

Multilevel Multilingual

Multilevel Models in Stata, R and Julia

Andrew Grogan-Kaylor

2024-04-07

Table of contents

1	Multilevel Multilingual	5
1.1	Introduction	5
1.2	The Data	6
1.3	An Introduction To Equations and Syntax	7
1.3.1	Stata	7
1.3.2	R	7
1.3.3	Julia	7
2	Descriptive Statistics	8
2.1	Descriptive Statistics	8
2.1.1	Stata	8
2.1.2	R	8
2.1.3	Julia	9
3	Unconditional Model	11
3.1	The Equation	11
3.2	Run Models	11
3.2.1	Stata	11
3.2.2	R	12
3.2.3	Julia	14
4	Cross Sectional Multilevel Models	15
4.1	The Equation	15
4.2	Correlated and Uncorrelated Random Effects	15
4.3	Run Models	16
4.3.1	Stata	16
4.3.2	R	17
4.3.3	Julia	19
5	Longitudinal Multilevel Models	21
5.1	The Data	21
5.2	The Equation	21
5.3	Run Models	21
5.3.1	Stata	21
5.3.2	R	24
5.3.3	Julia	27

List of Tables

1.1	Software for Multilevel Modeling	5
1.2	Sample of Simulated Multilevel Data	6
4.1	Correlated and Uncorrelated Random Effects	16

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, 2021), [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. However, there are substantial differences between the different packages: Stata is proprietary *for cost* software, which is very well documented and very intuitive. R is free open source software which is less intuitive, but there are many excellent resources for learning R. Julia is newer open source software, 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.

Table 1.1: Software for Multilevel Modeling

Software	Cost	Ease of Use
Stata	some cost	learning curve, but intuitive for both multilevel modeling and graphing.
R	free	learning curve: intuitive for multilevel modeling; but steeper learning curve for graphing.
Julia	free	learning curve: steeper 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 multi-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 `()`.

1.2 The Data

The examples use the `simulated_multilevel_data.dta` file from *Multilevel Thinking*. Here is a [direct link](#) to download the data.

Table 1.2: Sample of Simulated Multilevel Data

country	HDI	family	id	group	physical_punishment	warmth	outcome
1	69	1	1.1	2	2	3	59.18
1	69	2	1.2	2	4	0	61.54
1	69	3	1.3	1	4	4	51.87
1	69	4	1.4	2	0	6	51.71
1	69	5	1.5	2	3	2	55.88
1	69	6	1.6	1	5	3	60.78

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 \mathbf{x} and \mathbf{z} , clustering variable `group`, and a random slope for \mathbf{x} . i is the index for the person, while j is the index for the `group`.

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

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 || group: x
```

1.3.2 R

In R `lme4`, the general 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 general 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 Descriptive Statistics

2.1 Descriptive Statistics

2.1.1 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 group
```

Variable	Obs	Mean	Std. dev.	Min	Max
outcome	3,000	53.46757	6.65179	33.39014	76.75101
warmth	3,000	3.524333	1.889956	0	7
physical_p~t	3,000	2.494667	1.380075	0	5
HDI	3,000	64.76667	17.24562	33	87

arbitrary			
group			
variable	Freq.	Percent	Cum.
1	1,507	50.23	50.23
2	1,493	49.77	100.00
Total	3,000	100.00	

2.1.2 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.

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

df$group <- factor(df$group)

summary(df)
```

	country	HDI	family	id	group
1	: 100	Min. :33.00	Min. : 1.00	Length:3000	1:1507
2	: 100	1st Qu.:53.00	1st Qu.: 25.75	Class :character	2:1493
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				
	physical_punishment	warmth	outcome		
	Min. :0.000	Min. :0.000	Min. :33.39		
	1st Qu.:2.000	1st Qu.:2.000	1st Qu.:48.78		
	Median :3.000	Median :4.000	Median :53.64		
	Mean :2.495	Mean :3.524	Mean :53.47		
	3rd Qu.:3.250	3rd Qu.:5.000	3rd Qu.:58.06		
	Max. :5.000	Max. :7.000	Max. :76.75		

2.1.3 Julia

