Promoting Data Science

Volume 3 2023 Number 2

Journal of Behavioral Data Science V3N2 (2023)

https://isdsa.org

JOURNAL OF BEHAVIORAL DATA SCIENCE

ISBN: 2575-8306 (Print) 2574-1284 (Online)

https://jbds.isdsa.org

JOURNAL OF BEHAVIORAL DATA SCIENCE

No Publication Charge and Open Access

jbds@isdsa.org

List of Articles

Journal of Behavioral Data Science, 2023, *3* (2), 1–14.
DOI:https://doi.org/10.35566/jbds/v3n2/orourke

Considering the Distributional Form of Zeroes When Calculating Mediation Effects with Zero-Inflated Count Outcomes

Holly P. O'Rourke[1][0000−0002−2927−0333] and Da Eun Han[2][0000−0001−8699−439X]

[1] Arizona State University
holly.orourke@asu.edu
[2] University of Illinois at Urbana-Champaign
duoen10@gmail.com

Abstract. Recent work has demonstrated how to calculate conditional mediated effects for mediation models with zero-inflated count outcomes in a non-causal framework (O'Rourke & Vazquez, 2019); however, those formulas do not distinguish between logistic and count portions of the data distribution when calculating mediated effects separately for zeroes and counts. When calculating conditional mediated effects for the counts in a zero-inflated count outcome Y, the *b* path should use the partial derivative of the log-linear regression equation for X and M predicting Y. When calculating conditional mediated effects for the zeroes, the *b* path should use the partial derivative of the logistic regression equation for X and M predicting Y instead of the log-linear equation. This paper presents adjustments to the analytical formulas of conditional mediated effects for mediation with zero-inflated count outcomes when zeroes and counts are differentially predicted. Using a Monte Carlo simulation, we also empirically show that these adjustments produce different results than when the distributional form of zeroes is ignored.

Keywords: Mediation analysis · Count outcomes · Zero-inflation · ZIP · ZINB · Hurdle models

1 Introduction

Many theories in the social and behavioral sciences specify indirect mechanisms by which predictors influence outcomes. These mechanisms, also known as mediators, are incorporated into such theories through the use of mediation models. Mediation models are widely applied to theories of human behavior and test the indirect influence of a predictor variable (X) on an outcome (Y) via a mediator (M). Much methodological research on the mediation model has focused on models where the endogenous variables M and Y are continuously distributed and assume linear associations, and several extensions have been proposed as well for

models where M and Y are categorical (i.e., binary or count) variables that are modeled with logistic or other exponential family distributions (Coxe & MacKinnon, 2010; Gilula, 2012; Iacobucci, 2012; Imai, Keele, & Tingley, 2010; Mackinnon, 2008; MacKinnon & Cox, 2012; Mackinnon & Dwyer, 1993; Preacher, 2015; Valeri & VanderWeele, 2013; VanderWeele, Zhang, & Lim, 2016). However, these methods are not appropriate for use where categorical endogenous variables contain zero-inflation.

Zero-inflation occurs when the proportion of observations with a value of zero on a particular variable is larger than what is expected from the variable's typical zero-uninflated distribution (for example, Poisson or negative binomial if a variable is a measure of counts). Zero-inflated (ZI) count variables are common in the social sciences. For example, consider a study of externalizing behaviors in middle school; for a given count variable measuring bullying as "number of times child was a bully in the past month", many students would have a score of zero because most children do not engage in bullying behaviors. Another example from health intervention research would be measuring drinking outcomes in a study designed to help adults with alcohol use disorder quit drinking. For a drinking count variable measured as "number of drinks consumed in the past week", many participants would have a score of zero because they are actively trying to refrain from drinking.

The traditional methods cited above for categorical mediation analysis are not equipped to handle excess zeroes in the outcome, and using these models to fit data with zero-inflated distributions may result in biased estimates. A technical body of literature does exist for causal inference methods to assess mediation with categorical variables that contain zero-inflation (Cheng et al., 2018; Wang & Albert, 2012) but this literature is not as accessible to applied researchers due to the complexity of its application. The causal literature differs from the general linear model (GLM)-based mediation literature in that it requires a working knowledge of causal inference frameworks that involve formulas for probability, and causal methods often require additional sensitivity analyses for a formal test of mediation. Furthermore, the effects involved in mediation from the causal inference literature are defined differently from GLM mediation effects, requiring the calculation and interpretation of multiple effects to determine mediation even in simple cases.

Recent work on mediation for ZI counts has applied Geldhof, Anthony, Selig, and Mendez-Luck (2018)'s method of calculating mediation effects for count data that are conditional upon values of X to mediation models with zero-inflated count outcomes using a modeling framework that does not come from causal inference (O'Rourke & Vazquez, 2019). In this method, mediation effects are calculated separately for zeroes and counts when Y is zero-inflated. However, that method does not account for the unique distributional nature of the zeroes, as zeroes are predicted using the binomial logistic model while counts are fitted using the log-linear model. This article illustrates a revised formula for calculating the mediation effect for the zeroes when zeroes and counts are differentially predicted. We also demonstrate via Monte Carlo simulation that the

revised formula produces different results than the original formula that does not distinguish between zeroes and counts in a zero-inflated outcome.

1.1 Mediation

The simplest single-mediator model is described by two OLS regression equations using notation from Mackinnon (2008).

$$Y = i_1 + bM + c'X + e_1 \tag{1}$$

$$M = i_2 + aX + e_2 \tag{2}$$

In these equations, X is the predictor, M is the mediator, and Y is the outcome. From Equation 1, the influence of M on Y is known as the b parameter, and the influence of X on Y controlling for M is known as the c' parameter (also known as the "direct effect"). From Equation 2, the influence of X on M is known as the a parameter. The parameters i_1 and i_2 are model intercepts and e_1 and e_2 are model errors. The mediated effect that is the focus of this article is specified as the product of the a and b parameters (ab), commonly referred to in the mediation literature (and hereafter referred to) as the "mediated effect". Other specifications of the mediated effect and their equalities with respect to count outcomes are described elsewhere (Coxe & MacKinnon, 2010; MacKinnon, Lockwood, Brown, Wang, & Hoffman, 2007; Mackinnon, 2008; O'Rourke & Vazquez, 2019).

