Multilevel Multilingual

Multilevel Models in Stata, R and Julia

Andrew Grogan-Kaylor

2024-08-03

Table of contents

List of Tables

1 Multilevel Multilingual

> "This curious world which we inhabit is more wonderful than it is convenient…" (Thoreau, 1975)

> "Mathematics is my secret. My secret weakness. I feel like a stubborn, helpless fool in the middle of a problem. Trapped and crazed. Also, thrilled." (Schanen, 2021)

1.1 Introduction

Below, I describe the use of Stata (StataCorp, 2023), R (Bates et al., 2015; R Core Team, 2023), and Julia (Bates, 2024; Bezanson et al., 2017) to estimate multilevel models.

All of these software packages can estimate multilevel models and can visualize relationships in the data. However, there are substantial differences between the different packages: Stata is proprietary *for cost* software, which is very well documented and very intuitive. While it costs money to purchase Stata, the price is often very reasonable for academic and educational use. R is *free* open source software which is less intuitive, but there are many excellent resources for learning R. There is often a cost associated with purchasing books and other materials for learning R, which sometimes feels like it offsets the fact that R is free. Julia is newer open source software, also *free*, and ostensibly much faster than either Stata or R, which may be an important advantage when running multilevel models with very large data sets. At this point in time, both Stata and R feel much more *stable* than Julia which is still evolving software.

While any of these software packages can be used for learning and estimating multilevel models, I will offer my own opinion–based upon 15 years of teaching multilevel models at the doctoral level–that Stata offers the quickest pathway for learning the basic and advanced uses of multilevel models. I also believe the intuitive nature of Stata syntax contributes to accurate and replicable work in this area.

Table 1.1: Software for Multilevel Modeling

Software	Cost	Ease of Use
Stata	some cost	learning curve, but very intuitive for both multilevel modeling and graphing.

Software	Cost	Ease of Use
R	free	learning curve: intuitive for multilevel modeling; but steeper learning curve for graphing (`ggplot`).
Julia	free	steep learning curve in general: steep learning curve for multilevel modeling; and very steep learning curve for graphing. Graphics libraries are very much under development and in flux.

Results Will Vary Somewhat

Estimating multilevel models is a complex endeavor. The software details of how this is accomplished are beyond the purview of this book. Suffice it to say that across different software packages there will be differences in estimation routines, resulting in some numerical differences in the results provided by different software packages. Substantively speaking, however, results should agree across software.

Multi-Line Commands

Sometimes I have written commands out over multiple lines. I have done this for especially long commands, but have also sometimes done this simply for the sake of clarity. The different software packages have different approaches to multi-line commands.

1. By default, *Stata* ends a command at the end of a line. If you are going to write a multi-line command you should use the `///` line continuation characters.
2. *R* is the software that most naturally can be written using multiple lines, as R commands are usually clearly encased in parentheses (`()`) or continued with `+` signs.
3. Like *Stata*, *Julia* expects commands to end at the end of a line. If you are going to write a mult-line command, all commands except for the last line should end in a character that clearly indicates continuation, like a `+` sign. An alternative is to encase the entire Julia command in an outer set of parentheses (`()`).

Running Statistical Packages in Quarto

I used Quarto (Allaire et al., 2024) (https://quarto.org/) to create this Appendix. Quarto is a programming and publishing environment that can run multiple programming languages, including Stata, R and Julia, and that can write to multiple output formats including HTML, PDF, and MS Word. To run Stata, I used the `Statamarkdown` library (Hemken, 2023) in R to connect Stata to Quarto. Quarto has a built in connection to R,

and runs R without issue. To run Julia, I used the `JuliaCall` library (Li, 2019) in R to connect Quarto to Julia.
Of course, each of these programs can be run by itself, if you have them installed on your computer.

1.2 The Data

Datasets

The examples use the `simulated_multilevel_data.dta` and `simulated_multilevel_longitudinal_data.dta` files.
Here is a direct link to download the cross-sectional data.
Here is a direct link to download the longitudinal data.

Table 1.2: Sample of Simulated Multilevel Data

Table 1.2: Table continues below

country	HDI	family	id	identity	intervention	physical_punishment
1	69	1	1.1	1	0	3
1	69	2	1.2	1	1	2
1	69	3	1.3	0	1	3
1	69	4	1.4	1	0	0
1	69	5	1.5	1	0	4
1	69	6	1.6	0	1	5

Table 1.3: Sample of Simulated Multilevel Data

warmth	outcome
3	57.47
1	50.1
2	52.92
5	60.17
4	55.05
3	49.81

1.3 An Introduction To Equations and Syntax

To explain statistical syntax for each software, I consider the general case of a multilevel model with dependent variable `y`, independent variables `x` and `z`, clustering variable `group`, and a random slope for `x`. i is the index for the person, while j is the index for the group.

$$y = \beta_0 + \beta_1 x_{ij} + \beta_2 z_{ij} + u_{0j} + u_{1j} \times x_{ij} + e_{ij} \tag{1.1}$$

1.3.1 Stata

In Stata `mixed`, the syntax for a multilevel model of the form described in Equation 1.1 is:

```
mixed y x z || group: x
```

1.3.2 R

In R `lme4`, the syntax for a multilevel model of the form described in Equation 1.1 is:

```
library(lme4)

lmer(y ~ x + z + (1 + x || group), data = ...)
```

1.3.3 Julia

In Julia `MixedModels`, the syntax for a multilevel model of the form described in Equation 1.1 is:

```
using MixedModels

fit(MixedModel, @formula(y ~ x + z + (1 + x | group)), data)
```

2 Statistical Workflows

2.1 Statistical Software Is Best Run Using a Script

Many statistical workflows–whatever the statistical package being used–follow the same conceptual pattern.

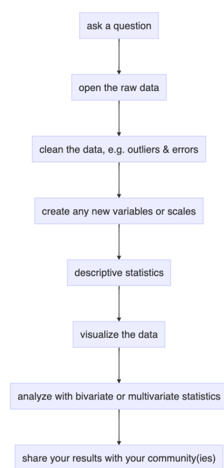

Figure 2.1: A Common Statistical Workflow

Increasingly, we want to think about workflows that are

- **documentable**, **transparent**, and **auditable**: We have a record of what we did if we want to double check our work, clarify a result, or develop a new project with a similar process. We, or others, can find the inevitable errors in our work, **and correct them**.
- **replicable**: Others can replicate our findings with the same or new data.
- **scalable**: We are developing a process that can be as easily used with *thousands* or *millions* of rows of data as it can with *ten* rows of data. We are developing a process that can be easily repeated if we are *constantly getting new or updated data*, e.g. getting new data every week, or every month.

2.2 Scripts

For most statistical workflows, we will often want to write a script or code. Data analysis scripts can be stored in a Quarto document (Allaire et al., 2024) as they are in this Appendix, or every statistical package has its own unique format for storing scripts as a text file: in Stata, scripts are stored in `.do` files; in R, scripts are stored in `.R` files, and in Julia, scripts are stored in `.jl` files.

2.3 Script Flow

A good practice when writing a script, is to have a script that begins with the raw data, moves through any necessary re-coding or cleaning of the data, generates descriptive statistics, generates the appropriate multivariate results, and then generates any necessary visualizations.

2.4 Storing Statistical Data

It is usually best to store quantitative data in a statistical format such as R (`.Rdata`), or Stata (`.dta`), or even a text format such as `.csv`. Spreadsheets are likely to be a bad tool for storing quantitative data.

2.5 Good Statistical Workflows Allow Multiple Statistical Packages

While this Appendix focuses on the use of each individual statistical package on its own, it is certainly possible to use multiple statistical packages as part of the same workflow. For example, one might employ Stata to carry out data management tasks, and then possibly use R to run a multilevel model with a more complicated multilevel structure, such as a cross-classified model, or Julia to more quickly run a model with a large data.

2.6 Good Statistical Workflows Require Safe Workspaces

It is also *very important* to be aware that good complex workflows are *highly iterative* and *highly collaborative*. Good complex workflows require a *safe workspace* in which team members feel free to admit their own errors, and help with others' mistakes in a non-judgmental fashion. Such a *safe environment* is necessary to build an environment where the *overall error rate* is low.

2.7 Good Statistical Workflows Require Patience And Can Be Psychologically Demanding

Developing a good documented and auditable workflow that is implemented in code requires a lot of patience, and often, **many iterations**. Working through these many iterations can be psychologically demanding. It is important to remember that careful attention to getting the details right early in the research process, while sometimes tiring and frustrating, will pay large dividends later on when the research is reviewed, presented, published and read.

