 |      38.62 | [37.72, 39.52]
   0.00 |      41.40 | [40.50, 42.29]
   0.60 |      44.17 | [43.26, 45.09]
   1.60 |      48.80 | [47.80, 49.80]

# public = public

gceffic |  Predicted |        95% CI
-------------------------------------
  -1.60 |      31.10 | [30.53, 31.67]
  -1.00 |      33.87 | [33.38, 34.37]
  -0.60 |      35.72 | [35.25, 36.19]
   0.00 |      38.50 | [38.03, 38.97]
   0.60 |      41.28 | [40.76, 41.79]
   1.60 |      45.90 | [45.24, 46.57]

Adjusted for:
* csclimat = -0.00
*   SCH_ID = 0 (population-level)
> plot(m4.efpub)
```

Figure 10.2 shows the predicted math scores for `gceffic` at different values by the grouping variable `public`. As shown in the graph, the predicted math scores increase with the increase of students' self-efficacy and the predicted math scores for students in Catholic or other private schools are higher than those in public schools.

In the example above, we disregard the random effects component and the predicted math scores are on the population level. To account for the variability of the random effects, we need to specify the `type = "re"` option. With this option, the predicted math scores are still on the population level and stay the same, but the confidence intervals change after accounting for the variances of the random effects. The command

FIGURE 10.2 ⬢ Predicted Math Scores for gceffic by public



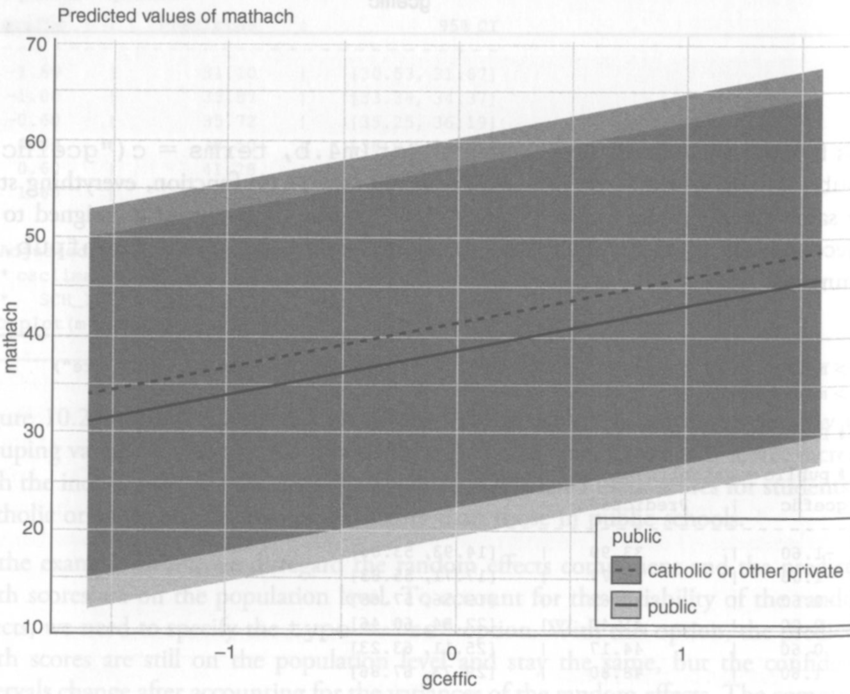
is as follows: m4.efpub.r <- ggpredict(m4.b, terms = c("gceffic", "public"), type = "re"). In the ggpredict() function, everything stays the same exception the added type = "re" option. The output is assigned to an object named m4.efpub.r and is plotted with the plot(m4.efpub.r) command.

