

Global Families Project

Global Families Project Team

6/27/23

Table of contents

1	Project Summary	6
2	Research Team	8
3	Simulated Multi-Country Data	9
3.1	Variables and Variable Labels	9
3.2	A Sample Of The Data	10
4	A Quick Introduction to R	11
4.1	Why Use R?	11
4.2	Get R	11
4.3	Get Data	11
4.3.1	Data in R Format	12
4.3.2	Data in Other Formats	12
4.4	Process and Clean Data	12
4.4.1	The \$ Sign	12
4.4.2	Recoding Data	12
4.4.3	Numeric and Factor Variables	13
4.5	Visualize Data	13
4.5.1	Histogram	13
4.5.2	Barplot	14
4.6	Analyze Data: Descriptive Statistics	15
5	A Quick Introduction To ggplot2	17
5.1	Why Use ggplot?	17
5.2	The Essential Idea Of ggplot Is Simple	17
5.3	Get Started	18
5.3.1	Call Libraries	18
5.3.2	Get Data	18
5.4	Some Examples	18
5.4.1	One Continuous Variable	18
5.4.2	One Categorical Variable	19
5.5	Add Some Options	20
6	Quantitative Data Analysis	22
6.1	Introduction	22

6.2	Some Tools for Analysis	22
6.3	Working With R	23
6.3.1	Our Data	23
6.3.2	Cleaning Data	23
6.3.3	Simple Analysis	25
References		27
Appendices		27
A	Simulating MICS Data	28
A.1	Call Relevant Libraries	28
A.2	Setup Some Basic Parameters of the Data	29
A.3	Simulate Data Based on MICS	29
A.3.1	Level 2	29
A.3.2	Level 1	30
A.3.3	Variable Labels	30
A.4	Explore The Simulated Data With A Graph	31
A.5	Explore The Simulated Data With A Logistic Regression	32
A.6	Write data to various formats	33

List of Figures

4.1	Histogram of Gender Inequality Index	14
4.2	Barplot of Aggression	15
5.1	Histogram of Gender Inequality Index	19
A.1	Graph of Simulated Data	32

List of Tables

A.1	?(caption)	31
A.2	?(caption)	34

1 Project Summary

Gender inequality perpetuates harmful norms that justify violence against women and children and is associated with higher rates of family violence.

Worldwide, parental physical abuse is a common form of family violence that children are exposed to at alarming rates. Parental engagement in physical abuse is linked to negative child outcomes including depression, anxiety, and aggression that may persist into adulthood. Globally, these continuing mental health and aggression problems may have high financial costs, with effects both on social service systems and developing economies.

Despite the substantial scholarship on parent- and family-level predictors of parent-to-child physical violence, important questions remain about societal-level predictors of parental physical abuse and its associations with young children's development in developing and transitional countries.

A further gap in prior literature is the lack of studies that have examined potential moderators such as child age and household economic status in the associations between gender inequality and parental violence against children.

Using data from over 520,000 families in 57 low- and middle-income countries (LMICs), the current project seeks to address these research gaps by examining the associations of country-level gender inequality and violent social contexts with caregivers' use of physically abusive behavior and child social-emotional development. We will employ multilevel models using data on parental physical violence against children, family socio-economic characteristics, and children's social-emotional development from the UNICEF Multiple Indicator Cluster Surveys (MICS) and data on country-level gender inequality and violent social contexts from the United Nations Development Programme on Human Development and the World Health Organization Global Health Observatory.

The specific aims are to 1) examine the associations of gender inequality with parental child physical abuse in LMICs, and the moderating roles of child age and household economic status in these associations, 2) examine the associations of violent social norms and crimes with parental physical abuse in LMICs, and 3) examine the associations of parental physical abuse with child social-emotional development in the context of gender inequality and violent norms and crimes in LMICs, and whether country-level normativeness of physical abuse moderates these associations.

The proposed studies will advance the understanding of macro-level social and economic indicators that perpetuate caregivers' physical violence against children in international contexts.

Study findings will inform cross-cultural programs and policies that reduce gender disparities and prevent parental physical abuse to promote child social-emotional development across the globe.

In addition, these studies will provide rigorous research engagement opportunities to undergraduate students and graduate students and strengthen the research environment at the University of Michigan-Flint.

2 Research Team

Julie Ma, Principal Investigator

Associate Professor of Social Work, UM-Flint

Professor Ma's research interests center around the effects of neighborhood disadvantage and negative parenting on the well-being of children. Her research builds on her experience in parent education programs that serve families in marginalized communities in Michigan. Much of her current research focuses on the risks of negative contextual and family influences such as neighborhood poverty and disorganization, and parental corporal punishment on behavior problems and maltreatment in early childhood.