Two common approaches to significance testing in mediation are the causal steps (Baron & Kenny, 1986; Judd & Kenny, 1981; MacKinnon, Lockwood, Hoffman, West, & Sheets, 2002) and product of coefficients (Sobel, 1982) approaches. The recommended test from the causal steps approach is the Joint Significance test, which has the best balance of power and Type I error (MacKinnon et al., 2002). The Joint Significance test uses individual $z-$ or $t-$tests of estimates of the respective a and b parameters to assess significance: if both tests are statistically significant, we can conclude that mediation is present. The product of coefficients approach was developed using a derived asymptotic standard error (Sobel, 1982) to compute a $z-$test for the mediated effect ab. More recently, it has become common to assess significance of the mediated effect by using bootstrapping to create asymmetric confidence intervals for ab (MacKinnon, Lockwood, & Williams, 2004). Bootstrapping is used because ab is a product of two variables and so it is not normally distributed (Aroian, 1947; Craig, 1936), meaning traditional formulas that assume a normal distribution of z produce biased confidence intervals for ab.

Mediation analysis is conducted with count outcomes in a similar manner to the approach described above. The difference is that instead of a normal distribution of continuous Y (the assumption under linear regression), the count outcome Y is assumed to have a Poisson distribution, negative binomial (NB) distribution if overdispersed, or beta-binomial distribution if overdispersed with a restricted upper bound. If Y is a count outcome, Poisson, NB, or beta-binomial regression can be used to fit the model specified in Equation 1, and (assuming continuous M) Equation 2 can be assessed as usual with linear regression.

1.2 Zero-Inflated Counts

Each of the models described for count outcomes has a ZI counterpart: the ZI Poisson (ZIP), ZI negative binominal (ZINB), or ZI beta-binomial (ZIBB) models. These are known as zero-inflated generalized linear models (ZI-GZLMs). Hurdle models, similar to ZI-GZLMs, can also be used to model ZI count outcomes. However, zeroes are treated differently in hurdle models compared with ZI models. ZI-GZLMs assume that there are two kinds of zeroes: "structural" (i.e., excess) zeroes that will never take on another value, and "sampling" zeroes that have some potential to be non-zero. (For the remainder of this paper, when we refer to the zeroes in Y, we are referring to the structural zeroes that are modeled separately from the counts and sampling zeroes.) Hurdle models do not make the structural vs. sampling zero distinction, but instead assume that all zeroes are generated from the same process. In other words, hurdle models treat all zeroes as structural excess zeroes that are the only source of overdispersion in the data, and these models do not include an additional probability mass which distinguishes structural zeroes from counts and sampling zeroes in other ZI-GZLMs.

In ZI-GZLMs, the probability of an occurrence of an excess zero is modeled as follows.

$$z \sim \text{Bernoulli}(\pi) \tag{3}$$

Where π is the probability of observing excess zeroes. Assuming a ZI Poisson distribution (the simplest in terms of parameterization because mean and variance are assumed to be equal), and where λ is the mean count from the Poisson distribution, the probability mass function of a ZIP model is as follows.

$$P(Y = 0) = \pi + (1 - \pi)e^{-\lambda} \tag{4}$$

$$P(Y \neq 0) = (1 - \pi)(\frac{\lambda^Y}{Y!})e^{-\lambda} \tag{5}$$

1.3 Mediation for Zero-Inflated Counts

One recent non-causal method of assessing mediation for count outcomes suggested computing multiple conditional mediated effects for chosen values of the predictor X (Geldhof et al., 2018), representing the nonlinear relation between X and Y as several conditionally linear relations that differ across values of X (Stolzenberg, 1980). This method was extended to mediation models for ZI count outcomes (O'Rourke & Vazquez, 2019) by calculating two separate sets of conditional mediated effects: one for the zeroes and one for the counts. For a model with any measurement level of X, continuous M, and ZI count Y, b paths were calculated separately for structural zeroes and counts using the first partial derivative with respect to M of the loglinear mediation regression equation shown in Equation 1. This loglinear mediation regression equation is given below.

$$\hat{Y} = e^{i_1 + bM + c'X} \tag{6}$$

The first partial derivative of Equation 6 being the following.

$$b_{LL} = \frac{\partial \hat{Y}}{\partial M} = b(e^{i_1 + bM + c'X}) \tag{7}$$

The formula in Equation 7 for b_{LL} (b path from the *loglinear* equation) was used in conjunction with the a path from Equation 2 to calculate conditional mediated effects as follows.

$$a * b_{LL} \tag{8}$$

Two sets of k conditional mediated effects were calculated separately for zeroes and counts using the formula in Equation 8, with k being equal to the number of chosen values of X (this number would typically be $k = 2$ for binary X, $k = 3$ for continuous X at low, medium. and high values) and M fixed at its mean. Equation 8 was used to calculate sets of conditional mediated effects for both ZIP and ZINB models, as both Poisson and negative binomial models utilize log link functions.

1.4 Distributional Form for Zeroes

The method described above produced the desired sets of conditional mediated effects, however, using the partial derivative formula b_{LL} for both the structural zeroes and the counts disregarded the form of the assumed distribution for the structural zeroes. Specifically, the structural zeroes are modeled with a logistic distribution. Therefore, the logistic regression for predicting structural zeroes has a logit link function of

$$ln(\frac{\pi}{1-\pi}) \tag{9}$$

The mean function corresponding to this logit link function is as follows.

$$\hat{Y} = \frac{e^{i_1 + bM + c'X}}{e^{i_1 + bM + c'X} + 1} \tag{10}$$

Taking the first partial derivative of Equation 10 with respect to M gives us b_{LG} (the b path from the *logistic* mediation regression equation).

$$b_{LG} = \frac{\partial \hat{Y}}{\partial M} = b\frac{e^{i_1 + bM + c'X}}{(e^{i_1 + bM + c'X} + 1)^2} \tag{11}$$

This would result in a conditional mediated effect for the zeroes of

$$a * b_{LG} \tag{12}$$

Both of the first partial derivative formulas for b presented here are known quantities for estimating a mediation path with either a count or binary non-ZI endogenous variable (Geldhof et al., 2018; Li, Schneider, & Bennett, 2007), but they have not been used in tandem to handle two-part mediation models for ZI endogenous variables.

