Multilevel Modeling in R, Using the **nlme** Package

William T. Hoyt (University of Wisconsin-Madison)
David A. Kenny (University of Connecticut)
March 21, 2013

Supplement to Kenny, D. A., & Hoyt, W. (2009) Multiple levels of analysis in psychotherapy research, *Psychotherapy Research, 19,* 462-468.

R is a free, open-source statistical software package that may be downloaded from the Comprehensive R Archive Network (CRAN) at www.r-project.org . R is growing in popularity among researchers in both the social and physical sciences because of its flexibility and expandability. In the 20 years following the initial release, R users contributed over 5000 add-on “packages” to supplement the capabilities of the base system.

One such package is **nlme**, developed in the late 1990s by J. C. Pinheiro, D. M. Bates, and others to provide an extensive toolkit for testing both linear and non-linear mixed effects models in R. In this supplement, we show how to use the lme() and gls() functions to reproduce the models introduced by Kenny and Hoyt (2009), and also introduce some extractor functions that can operate on the output from lme() and gls(), and can assist users in interpreting multilevel relationships.

Notation

R can run in interactive mode (typing commands in a *console* window, each of which is evaluated when the user hits ENTER) or in batch mode (typing commands into a separate *script* file, then running one line or multiple lines at a time, with output displaying in the *console*). Although the script method is recommended for anything other than very simple analyses, we will use interactive notation to differentiate commands from output for this document:

- Lines beginning with “>” (console prompt) are commands (expressions to be evaluated).
- Lines beginning with “+” continue a command begun on the previous line.
- “#” is the comment character—text following this character is ignored by R.
- Lines not beginning with one of these characters are output.

R or SPSS commands, variable names, and output are displayed in this document in a fixed width font (`Courier`), and our commentary is displayed in Arial font. All the R statements are in a file that can be downloaded at davidakenny.net\papers\k&h\kh.R and R screen output at davidakenny.net\papers\k&h\kh.txt.

R Basics

Data are stored in what are called in R *data frames*. Similar to other statistical languages, rows represent cases (usually patients, in psychotherapy research) and columns represent variables. Many data frames can be open simultaneously in an R session. Thus, to avoid ambiguity, most modeling functions include a `data` argument, in which the user specifies the name of the data frame in which the variables of interest are stored (e.g., `data = nimh`).

Note that in R upper and lower case matters. So “`nimh`” and “`NIMH`” are not the same. The `nimh` dataset can be downloaded as a .csv (comma-separated values) file. To import the data set into R, the user might type:

```
> setwd("~/Downloads")           # establishes working directory
> nimh <- read.csv("nimh.csv")   # reads downloaded file
```

This places the data frame `nimh` in the workspace, so R can access it. Note though that `nimh` exists only in the workspace for this R session and cannot be accessed outside of R unless it is saved to some other format (e.g., csv or sav). You can download nimh.csv at davidakenny.net\papers\k&h\nimh.csv.

Functions. Commands in R take the form of functions, with the function name followed by one or more *arguments* (i.e., specifications) in parentheses. To see a list and description of the arguments for any function, use the `help()` function, with the name of the function you want to learn about in the parentheses (e.g., `help(lm)`) and the help file for that function will open in a new window. When the argument name is omitted R uses *positional evaluation*—the first item in the parentheses is assumed to refer to the first argument in the `help()` file, and so on. So we will sometimes omit argument names, especially for the first argument, in specifying function syntax.

Some functions are distributed in supplemental *packages*, which are free for download on the CRAN. For the R code shown here, readers will first need to download the `nlme` package, and then load it to make the included functions available for use in the current R session.

```
> install.packages(“nlme”)  # add this package to library
> library(nlme)      # makes functions in this package available
```

The `install.packages()` function only needs to be run once, to install `nlme` in the user’s R library. The `library()` function needs to be run once in a given R session prior to using functions in this package.