Andy Grogan-Kaylor, Co-Investigator

Sandra K. Danziger Collegiate Professor, Professor of Social Work

Professor Grogan-Kaylor's research focuses on knowledge development and intervention research on children and families with the aim of reducing violence against children and improving family and child wellbeing. Grogan-Kaylor's current research projects examine parenting behaviors such as physical punishment and parental expressions of emotional warmth and support, and their effects on children's aggression, antisocial behavior, anxiety, and depression.

Shawna Lee, Co-Investigator

Professor of Social Work

Professor Lee is a professor at the University of Michigan School of Social Work. She is the director of the Parenting in Context Research Lab and the director of the Program Evaluation Group at the School. Lee has published on topics related to child maltreatment, fathers' parenting, father-child relationships, parenting stress and family functioning, and parental discipline. Her recent research focuses on parenting and stress during the COVID-19 pandemic.

3 Simulated Multi-Country Data

This website makes use of simulated data. Data come from 30 hypothetical countries. Data contain measures of a few key aspects of parenting¹ or caregiving that have proven salient in the empirical literature on parenting to date. The outcome is **aggression** against other children.

Download The Data

- [R format](#)
- [Stata Format](#)
- [SPSS](#)

```
load("./simulate-data/MICSsimulated.RData")
```

3.1 Variables and Variable Labels

```
labelled::look_for(MICSsimulated)
```

pos	variable	label	col_type	values
1	id	id	int	
2	country	country	int	
3	GII	gender inequality index	int	
4	cd1	spank	int	
5	cd2	beat	int	
6	cd3	shout	int	
7	cd4	explain	int	
8	aggression	aggression	int	

¹We use the term parenting throughout this site, but are aware that such parenting may come from biological parents, or from other caregivers.

3.2 A Sample Of The Data

A sample of the data is given below.

```
head(MICSsimulated)
```

	id	country	GII	cd1	cd2	cd3	cd4	aggression
1	1		1	20	0	1	0	1
2	2		1	20	0	0	1	0
3	3		1	20	0	0	0	1
4	4		1	20	1	1	0	1
5	5		1	20	0	0	1	1
6	6		1	20	1	0	0	1

4 A Quick Introduction to R

4.1 Why Use R?

R has a reputation for being difficult to learn, and a lot of that reputation is deserved. However, it is possible to teach R in an accessible way, and **a little bit of R can take you a long way.**

R is open source, and therefore free, statistical software that is particularly good at obtaining, analyzing and visualizing data.

R Commands are stored in a *script* or *code* file that usually ends in .R, e.g. `myscript.R`. The command file is distinct from your actual data, stored in an .RData file, e.g. `mydata.RData`.

A great deal of data analysis and visualization involves the same core set of steps.

Given the fact that we often want to apply the same core set of tasks to new questions and new data, there are ways to overcome the steep learning curve and learn a replicable set of commands that can be applied to problem after problem. **The same 5 to 10 lines of R code can often be tweaked over and over again for multiple projects.**

have a question → get data → process and clean data →
visualize data → analyze data → make conclusions

4.2 Get R

R is available at <https://www.r-project.org/>. R is a lot easier to run if you run it from RStudio, <http://www.rstudio.com>.

4.3 Get Data

Data may already be in R format, or may come from other types of data files like SPSS, Stata, or Excel. Especially in beginning R programming, getting the data into R can be the most complicated part of your program.

4.3.1 Data in R Format

```
load("./simulate-data/MICSsimulated.RData") # data in R format
```

4.3.2 Data in Other Formats

If data are in other formats, slightly different code may be required.

```
library(haven) # library for importing data
mydata <- read_sav("the/path/to/mySPSSfile.sav") # SPSS
mydata <- read_dta("the/path/to/myStatafile.dta") # Stata

library(readxl) # library for importing Excel files
mydata <- read_excel("the/path/to/mySpreadsheet.xls")

save(mydata, file = "mydata.RData") # save in R format
```

4.4 Process and Clean Data

4.4.1 The \$ Sign

The \$ sign is a kind of “connector”. `mydata$x` means: “The variable `x` in the dataset called `mydata`”.

4.4.2 Recoding Data

Data sometimes need to be recoded. For example, outliers may need to be changed to missing, or a value that is supposed to indicate missing data (e.g. -9) may need to be changed to missing.