The current paper aims to utilize both of these formulas for the b path, and to demonstrate that using the b path from the logit link function b_{LG} when calculating mediated effects for the zeroes results in different estimates of the conditional mediated effects than using the b path from the log link function b_{LL} for both zeroes and counts. In the next section, we describe a Monte Carlo simulation study that demonstrates that the formulas produce different estimates of the conditional mediated effects for the zeroes (hereafter referred to as "conditional mediated effects").

2 Simulation Study

2.1 Simulation Conditions

We conducted a Monte Carlo simulation in R 4.2.3 in conjunction with Mplus version 8.10 (Muthén & Muthén, 2017). In this simulation study, we manipulated two factors: Sample size ($N = 100$, 250, 500, 750, and 1500) and population distribution of the counts in the outcome (Poisson vs. negative binomial). Simulation manipulations resulted in 2 x 5 = 10 conditions. Sample sizes were chosen to represent a range from small to large samples based on sample sizes commonly observed in the behavioral sciences. Manipulation of sample size allowed for us to examine possible effects of sample size on results by examining whether the difference in estimates of the conditional mediated effect grew smaller as sample size increased. The Poisson and negative binomial distributions for counts were chosen as the two distributions that are most commonly observed and practically applied with ZI-GzLMs. Differences in estimates of the conditional mediated effects were expected to be stable across the two levels of distribution of the count outcomes.

Population parameter values were not varied over conditions, and the parameter values used in each simulation model are given in Table 1.

Table 1. Simulation Study Parameter Values

Parameter	Population Value
a	0.59
b (Zeroes)	-0.14
b (Counts)	0.14
c' (Zeroes)	-0.01
c' (Counts)	0.01
μ_M	0
σ_M	1
μ_Y (Zeroes)	0
μ_Y (Counts)	3
ϕ*	1

*for ZINB models only

Parameter value magnitudes were chosen to reflect large (0.59) effect size for a and small (0.14) effect sizes for b as established in prior simulation research on single mediator models (Fritz & MacKinnon, 2007; MacKinnon et al., 2002; O'Rourke & MacKinnon, 2015), and parameter value signs were chosen such that conditional mediated effects would differ in sign for the zeroes and counts, as discussed below. The c' path was assigned a very small effect size in accordance with a model that would approach full mediation, a condition where $c' = 0$ (Mackinnon, 2008), but still would factor into calculations of conditional mediated effects. Table 2 shows population calculations of the conditional mediated effects across values of X based on the parameter values given in Table 1, using both the log link and logit link b paths.

Table 2. Calculation of Population Conditional Mediated Effects for Log Link and Logit Link Formulas

Log Link Function b path		
	X = 0	X = 1
General Formula	$a * b(e^{i_1+bM+c'X})$	$a * b(e^{i_1+bM+c'X})$
$e^{i_1+bM+c'X}$	$e^{0+(-0.14)(0)+(-0.01)(0)} = 1$	$e^{0+(-0.14)(0)+(-0.01)(1)} = 1.01$
$b(e^{i_1+bM+c'X})$	$-0.14 * 1 = -0.14$	$-0.14 * 1.01 = -0.141$
$a * b(e^{i_1+bM+c'X})$	$0.59 * -0.14 = -0.0826$	$0.59 * -0.141 = -0.0834$
Conditional Mediated Effect	**-0.0826**	**-0.0834**
Logit Link Function b path		
	X = 0	X = 1
General Formula	$a * b\frac{e^{i_1+bM+c'X}}{(e^{i_1+bM+c'X}+1)^2}$	$a * b\frac{e^{i_1+bM+c'X}}{(e^{i_1+bM+c'X}+1)^2}$
$e^{i_1+bM+c'X}$	$e^{0+(-0.14)(0)+(-0.01)(0)} = 1$	$e^{0+(-0.14)(0)+(-0.01)(1)} = 1.01$
$\frac{e^{i_1+bM+c'X}}{(e^{i_1+bM+c'X}+1)^2}$	$\frac{1}{(1+1)^2} = 0.25$	$\frac{1.01}{(1.01+1.01)^2} = 0.2499$
$b\frac{e^{i_1+bM+c'X}}{(e^{i_1+bM+c'X}+1)^2}$	$-0.14 * 0.25 = -0.035$	$-0.14 * 0.2499 = -0.0349$
$a * b\frac{e^{i_1+bM+c'X}}{(e^{i_1+bM+c'X}+1)^2}$	$0.59 * -0.035 = -0.0207$	$0.59 * -0.0349 = -0.0206$
Conditional Mediated Effect	**-0.0207**	**-0.0206**

2.2 Data Generation and Data Analysis

The R MplusAutomation package (Hallquist & Wiley, 2018) was used to simulate data. For each of the conditions, 500 replications with complete data were simulated. The paths related to mediation (b and c') were specified to be equal in magnitude but opposite in sign for the zeroes and counts in Y in accordance with commonly observed patterns of results in applied ZI-GZLMs. Binary X was simulated with a Bernoulli distribution with $X \in 0, 1$, and M was simulated with a continuous Gaussian distribution $M \sim N(0, 1)$. The counts in Y were simulated to have a mean of 3 and the zeroes in Y, a mean of 0. Replications for Y with a ZIP distribution did not include a dispersion parameter, and when Y

was simulated to have a ZINB distribution, the dispersion parameter was $\phi = 1$ as specified by the variance of Y in the Mplus MODEL command.

After all datasets were generated, the R MplusAutomation package was then used to create and run Mplus scripts analyzing all replications within each condition and then import results into R. This process was repeated for each of the 10 conditions. For each replication, a ZI-GZLM was fitted to the data using Maximum Likelihood estimation. Conditional mediated effects were calculated at X = 0 and X = 1 using the Mplus "Model Constraint" command. For the zeroes in Y, sets of conditional mediated effects were calculated using both the original method with b_{LL} for the b path and the revised method with b_{LG} for the b path. This resulted in four conditional mediated effects for comparison in further analyses. Bootstrapped confidence intervals were also generated to assess significance of each conditional mediated effect. Sample Mplus and R scripts can be found on the GitHub project at https://github.com/horourke/MZI2.