R formulas. Statistical modeling functions in R generally accept a *formula* as their first argument. The formula begins with the *response* or *outcome variable*, followed by the “`~`” operator (which may be read “is modeled by” or “is predicted from” or “is regressed on”), followed by a predictor variable (or a list of variables separated by “`+`”, “`*`”, or other operators). Thus, the formula `y ~ x1 + x2` indicates that `y` is to be jointly predicted from scores on `x1` and `x2`. The formula `y ~ x1 * x2` indicates that `y` is predicted from `x1`, `x2`, and their interaction (product `x1·x2`). In R, an “empty” (intercept-only; no predictors) model is specified as `y ~ 1`.

Output objects. Unlike many statistical languages, R statistical functions provide the option to save the result of the analysis in the workspace. This output object can then be used as input (i.e., as an argument) for other functions. Unlike SPSS, where any specialized output must be requested in the initial function call, in R one can query a saved output object for additional details.

R code for Examples in Kenny and Hoyt (2009)

Case 1: Individual therapy (patient nested within therapist)

Kenny and Hoyt (2009) provided SPSS syntax for a number of applications involving nested data structures in psychotherapy research. The first was an empty model for a study of psychotherapy outcomes with therapist as the nesting factor (i.e., patients nested within therapists) but ignoring treatment effects. This illustrates partitioning of outcome score variance into therapist (s_{τ}^2) and residual (s_{ε}^2) components. The intraclass correlation coefficient ($\hat{\rho}_I$) is the ratio of therapist variance to total variance:

$$\hat{\rho}_I = \frac{s_\tau^2}{s_\tau^2 + s_\varepsilon^2}$$

When only post-treatment scores (*y*) are available, the variance estimates are generated as follows.

SPSS syntax:

```
MIXED y
 /PRINT = SOLUTION TESTCOV
 /RANDOM INTERCEPT|SUBJECT(therapist).
```

Note. See SPSS output table “Covariance Parameters” for variance estimates.

When using the `lme()` and `gls()` functions it can be helpful to first convert the data frame to a “grouped data” object. This is accomplished with the `groupedData()` function, with arguments specifying a model formula (including the “|” operator to identify the main grouping variable), the data frame, and (optionally) any additional nesting variables (here, `tx` is an “outer” factor, as therapists are nested within treatments):

```
> nimh2 <- groupedData(y ~ tx | therapist, data=nimh,
+                      outer= ~tx)
```

The data in `nimh2` are identical to those in `nimh`. The basic formula is stored along with the data in `nimh2`, which can improve default behavior of graphical and statistical functions designed to work with nested data. Although this step is unnecessary for the main analyses shown here, it simplifies some operations, especially graphical analyses of multilevel data, and so it is good practice.

R code and output:

```
> # Model 1:  Ignoring ypre (pre-scores)
> mod1 <- lme(y ~ 1, data=nimh2, random = ~ 1|therapist,
+             na.action="na.omit")          # (empty model)
> summary(mod1)
Linear mixed-effects model fit by REML
 Data: nimh2
       AIC      BIC    logLik
  616.3183 623.5748 -305.1591

Random effects:
 Formula: ~1 | therapist
        (Intercept) Residual
StdDev:    2.785076 8.971209

Fixed effects: y ~ 1
               Value Std.Error DF  t-value p-value
(Intercept) 8.671625  1.200477 66 7.223484       0

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.4239908 -0.7030746 -0.2414293  0.4067050  4.3330192

Number of Observations: 84
Number of Groups: 18
```

Notes:

1. The first argument to `lme()` is called `fixed`, and takes the form of a formula (see above) defining the fixed effects portion of the model. In this case, the formula `y ~ 1` specifies an empty (intercept-only) model.
2. The second argument to `lme()` is `data`, and gives the name of the data frame where the variables are located.
3. The third argument to `lme()` is `random`, and specifies the random effects portion of the model. This is usually given as a one-sided formula (starting with "`~`")—here stipulating that the intercept is computed at the level of `therapist` (nesting factor), and that `therapist` is treated as a random variable.
4. Not all cases in this data frame have values for `y`. The `na.action` argument tells how we want to treat cases with missing values for one or more of the

variables in the model. (Missing values in R appear in the data frame as `NA`.) The action “`na.omit`” omits these cases from the analysis (listwise deletion), analyzing only those cases with complete data on all specified variables.