Recoding uses the following construction:

```
data$variable[condition] <- new value
```

For example, change an outlier value: When `cd1` is 2 change it to missing (NA).

```
MICSsimulated$cd1[MICSsimulated$cd1 == 2] <- NA # outlier (2) to NA
```

Change variable `cd1` to missing (NA) when it is -9.

```
MICSsimulated$cd1[MICSsimulated$cd1 == -9] <- NA # missing (-9) to NA
```

4.4.3 Numeric and Factor Variables

R makes a strong distinction between *continuous numeric* variables that measure scales like mental health or neighborhood safety, and *categorical factor variables* that measure non-ordered categories like religious identity or gender identity.

Many statistical and graphical procedures are designed to recognize and work with different variable types. You often *don't* need to use all of the options. e.g. `mydata$w <- factor(mydata$z)` will often work just fine. **Changing variables from factor to numeric, and vice versa can sometimes be the simple solution that solves a lot of problems when you are trying to graph your variables.**

```
MICSsimulated$aggression <-  
  factor(MICSsimulated$aggression, # original numeric variable  
        levels = c(0, 1),  
        labels = c("no aggression", "aggression"),  
        ordered = TRUE) # whether order matters  
  
# MICSsimulated$z <- as.numeric(MICSsimulated$w) # factor to numeric
```

4.5 Visualize Data

4.5.1 Histogram

```
hist(MICSsimulated$GII, # what I'm graphing  
     main = "Gender Inequality Index", # title  
     xlab = "GII", # label for x axis  
     col = "blue") # color
```



Figure 4.1: Histogram of Gender Inequality Index

💡 Tip

You often *don't* need to use all of the options. e.g. `hist(mydata$x)` will work just fine.

4.5.2 Barplot

```
barplot(table(MICSsimulated$aggression), # what I'm graphing
        main = "Child Displays Aggression", # title
        xlab = "Aggression", # label for x axis
        col = "gold") # color
```

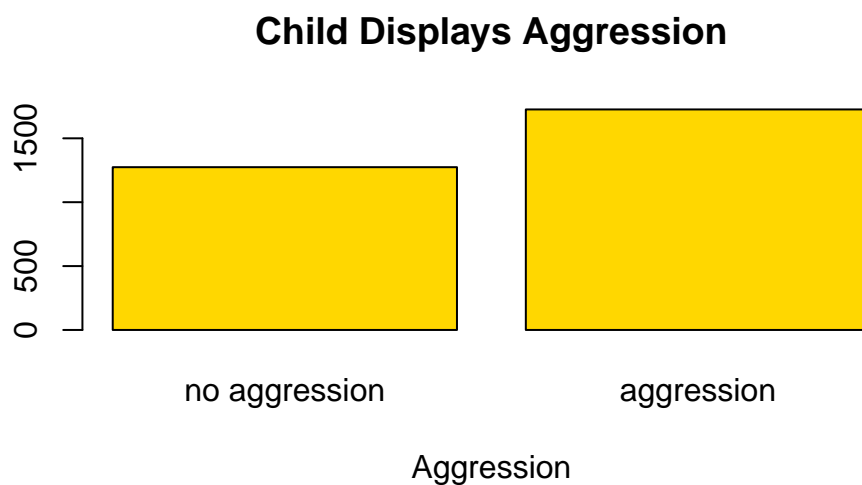


Figure 4.2: Barplot of Aggression

💡 Tip

You often *don't* need to use all of the options. e.g. `barplot(table(mydata$z))` will work just fine.

4.6 Analyze Data: Descriptive Statistics

```
summary(mydata$x) # for continuous or factor variables
```

```
table(mydata$z) # especially suitable for factor variables
```

```
summary(MICSsimulated$GII)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
15.0	22.0	24.0	24.2	27.0	31.0

```
table(MICSsimulated$aggression)
```

no aggression	aggression
1274	1726

5 A Quick Introduction To ggplot2

5.1 Why Use ggplot?¹

A great deal of data analysis and visualization involves the same core set of steps: get some data, clean it up a little, run some descriptive statistics, run some bivariate statistics, create a graph or a visualization. **ggplot2** can be an important part of a replicable, automated, documented workflow for complex projects.

have a question → get data → process and clean data →
visualize data → analyze data → make conclusions

Given the fact that we often want to apply the same core set of tasks to new questions and new data, there are ways to overcome the steep learning curve and learn a replicable set of commands that can be applied to problem after problem.

The same 5 to 10 lines of ggplot2 code can often be tweaked over and over again for multiple projects.

5.2 The Essential Idea Of ggplot Is Simple

There are 3 essential elements to any **ggplot** call:

1. A reference to the data you are using.
2. An *aesthetic* that tells **ggplot** which variables are being mapped to the *x axis*, *y axis*, (and often other attributes of the graph, such as the *color fill*). Intuitively, the aesthetic can be thought of as **what you are graphing**.
3. A *geom* or *geometry* that tells ggplot about the basic structure of the graph. Intuitively, the geom can be thought of as **how you are graphing it**.

You can also add other options, such as a *graph title*, *axis labels* and *overall theme* for the graph.

¹More information can be found here: <https://agrogan1.github.io/R/introduction-to-ggplot2/introduction-to-ggplot2.html>

5.3 Get Started

5.3.1 Call Libraries