2.8 Good Statistical Workflows Often Allow Multiple Principled Ways Forward

One of my most recent ideas about statistical workflows is that there are certainly *wrong* decisions that one can make with data.

For example, I would not want to write the paper that says that smoking prevents lung cancer, nor would I want to write a paper saying physical punishment is good for children.

That being said, I think there are often *multiple principled ways forward*.

Often the key is not so much to make the 100% correct decision, but to make one of *several possible principled decisions*.

Then after making a *principled decision*, one is *transparent* and *thorough* about describing the decision that one made.

For example, in implementing a multilevel analysis, I would have many choices: I could estimate only a random intercept; estimate one or more random slopes; or estimate all possible random slopes. The random effects could be correlated or uncorrelated. I could estimate only main effects, or could estimate interactions of several variables. Each of these would be a different, yet principled, approach to analyzing the data.

In science and statistics, we often want an answer that provides one clear direction. Instead, I'm increasingly convinced that the best science (and teaching!) often involves engaging in open discussion about the multiple possible alternatives, and then choosing one principled solution, and being transparent about its implementation.

3 Descriptive Statistics

3.1 Descriptive Statistics

3.1.1 Stata

```
use simulated_multilevel_data.dta // use data
```

We use `summarize` for *continuous* variables, and `tabulate` for *categorical* variables.

```
summarize outcome warmth physical_punishment HDI

tabulate identity

tabulate intervention
```

```
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
     outcome |      3,000    52.43327    6.530996   29.60798   74.83553
      warmth |      3,000    3.521667    1.888399          0          7
physical_p~t |      3,000    2.478667    1.360942          0          5
         HDI |      3,000    64.76667    17.24562         33         87


hypothetica |
 l identity |
      group |
   variable |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      1,507       50.23       50.23
          1 |      1,493       49.77      100.00
------------+-----------------------------------
      Total |      3,000      100.00
```

```
   recieved |
interventio |
          n |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      1,547       51.57       51.57
          1 |      1,453       48.43      100.00
------------+-----------------------------------
      Total |      3,000      100.00
```

3.1.2 R

```r
library(haven) # read data in Stata format

df <- read_dta("simulated_multilevel_data.dta")
```

R's descriptive statistics functions rely heavily on whether a variable is a *numeric* variable, or a *factor* variable. Below, I convert two variables to factors (`factor`) before using `summary`[1] to generate descriptive statistics.

```r
df$country <- factor(df$country)

df$identity <- factor(df$identity)

df$intervention <- factor(df$intervention)

summary(df)
```

```
    country          HDI            family             id              identity
 1      : 100   Min.   :33.00   Min.   :  1.00   Length:3000        0:1507
 2      : 100   1st Qu.:53.00   1st Qu.: 25.75   Class :character   1:1493
 3      : 100   Median :70.00   Median : 50.50   Mode  :character
 4      : 100   Mean   :64.77   Mean   : 50.50
 5      : 100   3rd Qu.:81.00   3rd Qu.: 75.25
 6      : 100   Max.   :87.00   Max.   :100.00
 (Other):2400
 intervention physical_punishment     warmth          outcome
 0:1547       Min.   :0.000       Min.   :0.000   Min.   :29.61
 1:1453       1st Qu.:2.000       1st Qu.:2.000   1st Qu.:48.02
```

[1] `skimr` is an excellent new alternative library for generating descriptive statistics in R.

```
Median :2.000      Median :4.000   Median :52.45
Mean   :2.479      Mean   :3.522   Mean   :52.43
3rd Qu.:3.000      3rd Qu.:5.000   3rd Qu.:56.86
Max.   :5.000      Max.   :7.000   Max.   :74.84
```

3.1.3 Julia

```julia
using Tables, MixedModels, MixedModelsExtras, StatFiles, DataFrames, CategoricalArrays, Datal

df = DataFrame(load("simulated_multilevel_data.dta"))
```

Similarly to R, Julia relies on the idea of *variable type*. I use `transform` to convert the appropriate variables to *categorical* variables.

```julia
@transform!(df, :country = categorical(:country))

@transform!(df, :identity = categorical(:identity))

@transform!(df, :intervention = categorical(:intervention))
```

```julia
describe(df) # descriptive statistics
```

```
9×7 DataFrame
 Row  variable            mean      min     median  max      nmissing  eltype
      Symbol              Union…    Any     Union…  Any      Int64     Union

   1  country                       1.0             30.0            0  Union{
   2  HDI                 64.7667   33.0    70.0    87.0            0  Union{
   3  family              50.5      1.0     50.5    100.0           0  Union{
   4  id                            1.1             9.99            0  Union{
   5  identity                      0.0             1.0             0  Union{
   6  intervention                  0.0             1.0             0  Union{
   7  physical_punishment 2.47867   0.0     2.0     5.0             0  Union{
   8  warmth              3.52167   0.0     4.0     7.0             0  Union{
   9  outcome             52.4333   29.608  52.449  74.8355         0  Union{
                                                                1 column omitted
```

3.2 Interpretation

Examining descriptive statistics is an important first step in any analysis. It is important to examine your descriptive statistics before skipping ahead to more sophisticated analyses, such as multilevel models.

In examining the descriptive statistics for this data, we get a sense of the data.

- `outcome` has a mean of approximately 52 and ranges from approximately 30 to 75.
- `warmth` and `physical punishment` are both variables that represent the number of times that parents use each of these forms of discipline in a week. The average of the former is about 3.5, while the average of the latter is about 2.5.
- `HDI`, the Human Development Index has an average of about 65, and a wide range.
- `identity` is a categorical variable for a hypothetical identity group, and has values of 0 and 1.
- `intervention` is also a categorical variable, and has values of 0 and 1.

4 Unconditional Models

4.1 Two Level Model

An *unconditional* multilevel model is a model with no independent variables. One should always run an unconditional model as the first step of a multilevel model in order to get a sense of the way that variation is apportioned in the model across the different levels.

4.1.1 The Equation

$$\text{outcome}_{ij} = \beta_0 + u_{0j} + e_{ij} \tag{4.1}$$

The Intraclass Correlation Coefficient (ICC) is given by:

$$\text{ICC} = \frac{var(u_{0j})}{var(u_{0j}) + var(e_{ij})} \tag{4.2}$$

In a two level multilevel model, the ICC provides a measure of the proportion of variation attributable to Level 2.

4.1.2 Run Models

4.1.2.1 Stata

```
use simulated_multilevel_data.dta // use data

mixed outcome || country:  // unconditional model

estat icc // ICC
```

```
Performing EM optimization ...

Performing gradient-based optimization:
Iteration 0:  Log likelihood = -9802.8371
Iteration 1:  Log likelihood = -9802.8371

Computing standard errors ...

Mixed-effects ML regression                     Number of obs    = 3,000
Group variable: country                         Number of groups =    30
                                                Obs per group:
                                                             min =   100
                                                             avg = 100.0
                                                             max =   100
                                                Wald chi2(0)     =     .
Log likelihood = -9802.8371                     Prob > chi2      =     .

------------------------------------------------------------------------------
     outcome | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       _cons |   52.43327   .3451217   151.93   0.000     51.75685     53.1097
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects parameters  |   Estimate   Std. err.     [95% conf. interval]
-----------------------------+------------------------------------------------
country: Identity            |
                  var(_cons) |   3.178658   .9226737      1.799552    5.614658
-----------------------------+------------------------------------------------
               var(Residual) |   39.46106   1.024013      37.50421       41.52
------------------------------------------------------------------------------
LR test vs. linear model: chibar2(01) = 166.31        Prob >= chibar2 = 0.0000


Intraclass correlation

------------------------------------------------------------------------------
                       Level |        ICC   Std. err.     [95% conf. interval]
-----------------------------+------------------------------------------------
                     country |   .0745469   .0201254      .0434963    .1248696
------------------------------------------------------------------------------
```

4.1.2.2 R

```r
library(haven)

df <- read_dta("simulated_multilevel_data.dta")
```

```r
library(lme4) # estimate multilevel models

fit0 <- lmer(outcome ~ (1 | country),
             data = df) # unconditional model

summary(fit0)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: outcome ~ (1 | country)
   Data: df

REML criterion at convergence: 19605.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.3844 -0.6655 -0.0086  0.6725  3.6626

Random effects:
 Groups   Name        Variance Std.Dev.
 country  (Intercept)  3.302   1.817
 Residual             39.461   6.282
Number of obs: 3000, groups:  country, 30

Fixed effects:
            Estimate Std. Error t value
(Intercept)   52.433      0.351   149.4
```

```r
library(performance)

performance::icc(fit0) # ICC
```

```
# Intraclass Correlation Coefficient

    Adjusted ICC: 0.077
  Unadjusted ICC: 0.077
```