5. Variance components are found in the output under `Random effects` (the intercept is the therapist effect), but these are given as SDs rather than variances. The variance estimate is the square of the SD, so to compute $\hat{\rho}_I$ the user might type:

```
> 2.7851^2/(2.78512^2 + 8.9712^2)  # ICC
[1] 0.08790618
```

(Note that typing a mathematical expression followed by ENTER causes R to evaluate that expression.)

Because many R functions generate output in a form that can serve as input for future analyses, we can create a more elegant procedure for extracting the ICC from our mod1 output.

A little R programming:

```
 > VarCorr(mod1)
therapist = pdLogChol(1)
            Variance  StdDev
(Intercept)  7.756648 2.785076
Residual    80.482583 8.971209
```

`VarCorr()` is an example of an *extractor function* designed to pull specific information out of an object of class `lme`. When applied to `mod1`, `VarCorr()` displays a matrix of variance estimates and standard deviations for the two variance components in the model. However, these values are formatted as a *character* variable, so we need to change their format to *numeric* if we want to use them to compute the ICC.

```
> varests <- as.numeric(VarCorr(mod1)[1:2])
> # vector of variance estimates
> ICC <- varests[1]/sum(varests)  # computes ICC
> ICC                             # displays value
 [1] 0.08790476
```

This approach is convenient as it allows computation of the ICC without retyping the variance estimates themselves. R also allows *user-defined functions*, which can be helpful for automating large or small calculations. For example, we could define a function called `ICClme()` for computing ICCs from an object containing output from the `lme()` function:

```
> ICClme <- function(out) {
```

```
+   varests <- as.numeric(VarCorr(out)[1:2])
+   return(paste("ICC =", varests[1]/sum(varests)))
+ }
```

This procedure defines a new function `ICClme()`, which will be stored in the R workspace (and can later be saved and shared with other users). The function definition is enclosed within curly braces and provides instructions on how to operate on the values provided for the function's arguments. `ICClme()` has only a single argument, "`out`", which points to an `lme()` output object.

We can then apply this new function to `mod1` to compute the ICC:

```
> ICClme(mod1)
[1] "ICC = 0.0879047551989659"
```

Because this dataset (like many studies of psychotherapy outcomes) has baseline data as well as post-treatment data, Kenny and Hoyt also described a modification of the above procedure, to compute $\hat{\rho}_I$ controlling for baseline status on the outcome variable. This was accomplished by including pre-treatment scores (which were first centered, to enhance interpretability of the intercept) as a predictor in the fixed portion of the model.

SPSS syntax:

```
MIXED y WITH ypre
 /PRINT = SOLUTION TESTCOV
 /FIXED = ypre
 /RANDOM INTERCEPT|SUBJECT(therapist).
```

The variance estimates under "Covariance Parameters" were s_τ^2 = 5.62 and s_ε^2 = 77.86, yielding $\hat{\rho}_I$ = .068.

R code and output:

```
> nimh2$ypre.c <- with(nimh, ypre - mean(ypre))
> # ypre.c = centered pre-score
> mod2 <- lme(y ~ ypre.c, data=nimh2,
+             random  = ~ 1 | therapist,
+             na.action="na.omit")
> summary(mod2)
Linear mixed-effects model fit by REML
 Data: nimh2
       AIC      BIC    logLik
```

```
  615.5249 625.1517 -303.7624

Random effects:
 Formula: ~1 | therapist
        (Intercept) Residual
StdDev:    2.370091 8.823759

Fixed effects: y ~ ypre.c
               Value Std.Error DF  t-value p-value
(Intercept) 8.841337 1.1330053 65 7.803438  0.0000
ypre.c      0.291651 0.1273675 65 2.289840  0.0253
 Correlation:
       (Intr)
ypre.c 0.063

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.4292448 -0.6020348 -0.2625727  0.3880986  4.2534321

Number of Observations: 84
Number of Groups: 18
```