2.3 Simulation Study Outcomes and Outcome Analyses

We assessed differences in results for the conditional mediated effect estimates by examining relative parameter difference (i.e., relative bias for the ab_{LG} estimate). The relative difference was calculated as the difference between the population value of ab_{LG} and the respective estimates $\widehat{ab_{LL}}$ and $\widehat{ab_{LG}}$, over the population value of ab_{LG}.

$$\frac{ab_{LG} - \widehat{ab_{LL}}}{ab_{LG}} \tag{13}$$

$$\frac{ab_{LG} - \widehat{ab_{LG}}}{ab_{LG}} \tag{14}$$

Efficiency was calculated using the standard deviations of the raw estimates of the conditional mediated effects averaged across each condition. We also examined statistical power for each condition, calculated as the proportion of replications for which the p value associated with each conditional mediated effect was less than .05 and the bootstrapped confidence intervals of each conditional mediated effect did not include zero.

In preparation for analysis of the relative difference outcome, data were restructured to long format such that use of the b formula (b_{LL} vs. b_{LG}) could be coded as an additional binary predictor of a given outcome. Analyses conducted in R examined the impact of the condition on the dependent variable of interest at the replication level, with one replication considered as one observation. Analysis of variance (ANOVA) was used to investigate the differences in study conditions for relative parameter differences. Analyses were conducted separately for each outcome at X = 0 and X = 1. Factors representing study conditions in each ANOVA were sample size, population distribution of the counts in the outcome, and method of calculating the b path. In addition to main effects, all possible two- and three-way interactions were included as predictors in each ANOVA. Only ANOVA estimates that were significant at $p < .05$ with

corresponding partial η^2 values of .02 (small amount of variance explained) or higher were considered meaningfully significant for the interpretation of results.

3 Results

3.1 Relative Difference

The average relative difference over replications for each condition is shown in Table 3. Results from the ANOVAs for both X = 0 and X = 1 indicated that only the method of calculating the b path (b_{LG} vs. b_{LL}) was a meaningfully significant predictor of relative difference. Method of calculating the b path explained 20.6% and 15.1% of the variability in the outcome respectively, which were large effect sizes (X = 0: $p < .001$, partial $\eta^2 = .206$; X = 1: $p < .001$, partial $\eta^2 = .151$).

Table 3. Relative Difference of Conditional Mediated Effects Collapsed Across Conditions

ZINB				
	X = 0		X = 1	
n	ab_{LL}	ab_{LG}	ab_{LL}	ab_{LG}
100	2.917	-0.052	4.620	-0.101
250	3.226	0.017	3.857	-0.005
500	3.089	0.011	3.330	0.000
750	3.062	0.007	3.176	0.000
1500	3.065	0.009	3.074	0.006
ZIP				
	X = 0		X = 1	
n	ab_{LL}	ab_{LG}	ab_{LL}	ab_{LG}
100	3.105	-0.001	4.599	-0.049
250	3.243	0.029	3.765	0.011
500	3.089	0.012	3.282	0.003
750	3.076	0.012	3.169	0.007
1500	3.066	0.012	3.073	0.010

Examining Table 3 for X = 0, the average relative difference was around 3 or above for all conditions for ab_{LL} estimates. When using the ab_{LG} formula to calculate conditional mediated effects, the average relative difference only reached an absolute value above .05 for the ZINB model at the smallest sample size, and the average relative difference was otherwise extremely small for all conditions using the ab_{LG} formula.

For X = 1, results from Table 3 indicate that the average relative difference of the estimates of ab_{LL} ranged from $[3.073, 4.620]$ for all conditions, with average relative difference decreasing as sample size increased[1]. As with calculations for

[1] Sample size was a statistically significant predictor of relative difference for the ANOVA where X = 1, however the partial $\eta^2 < .02$.

X = 0, the average relative difference of the conditional mediated effects using the ab_{LG} formula only reached an absolute value above .05 for the condition fitting the ZINB model at the smallest sample size, and the average relative difference was otherwise not problematic (i.e., below an absolute value of .05) for all conditions using the ab_{LG} formula.

3.2 Efficiency

For all values of X and regardless of sample size and distribution of outcomes, estimates of ab_{LG} had smaller variability (i.e., were more efficient) than for estimates of ab_{LL}, as shown by the averaged standard deviations of the estimates in Table 4. The difference in efficiency between the two sets of conditional mediated effect estimates decreased monotonically as sample size increased such that the conditional mediated effect estimates calculated with ab_{LL} were least efficient at the smallest sample sizes.

Table 4. Efficiency of Conditional Mediated Effects Collapsed Across Conditions

ZINB				
	X = 0		X = 1	
n	ab_{LL}	ab_{LG}	ab_{LL}	ab_{LG}
100	0.152	0.036	0.242	0.033
250	0.090	0.021	0.119	0.021
500	0.061	0.015	0.073	0.015
750	0.047	0.012	0.053	0.011
1500	0.034	0.008	0.037	0.008
ZIP				
	X = 0		X = 1	
n	ab_{LL}	ab_{LG}	ab_{LL}	ab_{LG}
100	0.147	0.034	0.221	0.032
250	0.086	0.021	0.109	0.020
500	0.059	0.014	0.069	0.014
750	0.045	0.011	0.050	0.011
1500	0.033	0.008	0.035	0.008

3.3 Power

Power values by condition can be found in Table 5. For conditional mediated effects where X = 0, there were negligible differences in power between the methods of calculating the b path. For conditional mediated effects where X = 1, power was slightly larger for estimates of ab_{LG} than for estimates of ab_{LL}. Power increased as the sample size increased for all conditions, and there were negligible differences in power between the ZIP and ZINB conditions. Power never reached a level of .8 (Cohen, 1988) in any of the conditions, likely due to the small magnitude of the b paths.