```
> m4.efpub.r <- ggpredict(m4.b, terms = c("gceffic", "public"), type = "re")
> m4.efpub.r

# Predicted values of mathach

# public = catholic or other private
gceffic  |  Predicted  |          95% CI
-----------------------------------------
 -1.60   |      33.99  |  [14.93, 53.06]
 -1.00   |      36.77  |  [17.71, 55.83]
 -0.60   |      38.62  |  [19.56, 57.68]
  0.00   |      41.40  |  [22.34, 60.46]
  0.60   |      44.17  |  [25.11, 63.23]
  1.60   |      48.80  |  [29.73, 67.86]
```

```
# public = public
gceffic  |   Predicted   |          95% CI
---------------------------------------------
-1.60    |     31.10     |    [12.05, 50.14]
-1.00    |     33.87     |    [14.83, 52.92]
-0.60    |     35.72     |    [16.68, 54.77]
 0.00    |     38.50     |    [19.46, 57.54]
 0.60    |     41.28     |    [22.23, 60.32]
 1.60    |     45.90     |    [26.85, 64.95]

Adjusted for:
* csclimat = -0.00
*   SCH_ID = 0 (population-level)
Intervals are prediction intervals.
> plot(m4.efpub.r)
```

Figure 10.3 shows the predicted math scores for `gceffic` by the grouping variable `public` with random effects. As shown in the graph, the predicted math scores are the same as those in Figure 10.2, but the confidence intervals are wider than those in Figure 10.2.

FIGURE 10.3 ● Predicted Math Scores for `gceffic` by `public` With Random Effects



10.5 MAKING PUBLICATION-QUALITY TABLES

10.5.1 Presenting the Results Using the `stargazer` Package

For illustration purposes, we will make a table containing the results for only two models, the unconditional model and the contextual model with cross-level interactions. The commands for creating this table are explained as follows. By following this example, readers should easily be able to create a table containing the results for all the fitted models.

We can use the `stargazer` package (Hlavac, 2018) to make a table containing the results of the two fitted models. After fitting the `m1` and `m4` models with the `lme()` function, we use the command as follows: `stargazer(m1, m4, type = "text", align = TRUE, out = "mul.txt")`. In the `stargazer()` function, we first specify the two model objects to be presented and then the type of table. The option `type = "text"` specifies the table type and the `align = TRUE` option aligns the results of the model. The `out = "mul.txt"` argument saves the output named `mul.txt`. Please note that only the fixed effects are presented in the table.

```
> library(stargazer)
Please cite as:
 Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics
Tables.
 R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

> stargazer(m1, m4, type="text", align=TRUE, out="mul.txt")
===========================================================
                          Dependent variable:
                          ----------------------------
                                  mathach
                             (1)              (2)
-----------------------------------------------------------
gceffic                                     4.627***
                                            (0.128)

publicpublic                               -2.897***
                                            (0.525)

csclimat                                    3.027***
                                            (0.325)

Constant                   39.124***       41.397***
                            (0.254)         (0.456)