4.1.2.3 Julia

```julia
using Tables, MixedModels, MixedModelsExtras,
StatFiles, DataFrames, CategoricalArrays, DataFramesMeta

df = DataFrame(load("simulated_multilevel_data.dta"))
```

```julia
@transform!(df, :country = categorical(:country))
```

```julia
m0 = fit(MixedModel,
         @formula(outcome ~ (1 | country)), df) # unconditional model
```

```
Linear mixed model fit by maximum likelihood
 outcome ~ 1 + (1 | country)
   logLik   -2 logLik     AIC       AICc        BIC
 -9802.8371 19605.6742 19611.6742 19611.6822 19629.6933

Variance components:
            Column   Variance Std.Dev.
country  (Intercept)   3.17863 1.78287
Residual              39.46106 6.28180
 Number of obs: 3000; levels of grouping factors: 30

  Fixed-effects parameters:

               Coef.  Std. Error       z  Pr(>|z|)

(Intercept)  52.4333    0.345121  151.93    <1e-99
```

```julia
icc(m0) # ICC
```

```
0.07454637475695493
```

4.1.3 Interpretation

In each case, the software finds that nearly 8% of the variation in the outcome is explainable by the clustering of the observations in each country.

4.2 Three Level Model

4.2.1 The Equation

$$\text{outcome}_{ij} = \beta_0 + u_{0j} + v_{0i} + e_{ij} \tag{4.3}$$

As discussed in the main text, in a three level model, there are two intraclass correlation coefficients (StataCorp, 2023). The formulas for the Intraclass Correlation Coefficient (ICC) are given by (StataCorp, 2023):

$$\text{ICC} = \frac{var(u_{0j})}{var(u_{0j}) + var(v_{0i}) + var(e_{ij})} \tag{4.4}$$

Following StataCorp (2023), Equation 4.4 is the correlation of responses for person-timepoints from the same country but different persons.

$$\text{ICC} = \frac{var(u_{0j}) + var(v_{0i})}{var(u_{0j}) + var(v_{0i}) + var(e_{ij})} \tag{4.5}$$

Again, closely following StataCorp (2023), Equation 4.5 is the correlation of responses for person-timepoints from the same country and same person.

4.2.2 Run Models

4.2.2.1 Stata

```
use simulated_multilevel_longitudinal_data.dta // use data

mixed outcome || country: || id: // unconditional model

estat icc // ICC
```

```
Performing EM optimization ...

Performing gradient-based optimization:
Iteration 0:  Log likelihood = -29058.266
Iteration 1:  Log likelihood = -29058.259
Iteration 2:  Log likelihood = -29058.259
```

```
Computing standard errors ...

Mixed-effects ML regression                         Number of obs = 9,000

        Grouping information
        -------------------------------------------------------------
                        |     No. of       Observations per group
         Group variable |     groups    Minimum    Average    Maximum
        ----------------+--------------------------------------------
                country |         30        300      300.0        300
                     id |      3,000          3        3.0          3
        -------------------------------------------------------------

                                                    Wald chi2(0)  =     .
Log likelihood = -29058.259                         Prob > chi2   =     .

------------------------------------------------------------------------------
     outcome | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       _cons |   53.37768   .3387943   157.55   0.000     52.71366    54.04171
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects parameters  |   Estimate   Std. err.     [95% conf. interval]
-----------------------------+------------------------------------------------
country: Identity            |
                  var(_cons) |   3.232092   .8891367      1.885043     5.54174
-----------------------------+------------------------------------------------
id: Identity                 |
                  var(_cons) |   11.72403   .5747501      10.64996    12.90641
-----------------------------+------------------------------------------------
               var(Residual) |   28.23424   .5154843      27.24178    29.26287
------------------------------------------------------------------------------
LR test vs. linear model: chi2(2) = 1314.88            Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.


Intraclass correlation

------------------------------------------------------------------------------
                       Level |        ICC   Std. err.     [95% conf. interval]
-----------------------------+------------------------------------------------
```

```
                    country |   .0748336   .0190847      .0450028    .1219141
                 id|country |   .3462837   .0171461      .3134867    .3806097
------------------------------------------------------------------------------
```

4.2.2.2 R

In R, the ICC for a three level model is easiest to estimate "by hand".

```r
library(haven)

dfL <- read_dta("simulated_multilevel_longitudinal_data.dta")
```

```r
library(lme4) # estimate multilevel models

fit0L <- lmer(outcome ~ (1 | country/id),
             data = dfL) # unconditional model

summary(fit0L)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: outcome ~ (1 | country/id)
   Data: dfL

REML criterion at convergence: 58116.8

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.7858 -0.6059 -0.0062  0.6017  3.4348

Random effects:
 Groups     Name        Variance Std.Dev.
 id:country (Intercept) 11.724   3.424
 country    (Intercept)  3.351   1.830
 Residual               28.234   5.314
Number of obs: 9000, groups:  id:country, 3000; country, 30

Fixed effects:
            Estimate Std. Error t value
(Intercept)  53.3777     0.3446   154.9
```

```
3.351 / (11.724 + 3.351 + 28.234)
```

```
[1] 0.07737422
```

```
(3.351 + 11.724) / (11.724 + 3.351 + 28.234)
```

```
[1] 0.3480801
```

4.2.2.3 Julia

In Julia, the ICC for a three level model is also easiest to estimate "by hand".

```
using Tables, MixedModels, StatFiles, DataFrames, CategoricalArrays, DataFramesMeta

dfL = DataFrame(load("simulated_multilevel_longitudinal_data.dta"))
```

```
@transform!(dfL, :country = categorical(:country))
```

```
m0L = fit(MixedModel, @formula(outcome ~
                                (1 | country) +
                                (1 | id)), dfL)
```

```
Linear mixed model fit by maximum likelihood
 outcome ~ 1 + (1 | country) + (1 | id)
   logLik   -2 logLik      AIC         AICc        BIC
 -29058.2592  58116.5184  58124.5184  58124.5229  58152.9384

Variance components:
            Column   Variance Std.Dev.
id       (Intercept)  11.72401 3.42403
country  (Intercept)   3.23190 1.79775
Residual              28.23426 5.31359
 Number of obs: 9000; levels of grouping factors: 3000, 30

  Fixed-effects parameters:

               Coef.  Std. Error       z  Pr(>|z|)

(Intercept)  53.3777    0.338785  157.56    <1e-99
```

```
3.23190 / (11.72401 + 3.23190 + 28.23426)
```

```
0.07482952718176382
```

```
(3.23190 + 11.72401) / (11.72401 + 3.23190 + 28.23426)
```

```
0.34628041519632824
```

4.2.3 Interpretation

Each software suggests that almost 8% of the variation in the outcome is within time points for different individuals within the same country, while almost 35% of the variation in the outcome is within time points for the same individual within the same country.

5 Cross Sectional Multilevel Models

5.1 The Equation

Recall the general model of Equation 1.1, and the syntax outlined in Section 1.3. Below in Equation 5.1, we consider a more substantive example.

$$\text{outcome}_{ij} = \beta_0 + \beta_1 \text{warmth}_{ij} + \tag{5.1}$$

$$\beta_2 \text{physical punishment}_{ij} +$$

$$\beta_3 \text{identity}_{ij} + \beta_4 \text{intervention}_{ij} + \beta_5 \text{HDI}_j +$$

$$u_{0j} + u_{1j} \times \text{warmth}_{ij} + e_{ij}$$

5.2 Correlated and Uncorrelated Random Effects

Consider the covariance matrix of random effects (e.g. u_{0j} and u_{1j}). In Equation 5.2 the covariances of the random effects are constrained to be zero.

$$\begin{bmatrix} var(u_{0j}) & 0 \\ 0 & var(u_{1j}) \end{bmatrix} \tag{5.2}$$

As discussed in the Chapter on multilevel models with cross-sectional data, however, one can consider a multilevel model in which the random effects are correlated, as is the case in Equation 5.3.

$$\begin{bmatrix} var(u_{0j}) & cov(u_{0j}, u_{1j}) \\ cov(u_{0j}, u_{1j}) & var(u_{1j}) \end{bmatrix} \tag{5.3}$$

Procedures for estimating models with uncorrelated and correlated random effects are detailed below (Bates et al., 2015; Bates, 2024; StataCorp, 2023).