Note the only change from Model 1 was that `ypre.c` was added as a predictor to the fixed portion of the model. We see that `ypre.c` was a significant predictor of y (*B* = 0.29, *t* = 2.29 on *N* – *k* – 1 = 65 *df*, *p* = .025), where *N* is the number of observations and *k* is the number of groups (therapists).

Note that, as in Model 1, the "Tukey five-number" summary of the standardized residual values (bottom of output) suggests some asymmetry in the distribution of residuals—a signal that more detailed diagnostic tests are in order.

We can use the the `VarCorr()` function, and also `ICClme()` function to obtain the variance estimates and compute the ICC:

```
> VarCorr(mod2)
therapist = pdLogChol(1)
            Variance  StdDev
(Intercept)  5.617333 2.370091
Residual    77.858717 8.823759

> ICClme(mod2)
[1] "ICC = 0.067292750435604"
```

Kenny and Hoyt introduced a second method for computing $\hat{\rho}_I$, based on an alternative model of nonindependence which can allow for $\hat{\rho}_I$ to be negative (see Kenny, Mannetti, Pierro, Livi, & Kashy, 2002). This uses a repeated measures design, with correlated errors for patients within therapist.

SPSS syntax:

```
MIXED y BY patid
 /PRINT = SOLUTION TESTCOV
 /REPEATED patid|SUBJECT(therapist) COVTYPE(CSR).
```

The $\hat{\rho}_I$ estimate is found in the Covariance Parameters table as CSR rho.

Because there are no random variables, this model cannot be specified using lme(). Instead we use the gls() (generalized least squares) function, also from package nlme.

R code and output:

```
> # Model 3:  Pt’s as repeated measures for each therapist
> # (Allows for possibility that ICC < 0)
> mod3 = gls(y ~ 1, data=nimh2,
+            na.action = "na.omit", verbose = TRUE,
+            correlation=corCompSymm(form = ~1|therapist))
> summary(mod3)   # ICC (called "Rho") = .0879 (like mod1)
Generalized least squares fit by REML
  Model: y ~ 1
  Data: nimh2
       AIC      BIC    logLik
  616.3183 623.5748 -305.1591

Correlation Structure: Compound symmetry
 Formula: ~1 | therapist
 Parameter estimate(s):
       Rho
0.08790543

Coefficients:
               Value Std.Error  t-value p-value
(Intercept) 8.671625  1.200478 7.223476       0

Standardized residuals:
       Min          Q1         Med          Q3         Max
-0.9231442 -0.7102328 -0.2844098  0.4607804  4.2931871

Residual standard error: 9.393575
Degrees of freedom: 84 total; 83 residual
```

The ICC appears in the output as “Rho” under “Parameter estmate(s)” and agrees with the estimate derived from Model 1 (empty model; no pre-scores as predictors) above.

When the intraclass correlation is non-zero (i.e., when therapist variance contributes to differences among outcomes), the independence assumption of ANOVA is violated, and treatment differences should be tested using a model that accounts for interdependence of observations (i.e., correlated outcomes for patients with the same therapist). This can be accomplished by including treatment (`t`) as a predictor.

SPSS syntax:

```
MIXED y WITH tx
 /PRINT = SOLUTION TESTCOV
 /FIXED = tx
 /RANDOM INTERCEPT|SUBJECT(therapist).
```

Note. The coefficient for `tx` in the estimates of fixed effects reflects the difference between IPT (coded 1) and CBT (coded 0) means on `y`. This coefficient is *B* = -3.60, *t*(16.67) = -1.56, *p* = .14.