Table 5. Power of Conditional Mediated Effects Collapsed Across Conditions

ZINB				
	X = 0		X = 1	
n	ab_{LL}	ab_{LG}	ab_{LL}	ab_{LG}
100	0.000	0.008	0.000	0.018
250	0.068	0.106	0.000	0.118
500	0.214	0.250	0.050	0.258
750	0.400	0.418	0.222	0.420
1500	0.692	0.700	0.612	0.702
ZIP				
	X = 0		X = 1	
n	ab_{LL}	ab_{LG}	ab_{LL}	ab_{LG}
100	0.000	0.020	0.000	0.032
250	0.090	0.120	0.000	0.134
500	0.274	0.298	0.084	0.308
750	0.432	0.442	0.286	0.446
1500	0.728	0.730	0.670	0.732

4 Discussion

We used a Monte Carlo simulation to demonstrate the differences in results using log-linear vs. logistic regression equations for zeroes when calculating conditional mediated effects in a mediation model where Y is a ZI count. The conditional mediated effects for the zeroes in Y were calculated using two different b path formulas, b_{LL} and b_{LG}, where b_{LG} used the distributional form of the zeroes. Comparing estimates of ab_{LL} and ab_{LG}, we found that the conditional mediated effects differed significantly in magnitude at both values of X, for both ZINB and ZIP models and across all sample sizes examined. Specifically, results for relative difference showed that estimates of ab_{LL} were significantly different from estimates of ab_{LG}, and that this difference held across sample sizes and outcome distributions. Conditional mediated effects for zeroes calculated using the b_{LG} formula were also more efficient, and when X was non-zero, had slightly higher power. These results indicate that the choice of b path formula has a meaningful impact on the interpretation of results for the conditional mediated effects and should be considered when using this method to conduct mediation analysis with ZI count variables.

For conditional mediated effects calculated at non-zero X, power was slightly higher for ab_{LG} than for ab_{LL}. This means that when using the different formulas for b, we may make different conclusions about the significance of the conditional mediated effects when X is non-zero (for example, we could observe that the conditional mediated effect $ab_{LL[X=1]}$ was not significant and $ab_{LG[X=1]}$ was significant), which further highlights the importance of considering the distributional form of the zeroes when conducting mediation analysis where ZI counts are present in the data.

In this paper, we focused specifically on the mediation model that contained Y as a ZI count, meaning we discussed the issue of separate distributions of zeroes and counts with respect to only the b path (from M to Y) in mediation. However, this issue is applicable as well to models with a ZI count mediator, in which case we would calculate an a path using a log-linear regression equation for counts in M and an a path using a logistic regression equation for zeroes in M. Furthermore, it would be possible to use this method in a model where both M and Y are ZI counts. Under such circumstances, the a path for the counts in M could be calculated using the first partial derivative with respect to X of the log-linear transformation of Equation 2, and the b path for the counts in Y could be calculated using Equation 7. The a and b paths for the zeroes in M and Y would then be calculated using the first partial derivatives of the logit transformations of their respective regression equations (Equation 11 for the b path).

The utilization of these formulas can also be applied to future methodological work on mediation analysis with ZI count variables. The method described in this paper that is an extension of O'Rourke and Vazquez (2019) can be extended to more complex models that are frequently used by applied researchers, such as models that incorporate time (i.e., longitudinal models). This process of mediation can be expanded to ZI-GZLMs for repeated measures nested within individuals that are fitted in the multilevel modeling framework.

It is important for researchers to have accessible methods of assessing mediation in complex nonlinear models. This paper advances accessible methodology in the pursuit of best practices for investigating mediators when data are nonnormal. The simulation results presented here highlight the complexity of calculating mediated effects in models where ZI counts are present.

References

Aroian, L. A. (1947, June). The probability function of the product of two normally distributed variables. *The Annals of Mathematical Statistics*, *18*(2), 265–271. doi: https://doi.org/10.1214/aoms/1177730442

Baron, R. M., & Kenny, D. A. (1986). The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology*, *51*(6), 1173–1182. doi: https://doi.org/10.1037/0022-3514.51.6.1173

Cheng, J., Cheng, N. F., Guo, Z., Gregorich, S., Ismail, A. I., & Gansky, S. A. (2018, September). Mediation analysis for count and zero-inflated count data. *Statistical Methods in Medical Research*, *27*(9), 2756–2774. doi: https://doi.org/10.1177/0962280216686131

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed ed.). Hillsdale, N.J: L. Erlbaum Associates.

Coxe, S., & MacKinnon, D. P. (2010, November). Abstract: Mediation analysis of Poisson distributed count outcomes. *Multivariate Behavioral Research*, *45*(6), 1022–1022. doi: https://doi.org/10.1080/00273171.2010.534375

Craig, C. C. (1936, March). On the frequency function of xy. *The Annals of Mathematical Statistics*, *7*(1), 1–15. doi: https://doi.org/10.1214/aoms/1177732541

Fritz, M. S., & MacKinnon, D. P. (2007, March). Required sample size to detect the mediated effect. *Psychological Science*, *18*(3), 233–239. doi: https://doi.org/10.1111/j.1467-9280.2007.01882.x

Geldhof, G. J., Anthony, K. P., Selig, J. P., & Mendez-Luck, C. A. (2018, March). Accommodating binary and count variables in mediation: A case for conditional indirect effects. *International Journal of Behavioral Development*, *42*(2), 300–308. doi: https://doi.org/10.1177/0165025417727876

Gilula, Z. (2012, October). Mediation with categorical variables: Consider ordinal models, empirical Bayes, and alternatives to R2. *Journal of Consumer Psychology*, *22*(4), 599. doi: https://doi.org/10.1016/j.jcps.2012.03.008

Hallquist, M. N., & Wiley, J. F. (2018, July). *MplusAutomation* : An R package for facilitating large-scale latent variable analyses in M *plus*. *Structural Equation Modeling: A Multidisciplinary Journal*, *25*(4), 621–638. doi: https://doi.org/10.1080/10705511.2017.1402334

Iacobucci, D. (2012, October). Mediation analysis and categorical variables: The final frontier. *Journal of Consumer Psychology*, *22*(4), 582–594. doi: https://doi.org/10.1016/j.jcps.2012.03.006

Imai, K., Keele, L., & Tingley, D. (2010). A general approach to causal mediation analysis. *Psychological Methods*, *15*(4), 309–334. doi: https://doi.org/10.1037/a0020761

Judd, C. M., & Kenny, D. A. (1981, October). Process analysis: Estimating mediation in treatment evaluations. *Evaluation Review*, *5*(5), 602–619. doi: https://doi.org/10.1177/0193841X8100500502

Li, Y., Schneider, J. A., & Bennett, D. A. (2007). Estimation of the mediation effect with a binary mediator. *Statistics in Medicine*, *26*(18), 3398–3414. doi: https://doi.org/10.1002/sim.2730

MacKinnon, D., Lockwood, C., Brown, C., Wang, W., & Hoffman, J. (2007, October). The intermediate endpoint effect in logistic and probit regression. *Clinical Trials*, *4*(5), 499–513. doi: https://doi.org/10.1177/1740774507083434

Mackinnon, D. P. (2008). *Introduction to statistical mediation analysis*. Mahwah, NJ: Erlbaum.