Table 5.1: Correlated and Uncorrelated Random Effects

Software	Uncorrelated Random Effects	Correlated Random Effects
Stata	default	add option: `, cov(uns)`
R	separate random effects from grouping variable with `||`	separate random effects from grouping variable with `|`
Julia	separate terms for each random effect e.g. `(1 | group) + (0 + x | group)`	separate random effects from grouping variable with `|`.

All models in the examples below are run with *uncorrelated* random effects, but could just as easily be run with *correlated* random effects.

5.3 Run Models

Continuous and Categorical Variables

Statistically–as noted in the main text–it is important to be clear on whether independent variables in one's model are continuous or categorical. *Continuous* variables can be entered straightforwardly into statistical syntax. *Categorical* variables, on the other hand usually require specific attention in statistical software. In Stata, categorical variables are indicated in a statistical model by prefixing them with an `i.`. In R, categorical variables are distinguished by making them into factors e.g. `x <- factor(x)`. In Julia, categorical variables are created by using the `@transform` syntax detailed below.

5.3.1 Stata

5.3.1.1 Get The Data

```
use simulated_multilevel_data.dta
```

5.3.1.2 Run The Model

```
mixed outcome warmth physical_punishment i.identity i.intervention HDI || ///
country: warmth
```

```
Performing EM optimization ...

Performing gradient-based optimization:
Iteration 0:  Log likelihood = -9626.6279
Iteration 1:  Log likelihood =  -9626.607
Iteration 2:  Log likelihood =  -9626.607

Computing standard errors ...

Mixed-effects ML regression                     Number of obs    =  3,000
Group variable: country                         Number of groups =     30
                                                Obs per group:
                                                             min =    100
                                                             avg =  100.0
                                                             max =    100
                                                Wald chi2(5)     = 334.14
Log likelihood =  -9626.607                     Prob > chi2      = 0.0000

------------------------------------------------------------------------------------
            outcome | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
--------------------+---------------------------------------------------------------
             warmth |   .8345368   .0637213    13.10   0.000     .7096453    .9594282
physical_punishment |  -.9916657   .0797906   -12.43   0.000    -1.148052   -.8352791
         1.identity |  -.3004767   .2170295    -1.38   0.166    -.7258466    .1248933
     1.intervention |   .6396427   .2174519     2.94   0.003     .2134448    1.065841
                HDI |   -.003228   .0199257    -0.16   0.871    -.0422817    .0358256
              _cons |   51.99991   1.371257    37.92   0.000      49.3123    54.68753
------------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects parameters  |   Estimate   Std. err.     [95% conf. interval]
-----------------------------+------------------------------------------------
country: Independent         |
                 var(warmth) |   .0227504   .0257784      .0024689    .2096436
                  var(_cons) |   2.963975   .9737647      1.556777    5.643163
-----------------------------+------------------------------------------------
```

```
                  var(Residual) |    34.97499    .9097109      33.23668    36.80422
------------------------------------------------------------------------------
LR test vs. linear model: chi2(2) = 205.74                Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

5.3.2 R

5.3.2.1 Get The Data

```r
library(haven)

df <- read_dta("simulated_multilevel_data.dta")
```

5.3.2.2 Change Some Variables To Categorical

```r
df$identity <- factor(df$identity)

df$intervention <- factor(df$intervention)
```

5.3.2.3 Run The Model

> **Caution**
>
> `lme4` does not directly provide p values in results, because of some disagreement over exactly how these p values should be calculated. Therefore, in this Appendix, I also call library `lmerTest` to provide p values for `lme4` results.

> **Tip**
>
> R prefers to use scientific notation when possible. I find that the use of scientific notation can be confusing in reading results. I turn off scientific notation by setting a penalty for its use: `options(scipen = 999)`.

```r
library(lme4)

library(lmerTest)
```

```r
options(scipen = 999)

fit1 <- lmer(outcome ~ warmth + physical_punishment +
               identity + intervention + HDI +
               (1 + warmth || country),
             data = df)

summary(fit1)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: outcome ~ warmth + physical_punishment + identity + intervention +
    HDI + (1 + warmth || country)
   Data: df

REML criterion at convergence: 19268.8

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9774 -0.6563  0.0186  0.6645  3.6730

Random effects:
 Groups    Name        Variance Std.Dev.
 country   (Intercept)  3.19120 1.786
 country.1 warmth       0.02464 0.157
 Residual              35.01779 5.918
Number of obs: 3000, groups:  country, 30

Fixed effects:
                      Estimate  Std. Error          df t value
(Intercept)          52.011324    1.414976   30.293141  36.758
warmth                0.834562    0.064250   41.896457  12.989
physical_punishment  -0.991893    0.079845 2968.012381 -12.423
identity1            -0.300354    0.217179 2970.108153  -1.383
intervention1         0.639060    0.217603 2971.186718   2.937
HDI                  -0.003394    0.020598   27.592814  -0.165
                                Pr(>|t|)
(Intercept)         < 0.0000000000000002 ***
warmth               0.000000000000000277 ***
physical_punishment < 0.0000000000000002 ***
identity1                        0.16678
```

```
intervention1                    0.00334 **
HDI                              0.87030
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) warmth physc_ idntt1 intrv1
warmth      -0.124
physcl_pnsh -0.149 -0.003
identity1   -0.072 -0.012 -0.003
interventn1 -0.082  0.034  0.022 -0.018
HDI         -0.943 -0.006  0.009 -0.001  0.000
```

5.3.3 Julia

5.3.3.1 Get The Data

```julia
using Tables, MixedModels, StatFiles, DataFrames, CategoricalArrays, DataFramesMeta

df = DataFrame(load("simulated_multilevel_data.dta"))
```

5.3.3.2 Change Some Variables To Categorical

```julia
@transform!(df, :country = categorical(:country))

@transform!(df, :identity = categorical(:identity))

@transform!(df, :intervention = categorical(:intervention))
```

5.3.3.3 Run The Model

```julia
m1 = fit(MixedModel, @formula(outcome ~ warmth + physical_punishment +
               identity + intervention + HDI +
               (1 | country) +
               (0 + warmth | country)), df)
```

```
Linear mixed model fit by maximum likelihood
 outcome ~ 1 + warmth + physical_punishment + identity + intervention + HDI + (1 | country) +
   logLik   -2 logLik     AIC       AICc        BIC
 -9626.6070 19253.2140 19271.2140 19271.2742 19325.2713

Variance components:
            Column    Variance Std.Dev.   Corr.
country  (Intercept)   2.963849 1.721583
         warmth        0.022756 0.150852   .
Residual              34.974984 5.913965
 Number of obs: 3000; levels of grouping factors: 30

  Fixed-effects parameters:

                          Coef.  Std. Error       z  Pr(>|z|)

(Intercept)          51.9999      1.37124     37.92    <1e-99
warmth                0.834537    0.0637228   13.10    <1e-38
physical_punishment  -0.991665    0.0797906  -12.43    <1e-34
identity: 1.0        -0.300475    0.217029    -1.38    0.1662
intervention: 1.0     0.639641    0.217452     2.94    0.0033
HDI                  -0.0032286   0.0199255   -0.16    0.8713
```

5.4 Interpretation

Models suggest that parental warmth is associated with increases in the beneficial outcome, while physical punishment is associated with decreases in the beneficial outcome. The intervention is associated with increases in the outcome. There is insufficient evidence that either identity group or the Human Development Index are associated with the outcome.

6 Longitudinal Multilevel Models

6.1 The Data

The data employed in these examples are a longitudinal extension of the data described in Section 1.2.

6.2 The Equation

$$\text{outcome}_{itj} = \beta_0 + \beta_1 \text{parental warmth}_{itj} + \beta_2 \text{physical punishment}_{itj} + \beta_3 \text{time}_{itj} + \tag{6.1}$$

$$\beta_4 \text{identity}_{itj} + \beta_5 \text{intervention}_{itj} + \beta_6 \text{HDI}_j +$$

$$u_{0j} + u_{1j} \times \text{parental warmth}_{itj} +$$

$$v_{0ij} + v_{1ij} \times \text{time}_{itj} + e_{itj}$$

6.3 Growth Trajectories

Remember, following the discussion in the main text, that in longitudinal multilevel models, the variable for *time* assumes an important role as we are often thinking of a *growth trajectory* over time.

As discussed in the main text, think about a model where *identity* is a (1/0) variable for membership in one of two groups:

$$\text{outcome} = \beta_0 + \beta_t \text{time} + \beta_{\text{identity}} \text{identity} + \beta_{\text{interaction}} \text{identity} \times \text{time} + u_{0i} + e_{it}$$

Then, each identity group has its own intercept and time trajectory:

Table 6.1: Slope and Intercept for Each Group

Group	Intercept	Slope (Time Trajectory)
0	β_0	β_t
1	$\beta_0 + \beta_{\text{identity}}$	$\beta_t + \beta_{\text{interaction}}$

Main Effects and Interactions

Thus, again following the main text, in longitudinal multilevel models, *main effects* modify the *intercept* of the time trajectory, while *interactions with time*, modify the *slope* of the time trajectory. Below, we run models with *main effects only*, then models with *main effects, and interactions with time.*

6.4 Run Models

Warning

Remember that we are estimating a model in which time points are nested inside families, who are in turn nested inside countries. For each software package, it is accordingly important to specify the way in which different levels of the data are nested. Pay careful attention to the syntax examples below with regard to `id` and `country`

6.4.1 Stata

6.4.1.1 Get The Data