R code and output:

```
> # Model 4:  Including treatment effect
> mod4 <- lme(y ~ tx, data=nimh, random  = ~ 1 | therapist,
+             na.action = "na.omit")
> summary(mod4)
Linear mixed-effects model fit by REML
 Data: nimh
       AIC      BIC    logLik
  612.4669 622.0938 -302.2335

Random effects:
 Formula: ~1 | therapist
        (Intercept) Residual
StdDev:    2.366985 8.978599

Fixed effects: y ~ tx
                 Value Std.Error DF   t-value p-value
(Intercept)  10.698848  1.728778 66  6.188676  0.0000
txIPT        -3.596908  2.307256 16 -1.558955  0.1386
 Correlation:
      (Intr)
txIPT -0.749

Standardized Within-Group Residuals:
       Min          Q1         Med          Q3         Max 
-1.4889673 -0.6507164 -0.2521612  0.5105497  4.2079855 

Number of Observations: 84
Number of Groups: 18
```

Notes.

1. By default, character data are read into R as *factors*, a data type with special properties. One of these properties is that these variables are automatically dummy coded (with the first category (in alphabetical order) as the “reference” group) when this variable is included as a predictor in a statistical model. In this output, `txIPT` represents the contrast, in the variable `tx`, between the IPT group (coded 1) and the reference group (CBT, coded 0). The coefficient comparing IPT to CBT is *B* = -3.60, *t* = -1.56, *p* = .14.
2. Although this default (dummy) coding is a convenience, often the user may wish to create a specific coding structure, to enhance the interpretability of the output.
3. The `intervals()` function provides 95% confidence intervals for both fixed effects coefficients and random effects SDs.

```
> intervals(mod1)
Approximate 95% confidence intervals

 Fixed effects:
                lower      est.     upper
(Intercept)  7.214528 10.697007 14.179487
tIPT        -8.555691 -3.563684  1.428324
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: therapist 
                    lower     est.    upper
sd((Intercept)) 0.6374549 2.410756 9.117107

 Within-group standard error:
    lower      est.     upper 
 7.640768  9.027711 10.666411 
```

It is also possible to model cross-level interactions. For example, therapist (Level 2) may interact with a patient characteristic (e.g., p_gender, a Level 1 variable) in predicting outcomes. Some therapists could have better outcomes with male than female patients, whereas for other therapists the reverse could be true. Cross-level interactions are modeled by including the lower-level variable as a predictor of the higher level random effect.

SPSS syntax:

```
MIXED y WITH p_gender
 /FIXED = p_gender
 /PRINT = SOLUTION TESTCOV
 /RANDOM INTERCEPT p_gender|SUBJECT(therapist) COVTYPE(UN).
```

(This model did not converge in SPSS, probably due to the very small variance component for therapist—see below.)

R code and output:

```
> mod5 <- lme(y ~ 1, data=nimh2,
+             random  = ~ p_gender | therapist,
+             na.action = "na.omit")
> summary(mod5)
Linear mixed-effects model fit by REML
 Data: nimh2
       AIC      BIC   logLik
  620.2919 632.3861 -305.146

Random effects:
 Formula: ~p_gender | therapist
 Structure: General positive-definite, Log-Cholesky
parametrization
            StdDev       Corr
(Intercept) 0.0008973665 (Intr)
p_gender    1.4922377288 0.119
Residual    9.0056539030

Fixed effects: y ~ 1
               Value Std.Error DF t-value p-value
(Intercept) 8.610599  1.164232 66 7.39595       0

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.4506811 -0.7062470 -0.2252426  0.4125560  4.4500517

Number of Observations: 84
Number of Groups: 18
```

Note that when `p_gender` is taken into account, `therapist` variance drops to very near zero. The implication is that the therapist differences detected in Models 1 and 3 (ICC = 0.088) are attributable to differences in gender composition of the therapist’s caseload.