```
using Tables, MixedModels, MixedModelsExtras, StatFiles, DataFrames, CategoricalArrays, DataAPI

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

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

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

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

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

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

8×7 DataFrame

Row	variable	mean	min	median	max	nmissing	eltyp
	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	group		1.0		2.0	0	Union
6	physical_punishment	2.49467	0.0	3.0	5.0	0	Union
7	warmth	3.52433	0.0	4.0	7.0	0	Union
8	outcome	53.4676	33.3901	53.6426	76.751	0	Union

1 column omitted

3 Unconditional 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.

3.1 The Equation

$$\text{outcome}_{ij} = \beta_0 + u_{0j} + e_{ij} \quad (3.1)$$

The Intraclass Correlation Coefficient (ICC) is given by:

$$\text{ICC} = \frac{\text{var}(u_{0j})}{\text{var}(u_{0j}) + \text{var}(e_{ij})} \quad (3.2)$$

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

3.2 Run Models

3.2.1 Stata

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

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

Performing EM optimization ...

Performing gradient-based optimization:

Iteration 0: Log likelihood = -9856.1548

Iteration 1: Log likelihood = -9856.1548

Computing standard errors ...

Mixed-effects ML regression
Group variable: country

Number of obs = 3,000
Number of groups = 30
Obs per group:
min = 100
avg = 100.0
max = 100
Wald chi2(0) = .
Prob > chi2 = .

Log likelihood = -9856.1548

outcome	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_cons	53.46757	.3539097	151.08	0.000	52.77392	54.16122

Random-effects parameters	Estimate	Std. err.	[95% conf. interval]	
country: Identity				
var(_cons)	3.348734	.9702594	1.897816	5.908906
var(Residual)	40.88284	1.060908	38.8555	43.01597

LR test vs. linear model: chibar2(01) = 169.64 Prob >= chibar2 = 0.0000

```
estat icc // ICC
```

Intraclass correlation

Level	ICC	Std. err.	[95% conf. interval]	
country	.0757091	.0203761	.0442419	.1265931

3.2.2 R

```
library(haven)

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

```
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: 19712.5

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.97650	-0.68006	0.00936	0.67580	3.03510

Random effects:

Groups	Name	Variance	Std.Dev.
country	(Intercept)	3.478	1.865
	Residual	40.883	6.394

Number of obs: 3000, groups: country, 30

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	53.47	0.36	148.5

```
library(performance)

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

Intraclass Correlation Coefficient

Adjusted ICC: 0.078
Unadjusted ICC: 0.078

3.2.3 Julia

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

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

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

```
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
	-9856.1548	19712.3097	19718.3097	19718.3177	19736.3288

Variance components:

	Column	Variance	Std.Dev.
country	(Intercept)	3.34871	1.82995
Residual		40.88285	6.39397

Number of obs: 3000; levels of grouping factors: 30

Fixed-effects parameters:

	Coef.	Std. Error	z	Pr(> z)
(Intercept)	53.4676	0.353908	151.08	<1e-99

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

```
0.07570852291396266
```

4 Cross Sectional Multilevel Models

4.1 The Equation

Recall the general model of Equation 1.1, and the syntax outlined in Section 1.3. Below in Equation 4.1, we consider a more substantive example.

$$\text{outcome}_{ij} = \beta_0 + \beta_1 \text{warmth}_{ij} + \quad (4.1)$$

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

$$\beta_3 \text{group}_{ij} + \beta_4 \text{HDI}_{ij} +$$

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

4.2 Correlated and Uncorrelated Random Effects

Consider the covariance matrix of random effects (e.g. u_{0j} and u_{1j}). In Equation 4.2 the covariances of the random effects are constrained to be zero.

$$\begin{bmatrix} \text{var}(u_{0j}) & 0 \\ 0 & \text{var}(u_{1j}) \end{bmatrix} \quad (4.2)$$