```
use simulated_multilevel_longitudinal_data.dta
```

6.4.1.2 Run The Models

6.4.1.2.1 Main Effects Only

```
mixed outcome t warmth physical_punishment i.identity i.intervention HDI || ///
country: warmth || id: t
```

```
Performing EM optimization ...

Performing gradient-based optimization:
Iteration 0:  Log likelihood =  -28523.49
Iteration 1:  Log likelihood = -28499.987
Iteration 2:  Log likelihood = -28499.739
Iteration 3:  Log likelihood = -28499.604
Iteration 4:  Log likelihood = -28499.603

Computing standard errors ...

Mixed-effects ML regression                         Number of obs =   9,000

        Grouping information
        -------------------------------------------------------------
                        |     No. of       Observations per group
         Group variable |     groups    Minimum    Average    Maximum
        ----------------+--------------------------------------------
                country |         30        300      300.0        300
                     id |      3,000          3        3.0          3
        -------------------------------------------------------------

                                                    Wald chi2(6)  = 1096.15
Log likelihood = -28499.603                         Prob > chi2   =  0.0000

------------------------------------------------------------------------------------
           outcome | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------------+----------------------------------------------------------------
                 t |    .943864   .0658716    14.33   0.000      .814758     1.07297
            warmth |   .9134959   .0423732    21.56   0.000      .830446    .9965459
physical_punishment |  -1.007897   .0497622   -20.25   0.000    -1.105429   -.9103647
        1.identity |  -.1276926   .1515835    -0.84   0.400    -.4247908    .1694057
    1.intervention |   .8589966   .1519095     5.65   0.000     .5612596    1.156734
               HDI |  -.0005657   .0196437    -0.03   0.977    -.0390666    .0379352
             _cons |   50.46724   1.338318    37.71   0.000     47.84418    53.09029
------------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects parameters  |   Estimate   Std. err.     [95% conf. interval]
-----------------------------+------------------------------------------------
country: Independent         |
                 var(warmth) |   .0107586   .0127845      .0010478    .1104703
                  var(_cons) |   3.167085   .9146761      1.798154    5.578181
```

```
-----------------------------+------------------------------------------------
id: Independent              |
                      var(t) |   3.58e-09   7.06e-07      3.5e-177    3.7e+159
                  var(_cons) |   8.387275   .4724188      7.510631    9.366242
-----------------------------+------------------------------------------------
               var(Residual) |   26.02733   .4753701      25.11211    26.97592
------------------------------------------------------------------------------
LR test vs. linear model: chi2(4) = 1247.03               Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.
```

6.4.1.2.2 Interactions With Time

```
mixed outcome c.t##(c.warmth c.physical_punishment i.identity i.intervention c.HDI) || count
```

```
Performing EM optimization ...

Performing gradient-based optimization:
Iteration 0:  Log likelihood =  -28522.21
Iteration 1:  Log likelihood = -28498.677
Iteration 2:  Log likelihood = -28498.468
Iteration 3:  Log likelihood =  -28498.31
Iteration 4:  Log likelihood = -28498.309

Computing standard errors ...

Mixed-effects ML regression                            Number of obs =   9,000

        Grouping information
        -------------------------------------------------------------
                        |     No. of       Observations per group
         Group variable |     groups    Minimum    Average    Maximum
        ----------------+--------------------------------------------
                country |         30        300      300.0        300
                     id |      3,000          3        3.0          3
        -------------------------------------------------------------

                                                       Wald chi2(11) = 1100.25
Log likelihood = -28498.309                            Prob > chi2   =  0.0000

-----------------------------------------------------------------------------------------
```

outcome	Coefficient	Std. err.	z	P>|z|	[95% conf. interval]	
t	.7582075	.326177	2.32	0.020	.1189123	1.397503
warmth	.8170757	.082662	9.88	0.000	.6550611	.9790903
physical_punishment	-1.009031	.1112932	-9.07	0.000	-1.227162	-.7909007
1.identity	-.2387167	.3039964	-0.79	0.432	-.8345387	.3571053
1.intervention	.6607606	.3044503	2.17	0.030	.064049	1.257472
HDI	.0013614	.0210842	0.06	0.949	-.0399628	.0426856
c.t#c.warmth	.0483637	.0356074	1.36	0.174	-.0214255	.1181529
c.t#c.physical_punishment	.0005421	.0494354	0.01	0.991	-.0963496	.0974338
identity#c.t						
1	.0554389	.1317444	0.42	0.674	-.2027754	.3136532
intervention#c.t						
1	.0992811	.131925	0.75	0.452	-.1592872	.3578493
c.t#c.HDI	-.0009551	.0038216	-0.25	0.803	-.0084453	.0065352
_cons	50.83632	1.483548	34.27	0.000	47.92862	53.74402

Random-effects parameters	Estimate	Std. err.	[95% conf. interval]	
country: Independent				
var(warmth)	.0106014	.0127458	.0010046	.1118779
var(_cons)	3.170088	.9153355	1.80009	5.582753
id: Independent				
var(t)	9.47e-10	2.07e-07	1.5e-195	6.0e+176
var(_cons)	8.39189	.4724106	7.515234	9.370809
var(Residual)	26.01583	.4751602	25.10101	26.964

LR test vs. linear model: chi2(4) = 1247.84 Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.

6.4.2 R

6.4.2.1 Get The Data

```
library(haven)

dfL <- read_dta("simulated_multilevel_longitudinal_data.dta")
```

6.4.2.2 Change Some Variables To Categorical

```
dfL$identity <- factor(dfL$identity)

dfL$intervention <- factor(dfL$intervention)
```

6.4.2.3 Run The Models

> **Caution**
>
> `lme4` does not directly provide p values in results, because of some disagreement over exactly how these p values should be calculated. Therefore, in this Appendix, I also call library `lmerTest` to provide p values for `lme4` results.

> **Tip**
>
> R prefers to use scientific notation when possible. I find that the use of scientific notation can be confusing in reading results. I turn off scientific notation by setting a penalty for its use: `options(scipen = 999)`.

6.4.2.3.1 Main Effects Only

```
library(lme4)

library(lmerTest)

options(scipen = 999)

fit2A <- lmer(outcome ~ t + warmth + physical_punishment +
```

```r
               identity + intervention + HDI +
               (1 | country/id),
             data = dfL)

summary(fit2A)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula:
outcome ~ t + warmth + physical_punishment + identity + intervention +
    HDI + (1 | country/id)
   Data: dfL

REML criterion at convergence: 57022.7

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.6850 -0.6094 -0.0035  0.6133  3.6792

Random effects:
 Groups     Name        Variance Std.Dev.
 id:country (Intercept)  8.438   2.905
 country    (Intercept)  3.675   1.917
 Residual               26.036   5.103
Number of obs: 9000, groups:  id:country, 3000; country, 30