```
library(ggplot2) # beautiful graphs  
  
library(ggthemes) # nice themes for ggplot2
```

5.3.2 Get Data

```
load("./simulate-data/MICSsimulated.RData") # data in R format
```

5.4 Some Examples²

5.4.1 One Continuous Variable

```
# anything that starts with a '#' is a comment  
  
ggplot(MICSsimulated, # the data I am using  
       aes(x = GII)) + # the variable I am using  
  geom_histogram() # how I am graphing it
```

²Changing variables from factor to numeric (e.g. `aes(x = as.numeric(outcome))`), and *vice versa* can sometimes be a simple solution that solves a lot of problems when you are trying to graph your variables.

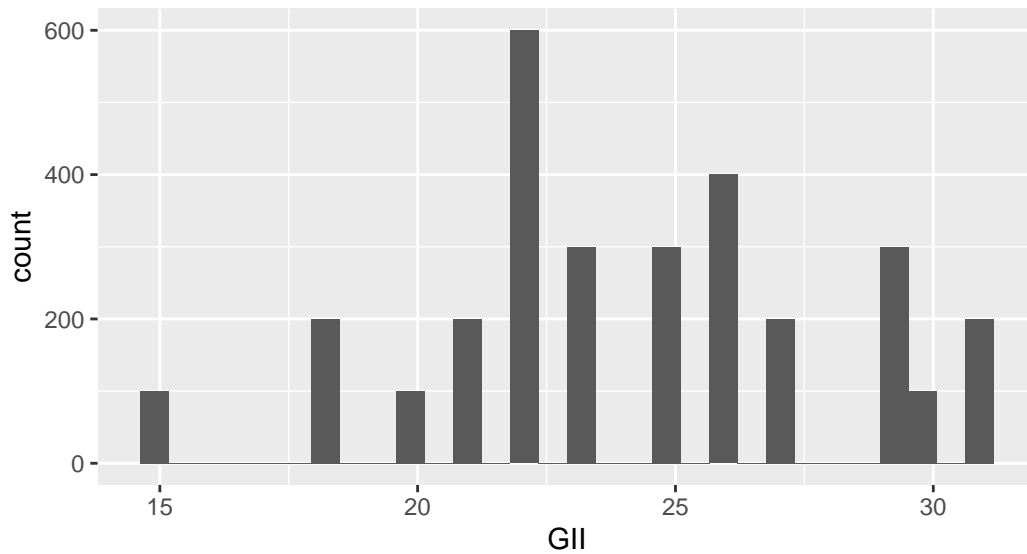


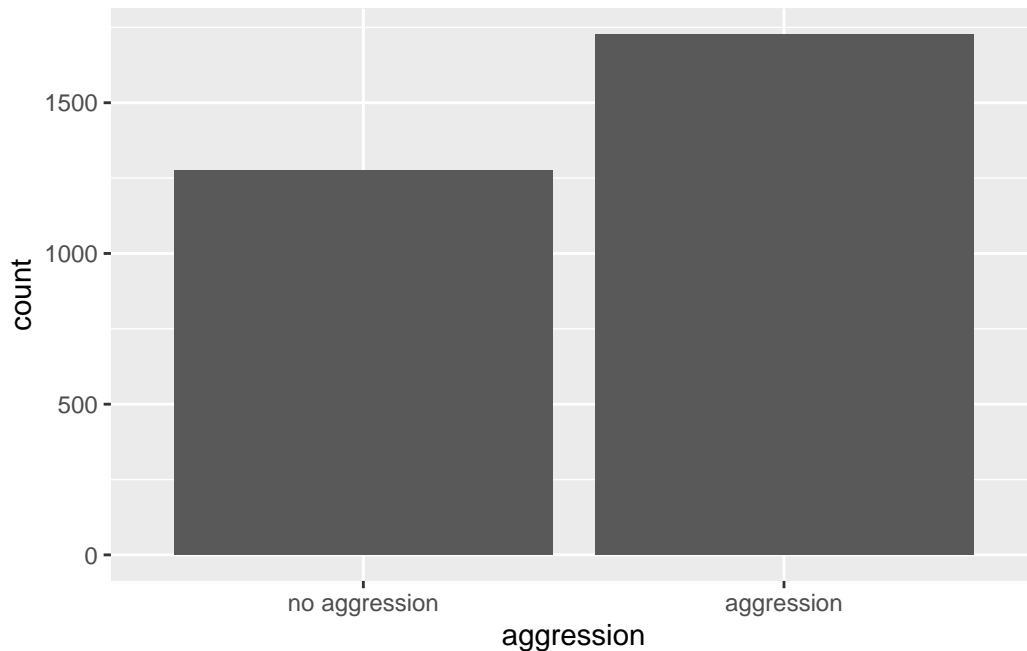
Figure 5.1: Histogram of Gender Inequality Index

5.4.2 One Categorical Variable

Make sure R knows this is a categorical variable.