-----------------------------------------------------------
Observations                 9,866           9,866
Log Likelihood            -37,674.340     -36,895.280
Akaike Inf. Crit.          75,354.680      73,806.560
Bayesian Inf. Crit.        75,376.270      73,864.140
===========================================================
Note:                   *p<0.1; **p<0.05;     ***p<0.01
```

We can also create the table in the HTML format and copy it into Microsoft Word. The command is as follows: `stargazer(m1, m4, type = "html", align = TRUE, out = "mul.htm")`. It produces Table 10.1, as shown here in its original format, presenting the results for the unconditional model and the contextual model without cross-level interactions.

We can add the variances and covariance directly from the output of the unconditional model (model 1) and the contextual model (model 4). The edited table is displayed as Table 10.2.

10.5.2 Presenting the Results Using the texreg Package

The results can also be displayed in a table using the `screenreg()` and `htmlreg()` functions from the `texreg` package (Leifeld, 2013). Since `texreg`

TABLE 10.1 ● Results of the Two Multilevel Models: The Unconditional Model and the Contextual Model (Shown in Original Format Generated by R)

	Dependent variable:	
	mathach	
	(1)	(2)
gceffic		4.627***
		(0.128)
publicpublic		−2.897***
		(0.525)
csclimat		3.027***
		(0.325)
Constant	39.124***	41.397***
	(0.254)	(0.456)
Observations	9,866	9,866
Log Likelihood	−37,674.340	−36,895.280
Akaike Inf. Crit.	75,354.680	73,806.560
Bayesian Inf. Crit.	75,376.270	73,864.140

*$p < .1$
**$p < .05$
***$p < .01$.

TABLE 10.2 ● Results of the Two Multilevel Models: The Unconditional Model and the Contextual Model (Edited)

	Dependent variable:	
	mathach	
	Unconditional Model (Model 1)	Contextual Model (Model 4)
	Coefficient (SE)	Coefficient (SE)
Fixed Effects		
gceffic		4.627***
		(0.128)
publicpublic		−2.897***
		(0.525)
csclimat		3.027***
		(0.325)
Constant	39.124***	41.397***
	(0.254)	(0.456)
Random Effects		
Slope variance (gceffic)	–	1.077
Within-school variance (σ^2)	109.362	94.357
Between-group variance (τ_{00})	31.740	19.776
Covariance	–	2.608
Observations	9,866	9,866
Log Likelihood	−37,674.340	−36,895.280
Akaike Inf. Crit.	75,354.680	73,806.560
Bayesian Inf. Crit.	75,376.270	73,864.140

*$p < .1$
**$p < .05$
***$p < .01$

has been installed for previous chapters, we load the package by typing `library (texreg)`. We use the following command: `screenreg(list(m1, m4))`.

```
> library(texreg)
Version:    1.36.23
Date:       2017-03-03
Author:     Philip Leifeld (University of Glasgow)

Please cite the JSS article in your publications - see citation("texreg").
> screenreg(list(m1, m4))
=================================================
                     Model 1        Model 2
-------------------------------------------------
(Intercept)          39.12 ***      41.40 ***
                     (0.25)         (0.46)
gceffic                             4.63 ***
                                    (0.13)
publicpublic                        -2.90 ***
                                    (0.52)
csclimat                            3.03 ***
                                    (0.32)
-------------------------------------------------
AIC                  75354.68       73806.56
BIC                  75376.27       73864.14
Log Likelihood       -37674.34      -36895.28
Num. obs.            9866           9866
Num. groups          617            617
=================================================
*** p < 0.001,       ** p < 0.01,     * p < 0.05
```

To create the table in the HTML format and copy it into Microsoft Word, we use the command as follows: `htmlreg(list(m1, m4), file = "mul.doc", doctype = TRUE, html.tag = TRUE, head.tag = TRUE)`. The table is omitted here.

10.6 REPORTING THE RESULTS

Since multilevel models estimate the fixed effects and random effects, the results of both need to be reported. The following are the basic guidelines for reporting. Several common reporting guidelines provided in previous chapters can also be applied to the reporting for the multilevel modeling. Please note that what needs to be reported in the Results section in a research article and the formats for displaying results vary across disciplines and journals.

First, as with other research examples, describe the purpose of your study and explain why the multilevel modeling is needed for the analysis.

Second, if a series of nested models is fitted, then report model-building steps and briefly describe each model. Report and interpret the intraclass correlation coefficient.

Third, if necessary, report the results of the fitted models in a table including both the parameter estimates for the fixed effects and the variances and covariances for the random effects. If available, include deviance statistics (i.e., $-2LL$) and AIC and BIC statistics for these models in the table.

Fourth, report and interpret the fixed effects of the predictor variables and variances and covariances in the final model. The following is an example of summarized results for the unconditional model and the contextual model.

Multilevel modeling was used to examine the relationships between high-school students' mathematics achievement and mathematics self-efficacy, school type, and school climate. Five models, from the unconditional (null) model to the contextual model with cross-level interactions, were fitted. Table 10.2 presents the parameter estimates for the fixed effects and random effects for the fitted models. For illustration purposes, the following interpretations only focused on the results of the unconditional model (model 1) and the contextual model without cross-level interactions (model 4).

Results for the Unconditional Model

The between-school variance (τ_{00}) was 31.740, and the within-school variance (σ^2) was 109.362. The ICC = 31.740/(31.740 + 109.362) = .225, which indicated that 22.5% of the total variance was explained by schools in level 2. This empirical evidence showed that it was appropriate to use multilevel modeling for data analysis.

Results for the Contextual Model Without Cross-Level Interactions

The coefficient for `gceffic` was 4.627, $t = 36.079$, $p < .001$, which indicated that students with higher mathematics self-efficacy tended to have higher mathematics achievement when holding other predictors constant. The effects of two school-level predictor variables were also significant. The coefficient for `public` was −2.897, $t = -5.520$, $p < .01$, which indicated that students' mathematics scores in public schools tended to be lower than those in private schools. The coefficient for `csclimat` was 3.027, $t = 9.316$, $p < .001$, which indicated that schools with better social climate tended to have higher mathematics scores.

Regarding the random effects, the variance and covariance components are displayed in Table 10.2. After two school-level variables were included in the contextual model (model 4), the between-school variance (τ_{00}) decreased from 27.427 to 19.776 when compared with that for the random coefficient model (model 3): (27.427 − 19.776)/27.427 = 27.9%, which indicated that there was a decrease of 27.9% in the between-school variance from the random-coefficient model (model 3) to the contextual model without the cross-level interactions (model 4) after the two school-level variables were included.

10.7 SUMMARY OF R COMMANDS IN THIS CHAPTER