Fixed effects:
                        Estimate   Std. Error          df t value
(Intercept)           50.3842343    1.4139114   29.8246912  35.635
t                      0.9433806    0.0658755 5998.3764548  14.321
warmth                 0.9140307    0.0379336 4745.3497493  24.096
physical_punishment   -1.0087537    0.0497972 6483.6771808 -20.257
identity1             -0.1319548    0.1517350 2968.7828107  -0.870
intervention1          0.8591494    0.1520510 2971.8111995   5.650
HDI                    0.0007909    0.0207656   28.0001855   0.038
                                Pr(>|t|)
(Intercept)         < 0.0000000000000002 ***
t                   < 0.0000000000000002 ***
warmth              < 0.0000000000000002 ***
physical_punishment < 0.0000000000000002 ***
identity1                          0.385
intervention1               0.0000000175 ***
```

```
HDI                                0.970
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) t      warmth physc_ idntt1 intrv1
t           -0.092
warmth      -0.091 -0.002
physcl_pnsh -0.092 -0.007 -0.012
identity1   -0.051  0.000 -0.013 -0.003
interventn1 -0.058  0.000  0.039  0.019 -0.018
HDI         -0.951  0.000 -0.004  0.005  0.000  0.002
```

6.4.2.3.2 Interactions With Time

```
fit2B <- lmer(outcome ~ t *(warmth + physical_punishment +
               identity + intervention + HDI) +
               (1 | country/id),
             data = dfL)

summary(fit2B)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula:
outcome ~ t * (warmth + physical_punishment + identity + intervention +
    HDI) + (1 | country/id)
   Data: dfL

REML criterion at convergence: 57042.8

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.7118 -0.6092 -0.0024  0.6150  3.6779

Random effects:
 Groups     Name        Variance Std.Dev.
 id:country (Intercept)  8.436   2.905
 country    (Intercept)  3.675   1.917
 Residual               26.046   5.104
Number of obs: 9000, groups:  id:country, 3000; country, 30
```

```
Fixed effects:
                        Estimate   Std. Error          df t value
(Intercept)           50.7590272    1.5518360   43.2608620  32.709
t                      0.7552909    0.3263028 6176.7440549   2.315
warmth                 0.8170912    0.0805355 8274.9995422  10.146
physical_punishment   -1.0097729    0.1113557 8084.6084915  -9.068
identity1             -0.2446453    0.3041604 8695.8966197  -0.804
intervention1          0.6604671    0.3046286 8697.0843430   2.168
HDI                    0.0026692    0.0221295   36.1037733   0.121
t:warmth               0.0486211    0.0356217 6404.8723333   1.365
t:physical_punishment  0.0004964    0.0494590 6753.0158441   0.010
t:identity1            0.0563140    0.1318043 5993.4518022   0.427
t:intervention1        0.0995037    0.1319917 5994.1433001   0.754
t:HDI                 -0.0009379    0.0038233 5993.9090880  -0.245
                                 Pr(>|t|)
(Intercept)           <0.0000000000000002 ***
t                                  0.0207 *
warmth                <0.0000000000000002 ***
physical_punishment   <0.0000000000000002 ***
identity1                          0.4212
intervention1                      0.0302 *
HDI                                0.9047
t:warmth                           0.1723
t:physical_punishment              0.9920
t:identity1                        0.6692
t:intervention1                    0.4510
t:HDI                              0.8062
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) t      warmth physc_ idntt1 intrv1 HDI    t:wrmt t:phy_
t           -0.421
warmth      -0.178  0.331
physcl_pnsh -0.190  0.360 -0.005
identity1   -0.093  0.166 -0.013 -0.002
interventn1 -0.107  0.192  0.039  0.019 -0.017
HDI         -0.925  0.264 -0.007  0.012 -0.001  0.003
t:warmth     0.158 -0.377 -0.882  0.001  0.011 -0.035  0.006
t:physcl_pn  0.170 -0.402  0.004 -0.894 -0.001 -0.017 -0.010 -0.003
t:identity1  0.081 -0.192  0.011  0.000 -0.867  0.014  0.001 -0.013  0.002
t:intrvntn1  0.093 -0.222 -0.035 -0.017  0.014 -0.867 -0.003  0.041  0.019
```

```
t:HDI         0.322 -0.765  0.015 -0.027  0.002 -0.007 -0.346 -0.016  0.029
            t:dnt1 t:ntr1
t
warmth
physcl_pnsh
identity1
interventn1
HDI
t:warmth
t:physcl_pn
t:identity1
t:intrvntn1 -0.016
t:HDI       -0.002  0.008
```

6.4.3 Julia

6.4.3.1 Get The Data

```julia
using Tables, MixedModels, StatFiles, DataFrames, CategoricalArrays, DataFramesMeta

dfL = DataFrame(load("simulated_multilevel_longitudinal_data.dta"))
```

6.4.3.2 Change Some Variables To Categorical

```julia
@transform!(dfL, :country = categorical(:country))

@transform!(dfL, :identity = categorical(:identity))

@transform!(dfL, :intervention = categorical(:intervention))
```

6.4.3.3 Run The Models

6.4.3.3.1 Main Effects Only

```julia
m2A = fit(MixedModel, @formula(outcome ~ t + warmth +
                                 physical_punishment +
                                 identity + intervention +
                                 HDI +
```

```
                                (1 | country) +
                                (0 + warmth | country) +
                                (1 | id)), dfL)
```

```
Linear mixed model fit by maximum likelihood
 outcome ~ 1 + t + warmth + physical_punishment + identity + intervention + HDI + (1 | count
    logLik   -2 logLik      AIC         AICc        BIC
 -28499.6031  56999.2063  57021.2063  57021.2356  57099.3610

Variance components:
            Column    Variance Std.Dev.   Corr.
id       (Intercept)   8.387214 2.896069
country  (Intercept)   3.167143 1.779647
         warmth        0.010762 0.103739   .
Residual              26.027363 5.101702
 Number of obs: 9000; levels of grouping factors: 3000, 30

  Fixed-effects parameters:

                           Coef.  Std. Error       z  Pr(>|z|)

(Intercept)          50.4673        1.33833    37.71    <1e-99
t                     0.943864      0.0658717  14.33    <1e-45
warmth                0.913496      0.0423744  21.56    <1e-99
physical_punishment  -1.0079        0.0497622 -20.25    <1e-90
identity: 1.0        -0.127692      0.151583   -0.84    0.3996
intervention: 1.0     0.858997      0.151909    5.65    <1e-07
HDI                  -0.000566026   0.0196439  -0.03    0.9770
```

6.4.3.3.2 Interactions With Time

```
m2B = fit(MixedModel, @formula(outcome ~ t * (warmth +
                                physical_punishment +
                                identity + intervention +
                                  HDI) +
                                (1 | country) +
                                (0 + warmth | country) +
                                (1 | id)), dfL)
```

```
Linear mixed model fit by maximum likelihood
```

```
 outcome ~ 1 + t + warmth + physical_punishment + identity + intervention + HDI + t & warmth
    logLik   -2 logLik      AIC         AICc        BIC
 -28498.3091  56996.6182  57028.6182  57028.6788  57142.2979

Variance components:
            Column    Variance Std.Dev.   Corr.
id       (Intercept)   8.391746 2.896851
country  (Intercept)   3.170032 1.780458
         warmth        0.010609 0.102999   .
Residual              26.015906 5.100579
 Number of obs: 9000; levels of grouping factors: 3000, 30

  Fixed-effects parameters:

                                Coef.  Std. Error      z  Pr(>|z|)