```
MICSsimulated$aggression <-
  factor(MICSsimulated$aggression, # original numeric variable
        levels = c(0, 1),
        labels = c("no aggression", "aggression"),
        ordered = TRUE) # whether order matters

ggplot(MICSsimulated, # the data I am using
       aes(x = aggression)) + # the variable I am using
  geom_bar() # how I am graphing it
```



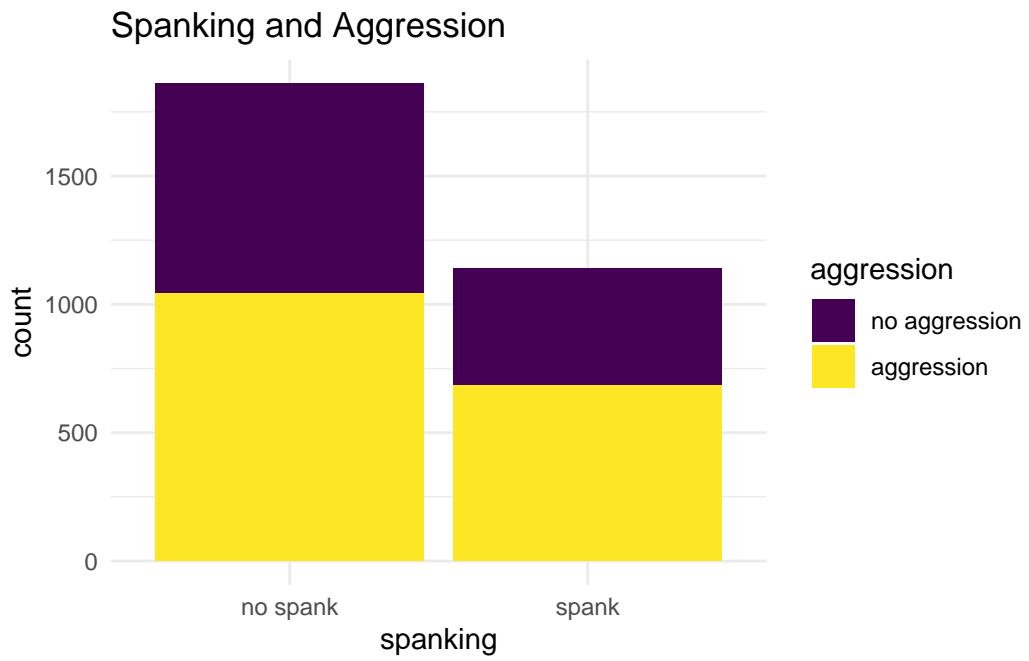
5.5 Add Some Options³

Make sure R knows this is a categorical variable.

```
MICSsimulated$cd1 <-
  factor(MICSsimulated$cd1, # original numeric variable
        levels = c(0, 1),
        labels = c("no spank", "spank"),
        ordered = TRUE) # whether order matters

ggplot(MICSsimulated, # the data I am using
       aes(x = cd1,
           fill = aggression)) + # the variables I am using
  geom_bar() + # graph with bars
  labs(title = "Spanking and Aggression",
       x = "spanking",
       y = "count") +
  theme_minimal() # theme
```

³Notice how use of `fill` governs both the use of color in the graph below, as well as the legend that is produced in the graph.



6 Quantitative Data Analysis

6.1 Introduction

A great deal of data analysis and visualization involves the same core set of steps.

have a question → get data → process and clean data → analyze data

6.2 Some Tools for Analysis

Below we describe some simple data cleaning with R. We begin, however, by comparing several different tools for analysis including: Excel, Google Sheets, R, and Stata.

Tool	Cost	Ease of Use	Analysis Capabilities	Suitability for Large Data	Keep Track of Complicated Workflows
Excel	Comes installed on many computers	Easy	Limited	Difficult when $N > 100$	Difficult to Impossible
Google Sheets	Free with a Google account	Easy	Limited	Difficult when $N > 100$	Difficult to Impossible
R	Free	Challenging	Extensive	Excellent with large datasets	Yes, with script
Stata	Some cost	Learning Curve but Intuitive	Extensive	Excellent with large datasets	Yes, with command file

6.3 Working With R

6.3.1 Our Data

We take a look at our *simulated* data.