As discussed in Chapter 6, however, one can consider a multilevel model in which the random effects are correlated, as is the case in Equation 4.3.

$$\begin{bmatrix} \text{var}(u_{0j}) & \text{cov}(u_{0j}, u_{1j}) \\ \text{cov}(u_{0j}, u_{1j}) & \text{var}(u_{1j}) \end{bmatrix} \quad (4.3)$$

Procedures for estimating models with uncorrelated and correlated random effects are detailed below (Bates et al., 2015; Bates, 2024; StataCorp, 2021).

Table 4.1: Correlated and Uncorrelated Random Effects

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

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

4.3 Run Models

4.3.1 Stata

4.3.1.1 Get The Data

```
use simulated_multilevel_data.dta
```

4.3.1.2 Run The Model

```
mixed outcome warmth physical_punishment group HDI || country: warmth
```

Performing EM optimization ...

Performing gradient-based optimization:

```
Iteration 0: Log likelihood = -9668.198
Iteration 1: Log likelihood = -9667.9551
Iteration 2: Log likelihood = -9667.9534
Iteration 3: Log likelihood = -9667.9533
Iteration 4: Log likelihood = -9667.9532
```


Computing standard errors ...

Mixed-effects ML regression
Group variable: country

Number of obs = 3,000
Number of groups = 30
Obs per group:
min = 100
avg = 100.0
max = 100
Wald chi2(4) = 401.26
Prob > chi2 = 0.0000

Log likelihood = -9667.9532

outcome	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
warmth	.9616447	.0581825	16.53	0.000	.8476091	1.07568
physical_punishment	-.8453802	.0798155	-10.59	0.000	-1.001816	-.6889448
group	1.084344	.2200539	4.93	0.000	.6530461	1.515642
HDI	.010557	.0204522	0.52	0.606	-.0295286	.0506426
_cons	49.87963	1.436612	34.72	0.000	47.06392	52.69534

Random-effects parameters	Estimate	Std. err.	[95% conf. interval]	
country: Independent				
var(warmth)	1.83e-06	.0000173	1.76e-14	190.9774
var(_cons)	3.370262	.9633726	1.924651	5.901676
var(Residual)	36.01906	.9346936	34.23291	37.89842

LR test vs. linear model: chi2(2) = 198.01

Prob > chi2 = 0.0000

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

4.3.2 R

4.3.2.1 Get The Data

```
library(haven)
```

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

4.3.2.2 Run The Model

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

summary(fit1)
```

Linear mixed model fit by REML ['lmerMod']

Formula: outcome ~ warmth + physical_punishment + group + HDI + ((1 |
country) + (0 + warmth | country))

Data: df

REML criterion at convergence: 19350.3

Scaled residuals:

Min	1Q	Median	3Q	Max
-3.4496	-0.6807	0.0016	0.6864	3.1792

Random effects:

Groups	Name	Variance	Std.Dev.
country	(Intercept)	3.611568	1.90041
country.1	warmth	0.001876	0.04331
Residual		36.049124	6.00409

Number of obs: 3000, groups: country, 30

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	49.88754	1.48203	33.662
warmth	0.96155	0.05875	16.367
physical_punishment	-0.84556	0.07986	-10.588
group	1.08471	0.22017	4.927
HDI	0.01044	0.02116	0.493

Correlation of Fixed Effects:

	(Intr) warmth physc_ group
warmth	-0.126

```

physcl_pnsh -0.135 -0.025
group       -0.218 -0.010 -0.019
HDI         -0.925 -0.006  0.008 -0.001

```

4.3.3 Julia

4.3.3.1 Get The Data

```

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

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

```

4.3.3.2 Change Country To Categorical

```

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

```

4.3.3.3 Run The Model

```

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

```

Linear mixed model fit by maximum likelihood