(Intercept)              50.8364       1.48355     34.27    <1e-99
t                         0.758209     0.326177     2.32    0.0201
warmth                    0.817076     0.0826636    9.88    <1e-22
physical_punishment      -1.00903      0.111293    -9.07    <1e-18
identity: 1.0            -0.238714     0.303996    -0.79    0.4323
intervention: 1.0         0.660761     0.30445      2.17    0.0300
HDI                       0.00136065   0.0210842    0.06    0.9485
t & warmth                0.0483635    0.0356074    1.36    0.1744
t & physical_punishment   0.000542203  0.0494355    0.01    0.9912
t & identity: 1.0         0.0554385    0.131745     0.42    0.6739
t & intervention: 1.0     0.0992809    0.131925     0.75    0.4517
t & HDI                  -0.000955067  0.00382162  -0.25    0.8027
```

6.5 Interpretation

The *main effects only model* suggests that time is associated with increases in the outcome. In the main effects model, main effects other than time, indicate whether a particular variable is associated with higher or lower intercepts of the time trajectory, at the beginning of the study time. Warmth is again associated with increases in the outcome, while physical punishment is associated with decreases in the outcome. Identity is again not associated with the outcome, while the intervention is associated with higher levels of the outcome. The Human Development Index is again not associated with the outcome.

The second model adds interactions with time to the first model. Results are largely similar to the prior model. However, here we not only examine whether main effects other than time are associated with higher or lower time trajectories, but also whether particular variables are associated with differences in the slope of the time trajectory. In this case, we find insufficient evidence that any independent variable is associated with changes in the slope of the time trajectory.

> **Which Interactions To Test?**
>
> In this example–for the sake of illustration–I test the interaction of *every* independent variable with time. In many cases, it may make sense to test only only one or two interactions of time with particular variables of key interest. Also, after finding, as I did in this model, that none of the interactions of other independent variables with time are significant, I might report the model with interactions, or might report only the results of the model with only main effects.

It may be illustrative to imagine how we would interpret the results had a particular interaction term been statistically significant. Let us consider one of the interaction terms with the largest coefficient, `intervention#time`. The interaction of the intervention with time is positive. Had this coefficient been statistically signifcant, it would have indicated that the intervention was associated with more rapid increases in the outcome over time *in addition to* the fact that the intervention is associated with higher initial levels of the outcome.

7 Multilevel Logistic Regression

Below, I detail the procedure for multilevel logistic regression models in Stata and R.

7.1 The Data

The data employed in these examples are the cross-sectional data described in Section 1.2.

7.2 The Equation

To explain statistical syntax for Stata and R, I consider the general case of a multilevel model with *categorical* dependent variable `y`, independent variables `x` and `z`, clustering variable `group`, and a random slope for `x`. i is the index for the person, while j is the index for the group.

$$\ln\left(\frac{p(y)}{1-p(y)}\right) = \beta_0 + \beta_1 x_{ij} + \beta_2 z_{ij} + u_{0j} \tag{7.1}$$

> Correlated and Uncorrelated Random Effects in Logistic Regression
>
> The reader is referred to the discussion of correlated and uncorrelated random effects in Section 5.2

7.2.1 Stata

In Stata `mixed`, the syntax for a multilevel model of the form described in Equation 7.1 is:

```
melogit y x z || group:
```

7.2.2 R

In R `lme4`, the syntax for a multilevel model of the form described in Equation 7.1 is:

```
library(lme4)

glmer(y ~ x + z + (1 | group), data = ...)
```

7.3 Run Models

> Less Variation In Logistic Than Linear Models
>
> Note that in *logistic* regression models, there is less variation to work with–due to the fact that the outcome is `1/0`, than there is in *linear* models. Therefore, in the models below, I do not attempt to estimate a random slope in addition to a random intercept, as I do in Section 5.

7.3.1 Stata

7.3.1.1 Get The Data

```
use simulated_multilevel_data.dta

generate outcome_category = outcome > 52 // dichotomous outcome
```

7.3.1.2 Run The Model

As suggested in Equation 7.1, odds ratios are obtained by exponentiating the β coefficients: e^{β}. Stata provides the odds ratios automatically with option `, or`.

```
melogit outcome_category warmth physical_punishment i.identity i.intervention HDI || ///
country:, or
```

```
Fitting fixed-effects model:

Iteration 0:  Log likelihood = -1965.6466
Iteration 1:  Log likelihood = -1963.7805
```

```
Iteration 2:  Log likelihood = -1963.7791
Iteration 3:  Log likelihood = -1963.7791

Refining starting values:

Grid node 0:  Log likelihood = -1908.9697

Fitting full model:

Iteration 0:  Log likelihood = -1908.9697  (not concave)
Iteration 1:  Log likelihood =  -1903.703
Iteration 2:  Log likelihood = -1902.2851
Iteration 3:  Log likelihood = -1901.3176
Iteration 4:  Log likelihood = -1901.2662
Iteration 5:  Log likelihood = -1901.2661

Mixed-effects logistic regression               Number of obs     =      3,000
Group variable: country                         Number of groups  =         30

                                                Obs per group:
                                                              min =        100
                                                              avg =      100.0
                                                              max =        100

Integration method: mvaghermite                 Integration pts.  =          7

                                                Wald chi2(5)      =     219.75
Log likelihood = -1901.2661                     Prob > chi2       =     0.0000
------------------------------------------------------------------------------------
   outcome_category | Odds ratio   Std. err.      z    P>|z|     [95% conf. interval]
--------------------+---------------------------------------------------------------
             warmth |   1.292603   .0278565    11.91   0.000     1.239142     1.34837
physical_punishment |   .7524276   .0222773    -9.61   0.000     .7100077     .797382
         1.identity |   .9517262   .0748541    -0.63   0.529     .8157636     1.11035
     1.intervention |   1.191581   .0940459     2.22   0.026     1.020803    1.390929
                HDI |   .9990491   .0061371    -0.15   0.877     .9870928     1.01115
              _cons |   .9115548   .3901774    -0.22   0.829     .3939478    2.109244
--------------------+---------------------------------------------------------------
country             |
         var(_cons)|   .2897697   .0880892                      .1596945    .5257944
------------------------------------------------------------------------------------
Note: Estimates are transformed only in the first equation to odds ratios.
Note: _cons estimates baseline odds (conditional on zero random effects).
```

```
LR test vs. logistic model: chibar2(01) = 125.03      Prob >= chibar2 = 0.0000
```

7.3.2 R

7.3.2.1 Get The Data

```r
library(haven)

df <- read_dta("simulated_multilevel_data.dta")

df$outcome_category <- 0 # initialize to 0

df$outcome_category[df$outcome > 52] <- 1 # dichotomous outcome
```

7.3.2.2 Change Some Variables To Categorical

```r
df$identity <- factor(df$identity)

df$intervention <- factor(df$intervention)
```

7.3.2.3 Run The Model

> Caution
>
> `lme4` does not directly provide p values in results, because of some disagreement over exactly how these p values should be calculated. Therefore, in this Appendix, I also call library `lmerTest` to provide p values for `lme4` results.

> Tip
>
> R prefers to use scientific notation when possible. I find that the use of scientific notation can be confusing in reading results. I turn off scientific notation by setting a penalty for its use: `options(scipen = 999)`.

```r
library(lme4)

library(lmerTest)
```

```r
options(scipen = 999)

fit3 <- glmer(outcome_category ~ warmth + physical_punishment +
                identity + intervention + HDI +
                (1 | country),
              family = binomial(link = "logit"),
              data = df)

summary(fit3)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: outcome_category ~ warmth + physical_punishment + identity +
    intervention + HDI + (1 | country)
   Data: df

     AIC      BIC   logLik deviance df.resid
  3816.6   3858.7  -1901.3   3802.6     2993

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.0109 -0.8798  0.4369  0.8428  2.8223

Random effects:
 Groups  Name        Variance Std.Dev.
 country (Intercept) 0.2894   0.5379
Number of obs: 3000, groups:  country, 30

Fixed effects:
                      Estimate Std. Error z value            Pr(>|z|)
(Intercept)         -0.0926371  0.4277643  -0.217              0.8286
warmth               0.2566693  0.0215443  11.914 <0.0000000000000002 ***
physical_punishment -0.2844595  0.0295990  -9.610 <0.0000000000000002 ***
identity1           -0.0494765  0.0786286  -0.629              0.5292
intervention1        0.1752879  0.0789030   2.222              0.0263 *
HDI                 -0.0009513  0.0061388  -0.155              0.8769
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
```

```
             (Intr) warmth physc_ idntt1 intrv1
warmth      -0.158
physcl_pnsh -0.170 -0.082
identity1   -0.086 -0.014  0.002
interventn1 -0.102  0.055  0.006 -0.020
HDI         -0.930 -0.007  0.012 -0.001  0.004
```

7.3.2.4 Calculate Odds Ratios

R requires one to use a bit of extra syntax to extract the odds ratios. As suggested in Equation 7.1, odds ratios are obtained by exponentiating the β coefficients: e^{β}.