```
load("./simulate-data/MICSsimulated.RData") # data in R format

labelled::look_for(MICSsimulated) # look at data
```

pos	variable	label	col_type	values
1	id	id	int	
2	country	country	int	
3	GII	gender inequality index	int	
4	cd1	spank	int	
5	cd2	beat	int	
6	cd3	shout	int	
7	cd4	explain	int	
8	aggression	aggression	int	

6.3.2 Cleaning Data

There are some basic data cleaning steps that are common to many projects.

- Only keep the variables of interest. Section [6.3.2.1](#)
- Add variable labels (if we can). Section [6.3.2.2](#)
- Add value labels (if we can). Section [6.3.2.3](#)
- Recode outliers, values that are errors, or values that should be coded as missing Section [6.3.2.4](#)

Much of R's functionality is accomplished through writing *code*, that is saved in a *script*. Notice how—as our tasks get more and more complicated—the saved script provides documentation for the decisions that we have made with the data.

6.3.2.1 Only keep the variables of interest.

We can easily accomplish this with the `subset` function

```
mynewdata <- subset(MICSsimulated,
                    select = c(id, country, aggression))
```

```
head(mynewdata)
```

	id	country	aggression
1	1	1	1
2	2	1	1
3	3	1	1
4	4	1	1
5	5	1	0
6	6	1	1

6.3.2.2 Add variable labels (if we can).

Adding *variable labels* is still somewhat new in R. The `labelled` library allows us to add or change variable labels. However, not every library in R recognizes *variable labels*.

```
library(labelled)

var_label(MICSsimulated$id) <- "id"

var_label(MICSsimulated$country) <- "country"

var_label(MICSsimulated$cd4) <- "explain"
```

6.3.2.3 Add value labels (if we can).

In contrast, *value labels* are straightforward in R, and can be accomplished by creating a *factor variable*. Below we demonstrate how to do this with the happy variable.

```
MICSsimulated$cd4 <- factor(MICSsimulated$cd4,
                           levels = c(0, 1),
                           labels = c("Did not explain",
                                       "Explained"))

head(MICSsimulated)
```

id	country	GII	cd1	cd2	cd3	cd4	aggression
----	---------	-----	-----	-----	-----	-----	------------

1	1	1	20	0	1	0	Explained	1
2	2	1	20	0	0	1	Did not explain	1
3	3	1	20	0	0	0	Explained	1
4	4	1	20	1	1	0	Explained	1
5	5	1	20	0	0	1	Explained	0
6	6	1	20	1	0	0	Explained	1

6.3.2.4 Recode outliers, values that are errors, or values that should be coded as missing.

We can easily accomplish this using Base R's syntax for recoding: `data$variable[rule] <- newvalue`.

```
MICSsimulated$aggression[MICSsimulated$aggression > 1] <- NA # recode > 1 to NA
```

```
MICSsimulated$GII[MICSsimulated$GII > 100] <- NA # recode > 100 to NA
```

```
head(MICSsimulated)
```

	id	country	GII	cd1	cd2	cd3	cd4	aggression
1	1	1	20	0	1	0	Explained	1
2	2	1	20	0	0	1	Did not explain	1
3	3	1	20	0	0	0	Explained	1
4	4	1	20	1	1	0	Explained	1
5	5	1	20	0	0	1	Explained	0
6	6	1	20	1	0	0	Explained	1

6.3.3 Simple Analysis

Our first step in analysis is to discover what kind of variables we have. We need to make a distinction between *continuous variables* that measure things like mental health or neighborhood safety, or age, and *categorical variables* that measure non-ordered categories like religious identity or gender identity.

- For continuous variables, it is most appropriate to take the *average* or *mean*.
- For categorical variables, it is most appropriate to generate a *frequency table*.

As a mostly command based language, R relies on the idea of `do_something(dataset$variable)`.

```
summary(MICSsimulated$GII) # descriptive statistics for GII
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
15.0	22.0	24.0	24.2	27.0	31.0

```
table(MICSsimulated$cd4) # frequency table of cd4
```

Did not explain	Explained
671	2329

References

- Kreft, I., & de Leeuw, J. (1998). *Introducing multilevel modeling*. SAGE Publications. <https://doi.org/10.4135/9781849209366>
- Luke, D. (2004). *Multilevel modeling*. SAGE Publications, Inc. <https://doi.org/10.4135/9781412985147>
- Rabe-Hesketh, S., & Skrondal, A. (2022). Multilevel and longitudinal modeling using Stata. In *Stata Press* (4th ed.). Stata Press.
- Raudenbush, S. W., & Bryk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods* (pp. xxiv, 485 p.). Sage Publications.
- Singer, J. D., & Willett, J. B. (2003). Applied longitudinal data analysis : Modeling change and event occurrence. In *Applied longitudinal data analysis : modeling change and event occurrence*. Oxford University Press.