Case 2: Groups (patient nested within group nested within therapist)

In the case of group treatments, there are likely two levels of nesting that may contribute to covariances between outcomes: group and therapist. Kenny and Hoyt considered the case where group is nested within therapist (i.e., each therapist leads multiple groups, each of which includes multiple patients). In this case, we can estimate variance components attributable to therapist (s_{τ}^2), group (s_{ϕ}^2), and residual variance (s_{ε}^2). To estimate all three components, the model must specify both of the nesting factors as random effects.

SPSS syntax:

```
MIXED y
 /PRINT = SOLUTION TESTCOV
 /RANDOM INTERCEPT|SUBJECT(therapist)
 /RANDOM INTERCEPT|SUBJECT(group).
```

As before, the variance estimates are found in the Covariance Parameters table.

R code:

```
> lme(y ~ 1, data=grdata, random  = ~ 1 | therapist/group)
```

Notes.

1. The “/” operator denotes nesting of the two factors. The outermost factor is listed first, so “`therapist/group`” specifies two random effects (`therapist` and `group within therapist`).
2. As before, the variance components are scaled as SDs and reported in the `Random effects` section of the output. Do not forget to square them to convert to variances. (Because this is a 3-level model, the `ICC()` function that we created above will not perform correctly. It is possible to create a more general version of this function to compute ICCs for multilevel models.)

It is often of interest to include predictors in multilevel models. Predictors may be at the level of patient (often referred to as Level 1), group (Level 2), or therapist (Level 3). Kenny and Hoyt demonstrated a model that looks at patient gender (`p_gender`) and therapist gender (`t_gender`) and their interaction, as well as the gender composition of the group (`part_gen`).

SPSS syntax:

```
MIXED y WITH p_gender t_gender part_gen
 /FIXED = p_gender t_gender p_gender*t_gender part_gen
 /PRINT = SOLUTION TESTCOV
 /RANDOM INTERCEPT p_gender|SUBJECT(therapist) COVTYPE(UN)
 /RANDOM INTERCEPT|SUBJECT(group).
```

In this example, we model the interaction between p_gender and therapist (by including p_gender as a predictor of the random effect at Level 3), but not between p_gender and group.

R code:

```
> lme(y ~ p_gender*t_gender + part_gen, data=grdata,
+     random  = list(~ p_gender | therapist, ~ 1 | group))
```

Notes.

1. When the random effects model is different for different levels, the separate models should be specified as a list in R. The list should be ordered from the outermost (highest level) factor to the innermost.

Case 3: Groups with rolling membership

In this case we contemplate a study of ongoing groups with gradually changing membership, a relatively common situation in group counseling settings. Perhaps as the study begins, Group A has eight members, and as one departs ("graduates"), a new member joins, so that over the course of the study, we might have data for 20 different patients who have been members of Group A. The variable `order` indexes the order of entry into the group. In this situation, and autoregressive correlational structure models the assumption that outcomes are most similar for group members entering the group consecutively (difference of 1 on the `order` variable), and dependencies diminish as differences on the `order` variable increase.

SPSS syntax:

```
MIXED y BY order
 /PRINT = SOLUTION TESTCOV
 /RANDOM INTERCEPT|SUBJECT(therapist)
 /RANDOM INTERCEPT|SUBJECT(group)
 /REPEATED order|SUBJECT(group*therapist) COVTYPE(AR1).
```

This is an intercept-only model (no `/FIXED` predictors). It can be used to determine the ICC for group, where the ICC is modeled as a lag-1 autocorrelation to reflect the rolling membership of the groups.

R code:

```
> lme(y ~ 1, data=grdata, random  = ~ 1 | therapist/group,
+     correlation=corAR1(form=~order|therapist/group))
```

Similarly, in R we use `lme()` to model the nested random effects (`therapist` and `group`), and the `correlation` argument to specify the hypothesized correlational structure, which is dependent on the order of entry within each group.