```
# Chap 10 R Script

# Remove all objects
rm(list = ls(all = TRUE))

# The following user-written packages need to be installed first by using
install.packages(" ") and then by loading it with library()

# library(nlme)                # It is part of R base distribution
# library(lmerTest)
# library(lme4)
# library(ggeffects)           # It is already installed for Chapter 2
# library(stargazer)           # It is already installed for Chapter 2
# library(texreg)              # It is already installed for Chapter 4

# Import the dataset
library(foreign)
chp10 <- read.dta("C:/CDA/els2002.dta")
chp10 <- chp10[!is.na(chp10$mathach)&!is.na(chp10$efficacy)&!is.na(chp10$
public)&!is.na(chp10$sclimate), ]
chp10$csclimat <- chp10$sclimate-mean(chp10$sclimate, na.rm=TRUE)
chp10$gceffic <- chp10$efficacy-mean(chp10$efficacy, na.rm=TRUE)
attach(chp10)

library(nlme)

# Null model with lme()
m1 <- lme(mathach ~ 1, random = ~1|SCH_ID, na.action="na.omit", method="ML",
data=chp10)
summary(m1)
intervals(m1)
VarCorr(m1)

# Random-intercept model
m2 <- lme(mathach ~ gceffic, random = ~1|SCH_ID, na.action="na.omit",
method="ML", data=chp10)
summary(m2)
intervals(m2)
VarCorr(m2)
```

```
# Model comparison
anova(m1, m2)

# Random-coefficient model
m3 <- lme(mathach ~ gceffic, random = ~gceffic|SCH_ID, na.action="na.omit",
method="ML", data=chp10)
summary(m3)
intervals(m3, which="fixed")
VarCorr(m3)
anova(m2, m3)

# Contextual model with predictor variables in both levels
m4 <- lme(mathach ~ gceffic + public + csclimat, random = ~gceffic|SCH_ID,
na.action="na.omit", method="ML", data=chp10)
summary(m4)
intervals(m4, which="fixed")
VarCorr(m4)
anova(m3, m4)