MacKinnon, D. P., & Cox, M. G. (2012, October). Commentary on "Mediation analysis and categorical variables: The final frontier" by Dawn Iacobucci. *Journal of Consumer Psychology*, *22*(4), 600–602. doi: https://doi.org/10.1016/j.jcps.2012.03.009

Mackinnon, D. P., & Dwyer, J. H. (1993, April). Estimating Mediated Effects in Prevention Studies. *Evaluation Review*, *17*(2), 144–158. doi: https://doi.org/10.1177/0193841X9301700202

MacKinnon, D. P., Lockwood, C. M., Hoffman, J. M., West, S. G., & Sheets, V. (2002). A comparison of methods to test mediation and other intervening variable effects. *Psychological Methods*, *7*(1), 83–104. doi:

https://doi.org/10.1037/1082-989X.7.1.83

MacKinnon, D. P., Lockwood, C. M., & Williams, J. (2004, January). Confidence limits for the indirect effect: Distribution of the product and resampling methods. *Multivariate Behavioral Research*, *39*(1), 99–128.

Muthén, L. K., & Muthén, B. O. (2017). *Mplus.* Los Angeles, CA: Muthén & Muthén.

O'Rourke, H. P., & Vazquez, E. (2019, July). Mediation analysis with zero-inflated substance use outcomes: Challenges and recommendations. *Addictive Behaviors*, *94*, 16–25. doi: https://doi.org/10.1016/j.addbeh.2019.01.034

O'Rourke, H. P., & MacKinnon, D. P. (2015, June). When the test of mediation is more powerful than the test of the total effect. *Behavior Research Methods*, *47*(2), 424–442. doi: https://doi.org/10.3758/s13428-014-0481-z

Preacher, K. J. (2015, January). Advances in mediation analysis: A survey and synthesis of new developments. *Annual Review of Psychology*, *66*(1), 825–852. doi: https://doi.org/10.1146/annurev-psych-010814-015258

Sobel, M. E. (1982). Asymptotic confidence intervals for indirect effects in structural equation models. *Sociological Methodology*, *13*, 290. doi: https://doi.org/10.2307/270723

Stolzenberg, R. M. (1980). The measurement and decomposition of causal effects in nonlinear and nonadditive models. *Sociological Methodology*, *11*, 459. doi: https://doi.org/10.2307/270872

Valeri, L., & VanderWeele, T. J. (2013, June). Mediation analysis allowing for exposure–mediator interactions and causal interpretation: Theoretical assumptions and implementation with SAS and SPSS macros. *Psychological Methods*, *18*(2), 137–150. doi: https://doi.org/10.1037/a0031034

VanderWeele, T. J., Zhang, Y., & Lim, P. (2016, September). Brief report: Mediation analysis with an ordinal outcome. *Epidemiology*, *27*(5), 651–655. doi: https://doi.org/10.1097/EDE.0000000000000510

Wang, W., & Albert, J. M. (2012, November). Estimation of mediation effects for zero-inflated regression models. *Statistics in Medicine*, *31*(26), 3118–3132. doi: https://doi.org/10.1002/sim.5380

Journal of Behavioral Data Science, 2023, *3* (2), 15–42.
DOI: https://doi.org/10.35566/jbds/v3n2/montoya

Lasso and Group Lasso with Categorical Predictors: Impact of Coding Strategy on Variable Selection and Prediction

Yihuan Huang, Tristan D. Tibbe[0000−0003−0684−8304], Amy Tang, and Amanda K. Montoya[0000−0001−9316−8184]

University of California, Los Angeles, USA
akmontoya@ucla.edu

Abstract. Machine learning methods are being increasingly adopted in behavioral research. Lasso regression performs variable selection and regularization, and is particularly appealing to behavioral researchers because of its connection to linear regression. Researchers may expect properties of linear regression to translate to lasso, but we demonstrate that this assumption is problematic for models with categorical predictors. Specifically, we demonstrate that while the coding strategy used for categorical predictors does not impact the performance of linear regression, it does impact lasso's performance. Group lasso is an alternative to lasso for models with categorical predictors. We investigate the discrepancy between lasso and group lasso models using a real data set: lasso performs different variable selection and has different prediction accuracy depending on the coding strategy, while group lasso performs consistent variable selection but has different prediction accuracy. Using a Monte Carlo simulation, we demonstrate a specific case where group lasso tends to include many variables when few are needed, leading to overfitting. We conclude with recommended solutions to this issue and future directions of exploration to improve the implementation of machine learning approaches in behavioral science. This project shows that when using lasso and group lasso with categorical predictors, the choice of coding strategy should not be ignored.

Keywords: Lasso regression · Categorical predictors · Regularization

1 Introduction

Many behavioral research questions involve categorical predictors, including education, ethnicity, religion, gender, or experimental conditions. Unlike numerical predictors, which typically have a natural scale, to be included in statistical models categorical predictors require researchers to select a method for encoding these variables (i.e., representing the categories using a numeric system). Thus, a single categorical predictor can be represented in a model using different sets of variables, each set embodying the

same predictor but representing different contrasts of the categories. This special property of categorical predictors motivates our exploration of categorical predictors in the case of linear regression and two machine learning algorithms: least absolute shrinkage and selection operator (lasso; Tibshirani, 1996) and group lasso regression (Yuan & Lin, 2006). We explore both variable selection and prediction accuracy for these models and how they are impacted by using different coding strategies for categorical predictors using a real-world data set.