```
exp(fixef(fit3))
```

```
        (Intercept)              warmth physical_punishment           identity1
          0.9115242           1.2926176           0.7524208           0.9517275
      intervention1                 HDI
          1.1915893           0.9990492
```

8 Reshaping Data

8.1 Introduction

Cross-sectional analyses (Section 5) make use of data in *wide* format: every row is a person, or family, and every person, or family, has a single row of data.

Longitudinal analyses (Section 6) make use of *long* data: every row is a person-timepoint, or family-timepoint, and every person, or family, has multiple rows of data.

Data therefore sometimes need to be *reshaped*, most often from *wide* format to *long* format. Almost any software that is capable of estimating multilevel models is capable of reshaping data.

Below, I detail the procedure for reshaping data in Stata and R.

8.2 Data in Wide Format

> **Note**
>
> The data below are in *wide* format.
> Every individual in the data set has a *single row of data*. Every row in the data set is an *individual* or *family*.

Table 8.1: Data in Wide Format

Table 8.1: Table continues below

id	physical_punishment1	warmth1	outcome1	physical_punishment2
1.1	3	3	57.47	3
1.10	2	0	62.9	3
1.100	2	5	62.71	1
1.11	4	4	55.61	2
1.12	5	4	41.15	5
1.13	4	5	63.66	3

Table 8.2: Data in Wide Format

Table 8.2: Table continues below

warmth2	outcome2	physical_punishment3	warmth3	outcome3	country	HDI
4	55.06	1	2	58.77	1	69
0	56.67	2	0	68.22	1	69
4	51.58	2	5	55.51	1	69
5	50.9	3	3	48.02	1	69
5	45.4	3	4	55.86	1	69
3	64.81	3	3	58.3	1	69

Table 8.3: Data in Wide Format

family	identity	intervention
1	1	0
10	1	0
100	1	1
11	1	1
12	0	0
13	0	1

8.3 Data Management

Because reshaping your data dramatically changes the structure of your data…

1. It is a good idea to have your raw data saved in a location where it will not be changed, and can be retrieved again if the reshape command does not work correctly, or if you simply want to modify your reshaping data workflow.(CF Section 2.3)
2. Usually we want to work with only a *subset* of your data, to keep only the data in which we are interested.

   - In Stata, the command to keep only variables of interest would be: `keep y x z t`.
   - In R, one option would be to use the subset function: `mysubset <- subset(mydata, select = c(y, x, z, t))`

8.4 Reshaping Data From Wide To Long

Usually, we are most interested in reshaping data from *wide* to *long*.

8.4.1 Stata

In Stata, I only list variables that vary over time, or are *time varying*. Stata assumes that variables that are *not listed* do *not vary over time*, or are *time invariant*.

Notice also that our *time varying* data are in the *stub-time* format, e.g. `warmth1`, `warmth2`, `physical_punishment1` `physical_punishment2`, etc. Because the variables are named in this way, Stata knows to use the *stub* (e.g. `warmth`) as the variable name, and the numeric value, (e.g. 1, 2, 3) as the timepoint.

> The `id` variable, whatever it is named, has to uniquely identify the observations. A useful Stata command here is `isid`, e.g. `isid id`. If your `id` variable is not unique, it is often due to missing values. `drop if id == .` usually solves the problem (assuming that your `id` variable is indeed named `id`, and not something else).

```
use simulated_multilevel_longitudinal_data_WIDE.dta, clear

describe

reshape long outcome physical_punishment warmth, i(id) j(wave)
```

```
Contains data from simulated_multilevel_longitudinal_data_WIDE.dta
 Observations:         3,000
    Variables:            15                  3 Jul 2024 14:29
------------------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
------------------------------------------------------------------------------------------
id              str7    %9s                   unique country family id
physical_puni~1 float   %9.0g                 1 physical_punishment
warmth1         float   %9.0g                 1 warmth
outcome1        float   %9.0g                 1 outcome
physical_puni~2 float   %9.0g                 2 physical_punishment
warmth2         float   %9.0g                 2 warmth
outcome2        float   %9.0g                 2 outcome
physical_puni~3 float   %9.0g                 3 physical_punishment
warmth3         float   %9.0g                 3 warmth
outcome3        float   %9.0g                 3 outcome
country         float   %9.0g                 country id
HDI             float   %9.0g                 Human Development Index
family          float   %9.0g                 family id
identity        float   %9.0g                 hypothetical identity group variable
```

```
intervention    float   %9.0g                 recieved intervention
-------------------------------------------------------------------------------------------
Sorted by: id

(j = 1 2 3)

Data                               Wide   ->   Long
-----------------------------------------------------------------------------
Number of observations            3,000   ->   9,000
Number of variables                  15   ->   10
j variable (3 values)                     ->   wave
xij variables:
             outcome1 outcome2 outcome3   ->   outcome
physical_punishment1 physical_punishment2 physical_punishment3->physical_punishment
                warmth1 warmth2 warmth3   ->   warmth
-----------------------------------------------------------------------------
```

8.4.2 R

In R, I only list variables that vary over time, or are *time varying*.

Notice also that our *time varying* data are in the *stub-time* format, e.g. `warmth1`, `warmth2`, `physical_punishment1` `physical_punishment2`, etc. In order to facilitate reshaping the data, it is helpful in R to insert an underscore (_) to separate the *stub* from the *time* variable.

```
library(dplyr) # data wrangling

library(tidyr) # tidy (reshape data)
```

```
# rename variables with "_" separator

df <- simulated_multilevel_longitudinal_data_WIDE %>%
  rename(outcome_1 = outcome1,
         outcome_2 = outcome2,
         outcome_3 = outcome3,
         physical_punishment_1 = physical_punishment1,
         physical_punishment_2 = physical_punishment2,
         physical_punishment_3 = physical_punishment3,
         warmth_1 = warmth1,
         warmth_2 = warmth2,
         warmth_3 = warmth3)
```

```
# pivot_longer() to long data

dfL <- df %>%
  pivot_longer(cols = c(outcome_1,
                        outcome_2,
                        outcome_3,
                        physical_punishment_1,
                        physical_punishment_2,
                        physical_punishment_3,
                        warmth_1,
                        warmth_2,
                        warmth_3),
               names_pattern = "(.+)_(.+)",
               names_to = c(".value", "t"))
```

8.5 Data in Long Format

Note

The data below are in *long* format.
Every individual/family in the data set has a *multiple rows of data*. Every row in the data set is an *individual-timepoint* or *family-timepoint*.

Table 8.4: Data in Long Format

Table 8.4: Table continues below

country	HDI	family	id	identity	intervention	t
1	69	1	1.1	1	0	1
1	69	1	1.1	1	0	2
1	69	1	1.1	1	0	3
1	69	2	1.2	1	1	1
1	69	2	1.2	1	1	2
1	69	2	1.2	1	1	3

Table 8.5: Data in Long Format

physical_punishment	warmth	outcome
3	3	57.47
3	4	55.06
1	2	58.77
2	1	50.1
3	0	53.31
3	1	49.79

9 Aggregating Data

In many instances, we may wish to aggregate data. For example, we may wish to create *contextual variables* representing the average level of an indicator across a group. In the examples I am using in this book, the group under consideration is the country. Aggregating data is also an important part of discussions of *within* and *between* variation, and is an important part of the correlated random effects model.

In the examples below, I create a group level variable for `warmth`, representing the average level of parental warmth in each country. If warmth is denoted by warmth_{ij} then the country level variable is denoted by $\overline{warmth}_{.j}$.

Below, I detail the procedure for aggregating data in Stata and R.

9.0.1 Stata

9.0.1.1 Get The Data

```
use simulated_multilevel_data.dta
```

9.0.1.2 Create A Group Level Variable

```
bysort country: egen mean_warmth = mean(warmth)
```

9.0.2 R

9.0.2.1 Get The Data

```
library(haven)

df <- read_dta("simulated_multilevel_data.dta")
```

9.0.2.2 Create A Group Level Variable

```
library(dplyr)

df <- df %>%
  group_by(country) %>%
  mutate(mean_warmth = mean(warmth))
```

References

Allaire, J. J., Teague, C., Scheidegger, C., Xie, Y., & Dervieux, C. (2024). *Quarto* (Version 1.4). https://doi.org/10.5281/zenodo.5960048

Bates, D. (2024). *MixedModels.jl Documentation*. https://juliastats.org/MixedModels.jl/stable/

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Hemken, D. (2023). *Statamarkdown: 'Stata' markdown*. https://CRAN.R-project.org/package=Statamarkdown

Li, C. (2019). JuliaCall: An R package for seamless integration between R and Julia. *The Journal of Open Source Software*, *4*(35), 1284. https://doi.org/10.21105/joss.01284

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Schanen, J. (2021). *Math person (Strogatz Prize entry)*. National Museum of Mathematics.

StataCorp. (2023). *Stata 18 mixed effects reference manual*. Stata Press.

Thoreau, H. D. (1975). The commercial spirit of modern times [1837]. In J. J. Moldenhauer, E. Moser, & A. C. Kern (Eds.), *Early essays and miscellanies*. Princeton University Press.