# Contextual model with cross-level interactions
m5 <- lme(mathach ~ gceffic + public + csclimat + public*gceffic + csclimat*gceffic,
random = ~gceffic|SCH_ID, method="ML", data=chp10)
summary(m5)
intervals(m5, which="fixed")
VarCorr(m5)

anova(m4, m5)

# Presenting the results with stargazer()
library(stargazer)
stargazer(m1, m4, type="text", align=TRUE, out="mul.txt")
stargazer(m1, m4, type="html", align=TRUE, out="mul.htm")

# Presenting the results with texreg()
library(texreg)
screenreg(list(m1, m4))
htmlreg(list(m1, m4), file="mul.doc", doctype=TRUE, html.tag=TRUE, head.tag=TRUE)

# Contextual model with lmer()
library(lmerTest)
library(lme4)
m4.b <- lmer(mathach ~ gceffic + public + csclimat + (gceffic|SCH_ID), data=chp10,
REML = FALSE)
summary(m4.b)
confint(m4.b)
confint(m4.b, method="Wald")

# Marginal effects/Predicted values with ggpredict() in ggeffects
library(ggeffects)
m4.b.pub <- ggpredict(m4.b, terms = "public")
m4.b.pub
as.data.frame(m4.b.pub)
plot(m4.b.pub)

m4.efpub <- ggpredict(m4.b, terms = c("gceffic", "public"))
m4.efpub
plot(m4.efpub)

m4.efpub.r <- ggpredict(m4.b, terms = c("gceffic", "public"), type = "re")
m4.efpub.r
plot(m4.efpub.r)

detach(chp10)
```

Glossary

Cross-level interactions in a two-level model involve including level 2 predictor variables in the equation for the random slope. In other words, there are interaction terms between level 1 and level 2 variables.

Fixed effects are the regression coefficients that estimate the relationships between the predictor variables and the outcome variable from the entire population.

Grand-mean centering involves subtracting the grand mean of the predictor variable from each value of the sample.

In group-mean centering we subtract the group mean from each value, which is the mean of each group or cluster where individuals are nested from each score.

In the random-intercept model the intercept is allowed to vary across groups or clusters.

Multilevel data, nested data, or hierarchical structured data have a data format in which observations at lower levels are nested within a higher level.

Random-coefficient models include both random intercept and random slopes. In addition to the random intercept, level 1 slopes (i.e., the coefficients of the level 1 predictors) can also be specified to be random.

Random effects are the randomly varying parameters across higher level units.

The contextual model is a special case of the random-intercept model or the random-coefficient model when both level 1 and level 2 predictor variables are included.

The intraclass correlation coefficient (ICC) is used to measure the proportion of variance in the outcome variable explained by groups or clusters. It is the ratio of the between-group variance to the total variance.

The purpose of centering is to make the results more interpretative. By subtracting the mean of a predictor variable from each value, we obtain a meaningful zero for the predictor variable.

The unconditional means model or the null model is known as the one-way random-effects ANOVA. Neither level 1 nor level 2 predictor variables are included in the model.

Exercises

Use the ELS:2002 data available at **https://edge.sagepub.com/liu1e** for the following problems. The following variables are used for the multilevel modeling.

`mathach`: mathematics IRT scores of high-school students

`gender`: gender (1 = female; 0 = male)

`byses`: socioeconomic status composite for base-year data.

We will conduct a study investigating the relationships between students' math achievement and the two student-level predictors. The outcome variable is `mathach`, and the predictor variables are `gender` and `byses`. The multilevel modeling for the continuous response variable will be used for data analysis. Answer the following questions or perform the following analyses:

1. Fit an unconditional model and obtain the between-group variance. Compute the ICC. What does it tell us?
2. Fit a two-level, random-intercept model with a random intercept and two student-level predictor variables `gender` and `byses`. Use the grand-mean centering for `byses` before fitting the model.
3. Interpret the fixed effects of the two predictor variables.
4. Conduct a likelihood ratio test comparing the random-intercept model and the unconditional model. Interpret the results.