We use a data set focusing on stress during COVID-19 as the primary outcome, measured in over 100,000 participants (Yamada et al., 2021). The *stress* score is an aggregated score from the Perceived Stress Scale (PSS-10) on a 1-5 scale. The data set includes categorical predictors, such as *Education*, *Gender* and *Marital Status*, and continuous predictors, such as *Age* and *Trust in the Country*. The overall goal is to predict participant's *Stress* using the available predictors.

In the remainder of this section, we introduce the three analytical approaches examined in this paper: linear regression, lasso regression, and group lasso regression. We focus on the application of these methods with a continuous outcome and one or more categorical predictors. After introducing these methods, we demonstrate their use with the applied example, exploring peculiar behavior of the machine learning approaches that does not occur with linear regression.

1.1 Linear Regression With Categorical Predictors

Categorical predictors need to be encoded into a set of variables to be included in regression models. Different coding strategies can be implemented, such as dummy, contrast, sequential, or Helmert coding. Tables 1–4 show different ways to encode a categorical variable, *Education*, with 7 categories (no education, up to 6 years of school, up to 9 years of school, up to 12 years of school, some college or equivalent, college degree, PhD/doctorate). Dummy coding uses only 0's and 1's to indicate category membership. One category is selected as the *reference category* (or *reference group*) and is assigned a score of 0 on all indicators. For other categories, only the indicator corresponding to the category is coded as 1 and all other indicators are set to 0 (Table 1). Contrast coding is similar to dummy coding, but the reference category which is coded as all 0 in dummy coding is now coded with all -1 instead, changing the interpretation of the intercept and slope coefficients (Table 2). Sequential coding compares each category to the previous category (Table 3), while Helmert coding examines how each category is compared to the average of all subsequent categories (Table 4). Note that if a categorical variable has k categories, $k-1$ indicators are needed, regardless of the coding strategies used. This type of design matrix is defined as nonsingular because the matrix is invertible. The design matrix has to be nonsingular for linear regression but this is not necessarily the case for lasso or group lasso. In Appendix B we discuss singular matrix options for lasso regression.

In linear regression, each coding scheme represents categories using a different numerical system, which leads to different interpretations of their coefficients. However, each coding scheme always predicts the category mean for each category (or adjusted means if covariates are included), and the explained variance is the same regardless of coding choice (Darlington & Hayes, 2016). Therefore, researchers can choose coding

Table 1: Dummy Coding

Education	D_1	D_2	D_3	D_4	D_5	D_6
1. no education	0	0	0	0	0	0
2. up to 6 years of school	1	0	0	0	0	0
3. up to 9 years of school	0	1	0	0	0	0
4. up to 12 years of school	0	0	1	0	0	0
5. some college or equivalent	0	0	0	1	0	0
6. college degree	0	0	0	0	1	0
7. PhD/doctorate	0	0	0	0	0	1

Note. No education is selected as the reference group (coded 0 on all indicators) and every other category scores 1 on a single indicator and 0 on all other indicators.

Table 2: Contrast Coding

Education	C_1	C_2	C_3	C_4	C_5	C_6
1. no education	1	0	0	0	0	0
2. up to 6 years of school	0	1	0	0	0	0
3. up to 9 years of school	0	0	1	0	0	0
4. up to 12 years of school	0	0	0	1	0	0
5. some college or equivalent	0	0	0	0	1	0
6. college degree	0	0	0	0	0	1
7. PhD/doctorate	-1	-1	-1	-1	-1	-1

Note. PhD/doctorate is selected as the omitted category (coded -1 on all indicators) and every other category scores 1 on a single indicator and 0 on all other indicators.

strategies among all these options according to their needs without concern about model performance. Dummy and contrast coding are often used for nominal categorical variables, while sequential and Helmert coding are particularly helpful when categories are ordered.

When using different coding strategies, the regression coefficients have different interpretations. For example, a researcher might want to know whether *Stress* during the COVID-19 pandemic can be predicted by *Education*. The seven categories within the variable *Education* are encoded by 6 indicators. Linear regression fits the following model:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \beta_5 X_{5i} + \beta_6 X_{6i} + \varepsilon_i, \tag{1}$$

where Y_i is the outcome value for the i^{th} observation (person), X_{ji} is the j^{th} variable to convey category membership for the i^{th} observation, and ε_i is the error term for the i^{th} observation. Equation 1 is the general equation for all coding strategies. If different coding strategies are used, the intercept β_0 and coefficients for different indicators, β_1 through β_6, have different meanings. For example, suppose the fitted linear regression model (with $\hat{Y}_i$ representing the predicted value for the i^{th} observation) is

$$\hat{Y}_i = 2 + 0.3X_{1i} + 1.5X_{2i} + 0.2X_{3i} + 0.5X_{4i} - 0.2X_{5i} - 0.4X_{6i}. \tag{2}$$

The interpretation of these coefficients would depend on which coding strategy was used. If dummy coding was used with no education as the reference group (as in Table

Table 3: Sequential Coding

Education	S_1	S_2	S_3	S_4	S_5	S_6
1. no education	0	0	0	0	0	0
2. up to 6 years of school	1	0	0	0	0	0
3. up to 9 years of school	1	1	0	0	0	0
4. up to 12 years of school	1	1	1	0	0	0
5. some college or equivalent	1	1	1	1	0	0
6. college degree	1	1	1	1	1	0
7. PhD/doctorate	1	1	1	1	1	1

Note. The lowest category scores 0 on all indicators. Each subsequent category scores 1 on one more indicator than the previous.

Table 4: Helmert Coding

Education	H_1	H_2	H_3	H_4	H_5	H_6
1. no education	-6/7	0	0	0	0	0
2. up to 6 years of school	1/7	-5/6	0	0	0	0
3. up to 9 years of school	1/7	1/6	-4/5	0	0	0
4. up to 12 years of school	1/7	1/6	1/5	-3/4	0	0
5. some college or equivalent	1/7	1/6	1/5	1/4	-2/3	0
6. college degree	1/7	1/6	1/5	1/4	1/3	-1/2
7. PhD/doctorate	1/7	1/6	1/5	1/4	1/3	1/2

Note. The lowest indicator scores $-(k-1)/k$ on the first indicator and 0 on all subsequent indicators. The next highest scores $1/k$ on the first indicator, $-(k-2)/(k-1)$ on the second indicator, and 0 on all subsequent indicators. The next highest scores $1/k$ on the first indicator, $1/(k-1)$ on the second indicator, $-(k-3)/(k-2)$ on the third indicator, and 0 on all subsequent indicators. And so on.