A Simulating MICS Data

This appendix details the process of creating the simulated MICS data that is employed in the examples on this website.

MICS data are freely available, but usage of MICS requires completing a user agreement, and registering for a user account, on the MICS website, and thus MICS data should not be shared openly on a public website.

This Appendix is highly technical. It is not necessary to understand this Appendix to benefit from the rest of this website. However, the details of creating this simulated data may be of interest to some users.

A.1 Call Relevant Libraries

We need to call a number of relevant R libraries to simulate the data.

```
library(tibble) # new dataframes

library(ggplot2) # nifty graphs

library(labelled) # labels

library(haven) # write Stata

library(tidyr) # tidy data

library(dplyr) # wrangle data

library(lme4) # multilevel models

library(sjPlot) # nice tables
```

A.2 Setup Some Basic Parameters of the Data

Because simulation is a random process, we set a *random seed* so that the simulation produces the same data set each time it is run.

We are going to simulate data with 30 countries, and 100 individuals per country.

```
set.seed(1234) # random seed

N_countries <- 30 # number of countries

N <- 100 # sample size / country
```

A.3 Simulate Data Based on MICS

This is multilevel data where individuals are nested, or clustered, inside countries. Excellent technical and pedagogical discussions of multilevel models can be found in Raudenbush & Bryk (2002), Singer & Willett (2003), Rabe-Hesketh & Skrondal (2022), Luke (2004), and Kreft & de Leeuw (1998).

A.3.1 Level 2

Simulating the second level of the data is relatively easy. We simply need to provide the number of countries, and then generate random effects for each country. Random effects are discussed in the above references, but essentially represent country level differences in the data.

We also create GII, a gender inequality index variable, since this is a country level, or Level 2 variable.

```
country <- seq(1:N_countries) # sequence 1 to 30

GII <- rbinom(N_countries, 100, .25) # gender inequality index

u0 <- rnorm(N_countries, 0, .25) # random intercept

u1 <- rnorm(N_countries, 0, .05) # random slope

randomeffects <- data.frame(country, GII, u0, u1) # dataframe of random effects
```

A.3.2 Level 1

Simulating the Level 1 data is more complex.

We `uncount` the data by 100 to create 100 observations for each country. We then create an `id` number.

We create randomly simulated parental discipline variables with proportions similar to those in MICS.

Lastly, we need to create the dependent variable. Because this is a dichotomous outcome, the process is somewhat complex. We need to create a linear combination `z`, using regression weights derived from MICS. We then calculate predicted probabilities, and lastly generate a dichotomous `aggression` outcome from those probabilities.

```
MICSsimulated <- randomeffects %>%
  uncount(N) %>% # N individuals / country
  mutate(id = row_number()) %>% # unique id
  mutate(cd1 = rbinom(N * N_countries, 1, .38), # spank
         cd2 = rbinom(N * N_countries, 1, .05), # beat
         cd3 = rbinom(N * N_countries, 1, .64), # shout
         cd4 = rbinom(N * N_countries, 1, .78)) %>% # explain
  mutate(z = 0 + # linear combination based on MICS
         .01 * GII +
         .23 * cd1 +
         .52 * cd2 +
         .42 * cd3 +
         -.21 * cd4 +
         u0) %>%
  mutate(p = exp(z) / (1 + exp(z))) %>% # probability
  mutate(aggression = rbinom(N * N_countries, 1, p)) %>% # binomial y
  select(id, country, GII,
         cd1, cd2, cd3, cd4,
         aggression)
```

A.3.3 Variable Labels

We add variable labels to the data which will help us to understand the data as we analyze it.

```
var_label(MICSsimulated$id) <- "id"
```

Table A.1: ?(caption)

pos	variable	label	col_type	values
1	id	id	int	
2	country	country	int	
3	GII	gender inequality index	int	
4	cd1	spank	int	
5	cd2	beat	int	
6	cd3	shout	int	
7	cd4	explain	int	
8	aggression	aggression	int	

```

var_label(MICSsimulated$country) <- "country"

var_label(MICSsimulated$GII) <- "gender inequality index"

var_label(MICSsimulated$cd1) <- "spank"

var_label(MICSsimulated$cd2) <- "beat"

var_label(MICSsimulated$cd3) <- "shout"

var_label(MICSsimulated$cd4) <- "explain"

var_label(MICSsimulated$aggression) <- "aggression"

labelled::look_for(MICSsimulated) # list out variable labels