```

outcome ~ 1 + warmth + physical_punishment + group + HDI + (1 | country) + (0 + warmth | co
logLik    -2 logLik      AIC      AICc      BIC
-9667.9532 19335.9065 19351.9065 19351.9546 19399.9574

```

Variance components:

```

          Column  Variance Std.Dev.  Corr.
country (Intercept)   3.36982 1.83571
      warmth           0.00000 0.00000  .

```

```

Residual           36.01912 6.00159

```

Number of obs: 3000; levels of grouping factors: 30

Fixed-effects parameters:

	Coef.	Std. Error	z	Pr(> z)
(Intercept)	49.8796	1.43653	34.72	<1e-99
warmth	0.961645	0.058182	16.53	<1e-60
physical_punishment	-0.84538	0.0798155	-10.59	<1e-25
group	1.08434	0.220054	4.93	<1e-06
HDI	0.0105571	0.0204509	0.52	0.6057

5 Longitudinal Multilevel Models

5.1 The Data

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

5.2 The Equation

$$\text{outcome}_{itj} = \beta_0 + \beta_1 \text{parental warmth}_{itj} + \beta_2 \text{physical punishment}_{itj} + \beta_3 \text{time}_{itj} + \quad (5.1)$$

$$\beta_4 \text{group}_{itj} + \beta_5 \text{HDI}_{itj} +$$

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

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

5.3 Run Models

5.3.1 Stata

5.3.1.1 Get The Data

```
use simulated_multilevel_longitudinal_data.dta
```

5.3.1.2 Run The Model

5.3.1.2.1 Main Effects Only

```
mixed outcome t warmth physical_punishment group HDI || country: warmth
```

Performing EM optimization ...

Performing gradient-based optimization:

Iteration 0: Log likelihood = -28795.37

Iteration 1: Log likelihood = -28795.232

Iteration 2: Log likelihood = -28795.232

Computing standard errors ...

Mixed-effects ML regression

Group variable: country

Number of obs = 9,000

Number of groups = 30

Obs per group:

min = 300

avg = 300.0

max = 300

Wald chi2(5) = 1366.93

Prob > chi2 = 0.0000

Log likelihood = -28795.232

outcome	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
t	.9882371	.0761439	12.98	0.000	.8389979	1.137476
warmth	.9427117	.0342282	27.54	0.000	.8756256	1.009798
physical_punishment	-.9020727	.0452759	-19.92	0.000	-.9908119	-.8133336
group	.9861238	.1249047	7.90	0.000	.7413151	1.230933
HDI	.0073726	.020661	0.36	0.721	-.0331222	.0478674
_cons	49.45537	1.414072	34.97	0.000	46.68384	52.2269

Random-effects parameters	Estimate	Std. err.	[95% conf. interval]	
country: Independent				
var(warmth)	.0024684	.0082517	3.52e-06	1.72956
var(_cons)	3.663663	.9914845	2.155548	6.22692

```

var(Residual) | 34.78483 .5200702 33.7803 35.81923
-----
LR test vs. linear model: chi2(2) = 805.75 Prob > chi2 = 0.0000

```

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

5.3.1.2.2 Interactions With Time

```
mixed outcome c.t##(c.warmth c.physical_punishment i.group c.HDI) || country: warmth
```

Performing EM optimization ...

Performing gradient-based optimization:

Iteration 0: Log likelihood = -28794.99

Iteration 1: Log likelihood = -28794.855

Iteration 2: Log likelihood = -28794.855

Computing standard errors ...

Mixed-effects ML regression

Group variable: country

Number of obs = 9,000

Number of groups = 30

Obs per group:

min = 300

avg = 300.0

max = 300

Wald chi2(9) = 1365.73

Prob > chi2 = 0.0000

Log likelihood = -28794.855

```

-----
outcome | Coefficient Std. err. z P>|z| [95% conf. interval]
-----+-----
t | 1.047448 .3619795 2.89 0.004 .3379816 1.756915
warmth | .8869901 .0876058 10.12 0.000 .715286 1.058694
physical_punishment | -.893285 .1194705 -7.48 0.000 -1.127443 -.659127
2.group | .9648545 .3292217 2.93 0.003 .3195918 1.610117
HDI | .0120622 .022474 0.54 0.591 -.0319861 .0561104
|
c.t#c.warmth | .0277903 .0402665 0.69 0.490 -.0511306 .1067112
|
c.t#|
c.physical_punishment | -.0041479 .0553051 -0.08 0.940 -.1125439 .1042482

```

group#c.t							
2		.0105177	.1523009	0.07	0.945	-.2879865	.3090219
c.t#c.HDI		-.002342	.0044172	-0.53	0.596	-.0109996	.0063155
_cons		50.32233	1.572089	32.01	0.000	47.2411	53.40357

Random-effects parameters		Estimate	Std. err.	[95% conf. interval]	
<hr/>					
country: Independent					
var(warmth)		.0025661	.0083259	4.44e-06	1.482773
var(_cons)		3.66269	.991533	2.154617	6.226305
<hr/>					
var(Residual)		34.78158	.5200283	33.77713	35.8159

LR test vs. linear model: $\chi^2(2) = 805.90$ Prob > $\chi^2 = 0.0000$

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

5.3.2 R

5.3.2.1 Get The Data

```
library(haven)

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

5.3.2.2 Run The Model

5.3.2.2.1 Main Effects Only

```
fit2A <- lmer(outcome ~ t + warmth + physical_punishment +
              group + HDI +
              (1 | country/id),
              data = dfL)

summary(fit2A)
```



```
Linear mixed model fit by REML ['lmerMod']
Formula: outcome ~ t + warmth + physical_punishment + group + HDI + (1 |
  country/id)
Data: dfL
```

REML criterion at convergence: 57088.4

Scaled residuals:

Min	1Q	Median	3Q	Max
-3.4471	-0.6226	0.0081	0.6153	3.1993

Random effects:

Groups	Name	Variance	Std.Dev.
id:country	(Intercept)	8.864	2.977
country	(Intercept)	3.924	1.981
Residual		26.008	5.100

Number of obs: 9000, groups: id:country, 3000; country, 30

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	49.494782	1.471780	33.629
t	0.987964	0.065840	15.005
warmth	0.946259	0.038200	24.771
physical_punishment	-0.926880	0.049970	-18.549
group	0.985786	0.153550	6.420
HDI	0.007543	0.021437	0.352

Correlation of Fixed Effects:

	(Intr)	t	warmth	physc_	group
t	-0.090				
warmth	-0.085	0.008			
physcl_pnsh	-0.085	0.003	-0.019		
group	-0.154	0.000	-0.013	-0.008	
HDI	-0.943	0.000	-0.003	0.003	0.000

5.3.2.2.2 Interactions With Time

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

```
summary(fit2B)
```

Linear mixed model fit by REML ['lmerMod']

Formula: outcome ~ t * (warmth + physical_punishment + group + HDI) +
(1 | country/id)

Data: dfL

REML criterion at convergence: 57107.3

Scaled residuals:

Min	1Q	Median	3Q	Max
-3.4431	-0.6248	0.0071	0.6183	3.1961

Random effects:

Groups	Name	Variance	Std.Dev.
id:country	(Intercept)	8.868	2.978
country	(Intercept)	3.925	1.981
Residual		26.014	5.100

Number of obs: 9000, groups: id:country, 3000; country, 30

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	49.453036	1.637740	30.196
t	1.008199	0.364915	2.763
warmth	0.865659	0.080487	10.755
physical_punishment	-0.908148	0.110449	-8.222
group	0.966988	0.304936	3.171
HDI	0.012277	0.022761	0.539
t:warmth	0.040170	0.035364	1.136
t:physical_punishment	-0.008932	0.049262	-0.181
t:group	0.009180	0.131714	0.070
t:HDI	-0.002359	0.003820	-0.618

Correlation of Fixed Effects:

	(Intr)	t	warmth	physc_	group	HDI	t:wrmt	t:phy_	t:grop
t		-0.446							
warmth		-0.159	0.278						
physcl_pnsh		-0.169	0.302	-0.022					
group		-0.274	0.459	-0.010	-0.014				
HDI		-0.900	0.227	-0.008	0.009	-0.001			
t:warmth		0.141	-0.316	-0.880	0.017	0.010	0.007		

```
t:physcl_pn  0.150 -0.338  0.017 -0.892  0.010 -0.007 -0.015
t:group      0.237 -0.532  0.009  0.012 -0.864  0.001 -0.012 -0.008
t:HDI        0.302 -0.676  0.018 -0.020  0.002 -0.336 -0.018  0.016 -0.002
```

5.3.3 Julia

5.3.3.1 Get The Data

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

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

5.3.3.2 Run The Model

5.3.3.2.1 Change Country To Categorical

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

5.3.3.2.2 Main Effects Only

```
m2A = fit(MixedModel, @formula(outcome ~ t + warmth +
                                physical_punishment +
                                group + HDI +
                                (1 | country) +
                                (0 + warmth | country) +
                                (1 | id)), dfL)
```

Linear mixed model fit by maximum likelihood

outcome ~ 1 + t + warmth + physical_punishment + group + HDI + (1 | country) + (0 + warmth

	logLik	-2 logLik	AIC	AICc	BIC
	-28533.9968	57067.9935	57087.9935	57088.0180	57159.0433

Variance components:

	Column	Variance	Std.Dev.	Corr.
id	(Intercept)	8.85263	2.97534	
country	(Intercept)	3.65031	1.91058	
	warmth	0.00000	0.00000	.
Residual		26.00093	5.09911	

Number of obs: 9000; levels of grouping factors: 3000, 30

Fixed-effects parameters:

	Coef.	Std. Error	z	Pr(> z)
(Intercept)	49.4945	1.42422	34.75	<1e-99
t	0.987965	0.0658315	15.01	<1e-50
warmth	0.946255	0.0381869	24.78	<1e-99
physical_punishment	-0.926774	0.0499549	-18.55	<1e-76
group	0.985819	0.153487	6.42	<1e-09
HDI	0.00754357	0.0207101	0.36	0.7157

5.3.3.2.3 Interactions With Time

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

Linear mixed model fit by maximum likelihood

outcome ~ 1 + t + warmth + physical_punishment + group + HDI + t & warmth + t & physical_punishment

logLik	-2 logLik	AIC	AICc	BIC
-28533.1579	57066.3158	57094.3158	57094.3626	57193.7855

Variance components:

	Column	Variance	Std.Dev.	Corr.
id	(Intercept)	8.86088	2.97672	
country	(Intercept)	3.65109	1.91078	
	warmth	0.00000	0.00000	.
Residual		25.99020	5.09806	

Number of obs: 9000; levels of grouping factors: 3000, 30

Fixed-effects parameters:

	Coef.	Std. Error	z	Pr(> z)
(Intercept)	49.4526	1.59495	31.01	<1e-99

t	1.00828	0.364747	2.76	0.0057
warmth	0.865674	0.0804504	10.76	<1e-26
physical_punishment	-0.908024	0.110399	-8.22	<1e-15
group	0.967016	0.304798	3.17	0.0015
HDI	0.0122774	0.0220761	0.56	0.5781
t & warmth	0.0401613	0.0353475	1.14	0.2559
t & physical_punishment	-0.00895284	0.0492392	-0.18	0.8557
t & group	0.00918249	0.131654	0.07	0.9444
t & HDI	-0.00235908	0.00381845	-0.62	0.5367

References

- 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>
- 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. (2021). *Stata 17 multilevel mixed effects reference manual*. Stata Press.
- Thoreau, H. D. (1975). The commercial spirit of modern times. In J. J. Moldenhauer, E. Moser, & A. C. Kern (Eds.), *Early essays and miscellanies*. Princeton University Press.