```

A.4 Explore The Simulated Data With A Graph

Exploring the simulated data with a graph helps us to ensure that we have simulated plausible data.

```

ggplot(MICSsimulated,
  aes(x = cd1, # x is spanking
      y = aggression, # y is aggression
      color = factor(country))) + # color is country
  geom_smooth(method = "glm", # glm smoother
             method.args = list(family = "binomial"),

```

```

    alpha = .1) + # transparency for CI's
labs(title = "Aggression as a Function of Spanking",
     x = "spank",
     y = "aggression") +
scale_color_viridis_d(name = "Country") + # nice colors
theme_minimal()

```

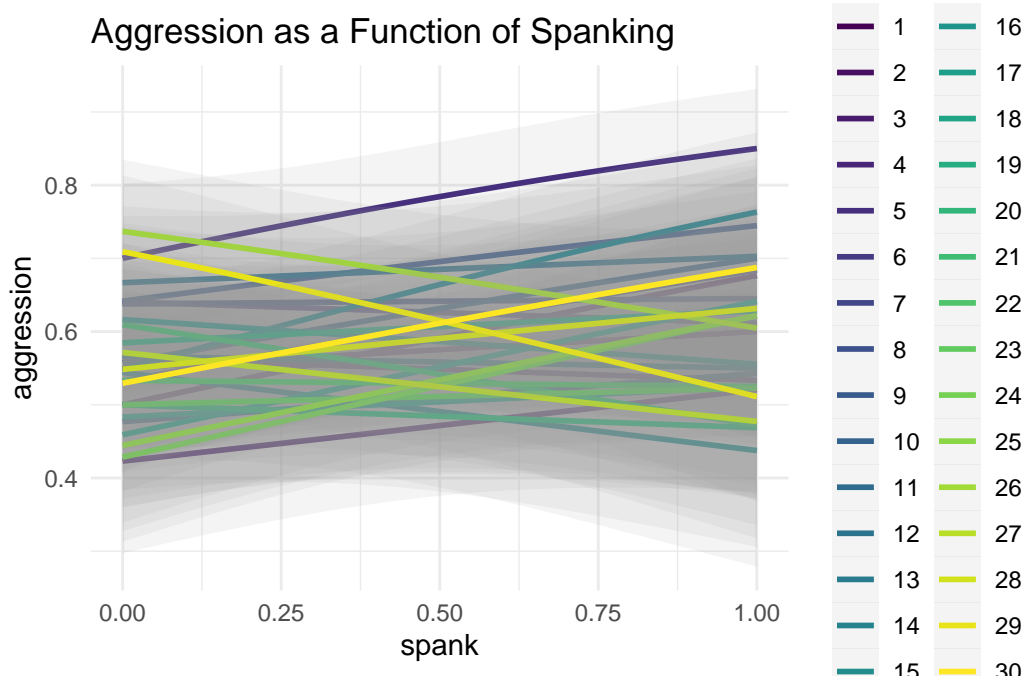


Figure A.1: Graph of Simulated Data

A.5 Explore The Simulated Data With A Logistic Regression

Similarly, exploring the data with a logistic regression confirms that we have created plausible data.

```

fit1 <- glmer(aggression ~ cd1 + cd2 + cd3 + cd4 + GII +
              (1 | country),
              family = "binomial",
              data = MICSsimulated,
              control = glmerControl(optimizer = "bobyqa"))

```



```
tab_model(fit1, # nice table
          transform = NULL) # untransformed estimates
```

A.6 Write data to various formats

Lastly, we write the data out to various formats: R, Stata, and SPSS.

```
save(MICSsimulated,
     file = "./simulate-data/MICSsimulated.RData") # R

write_dta(MICSsimulated,
          "./simulate-data/MICSsimulated.dta") # Stata

write_sav(MICSsimulated,
          "./simulate-data/MICSsimulated.sav") # SPSS
```

Table A.2: ?(caption)

aggression	
Predictors	
Log-Odds	
CI	
p	
(Intercept)	
-0.42	
-1.05 – 0.21	
0.190	
spank	
0.16	
0.01 – 0.32	
0.034	
beat	
0.61	
0.25 – 0.96	
0.001	
shout	
0.35	
0.20 – 0.51	
<0.001	
explain	
-0.41	
-0.59 – -0.23	
<0.001	
gender inequality index	
0.03	
0.01 – 0.05	
0.015	
Random Effects	
₂	
3.29	
₀₀ country	
0.03	
ICC	
0.01	
N _{country}	
30	
Observations	
3000	
Marginal R ² / Conditional R ²	
0.029 / 0.037	