1), we would interpret the coefficient for X_4, 0.5, as the difference between the average stress score of individuals with no education and the average stress score with some college education. However, if contrast coding was used (as in Table 2), 0.5 would indicate the difference between the average stress score of individuals with up to 12 years of school and the average score of all categories. If sequential coding was used (as in Table 3), 0.5 would be interpreted as the difference between the average stress score of individuals with some college education and the average stress score of individuals with up to 12 years of school. If Helmert coding was used (as in Table 4), 0.5 would indicate that on average individuals with up to 12 years of school are 0.5 points less stressed than the average of those who have some college education, those who have a college degree and those who have a PhD/Doctorate. The interpretations of the coefficients are inseparable from the coding strategy used.

Different selections of reference categories in dummy and contrast coding and ordering of categories in Helmert and sequential coding can also produce coefficients with different meanings. For example, if no education is the reference category for dummy coding, β_0 represents the average stress score for people with no education and β_1 through β_6 will represent the difference between no education and the corresponding coded category. On the other hand, if up to 6 years of school is the reference cate-

gory, β_0 represents the average stress for individuals with up to 6 years of school, and β_1 through β_6 will represent the difference between "up to 6 years of school" and the corresponding coded category.

Though different ways to code categorical variables produce different model coefficients, they do not affect the predictions/prediction accuracy of linear regression. To demonstrate that linear regression with a categorical predictor will predict the same category means for each coding scheme, we used *Education* to predict *Stress*. We randomly sampled 10,000 participants from the COVID-19 Stress Data (Yamada et al., 2021) to serve as our sample data set, and then we randomly split our sample into training (80%) and test (20%) data. Next, we fit linear regression on the training data set with four different coding strategies from Tables 1 - 4 applied to the variable *Education*. Table 5 contains the model coefficients.

Table 5: Linear Regression Example for Coding

Coefficient	Dummy	Contrast	Sequential	Helmert
β_0	2.852	2.955	2.852	2.955
β_1	0.031	-0.103	0.031	0.121
β_2	0.110	-0.072	0.079	0.107
β_3	0.145	0.007	0.035	0.036
β_4	0.161	0.041	0.016	0.001
β_5	0.138	0.058	-0.023	-0.022
β_6	0.139	0.035	0.001	0.001

Note. Each column of the table represents one coding strategy and rows represent the coefficients of the indicator X_j for each coding strategy.

Using the values of X_1–X_6 from Table 1–4 and the coefficient estimates from Table 5, we reconstruct the predicted score (i.e., category mean) for the "some college or equivalent" category for dummy, contrast, sequential, and Helmert coding respectively.

$$2.852+0.031(0)+0.110(0)+0.145(0)+0.161(1)+0.138(0)+0.139(0)=3.013 \quad (Dummy)$$

$$2.955-0.103(0)-0.072(0)+0.007(0)+0.041(0)+0.058(1)+0.035(0)=3.013 \quad (Contrast)$$

$$2.852+0.031(1)+0.079(1)+0.035(1)+0.016(1)-0.023(0)+0.001(0)=3.013 \quad (Sequential)$$

$$2.955+0.121(\frac{1}{7})+0.107(\frac{1}{6})+0.036(\frac{1}{5})+0.001(\frac{1}{4})-0.022(-\frac{2}{3})+0.001(0)=3.013 \quad (Helmert)$$

The predicted score for "some college or equivalent" using dummy coding is the same as that for contrast, sequential, and Helmert coding. Following a similar procedure, it can be shown that all predicted scores match the category means for each coding strategy (Cohen, Cohen, West, & Aiken, 2003; Darlington & Hayes, 2016).

Since predicted scores are the same across coding strategies in linear regression, this means prediction accuracy is also the same across the different coding strategies. In our example data, prediction accuracy quantifies how far a model's predicted *stress* scores are from the observed *stress* scores of participants in the test data. We use Mean Squared

Error (MSE) to measure the prediction accuracy. Mathematically, MSE is calculated as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{3}$$

where n represents the number of observations in the test data; Y_i represents the observed outcome value of the i^{th} observation in the test data; and $\hat{Y}_i$ represents the predicted outcome value of the i^{th} observation from the model (which is generated using the training data). When we calculate the MSE of four linear regression models each fit using one of the four coding strategies mentioned previously, we find that all models have the exact same MSE of 0.13674. This illustrates that prediction accuracy is not affected by coding strategy when using linear regression.

While these results may seem trivial and require only a basic understanding of linear regression to understand, they stand in stark contrast to similar results we will examine in alternative regularized regression approaches. In summary, linear regression models with different coding strategies predict the same scores (i.e., category means) and give the same prediction accuracy, though they produce different coefficients. These properties persist when there are additional predictors (categorical and/or continuous) in the model, where the predicted scores (which are now *adjusted means*) are the same for all coding strategies, and thus prediction accuracy is always the same as well.

1.2 Lasso and Group Lasso Regression

In contrast to linear regression, lasso regression is useful when the proposed model involves many predictors, but only a few may be true predictors of the outcome (i.e., sparsity). Lasso is gaining popularity in behavioral science presumably because it shares many properties with linear regression, an already common statistical approach in the field (McNeish, 2015). For example, a lasso model fit to the COVID-19 data using *Education* to predict *Stress* would share the same equation as linear regression given in Equation 1. However, the values of the β_j coefficients would differ between the two methods because linear and lasso regression differ in the way they estimate the vector containing these regression coefficients, β. In linear regression, the estimated coefficient vector is calculated as follows,

$$\hat{\beta}_{linear} = \underset{\beta}{\operatorname{argmin}}(|Y - X\beta|_2^2), \tag{4}$$

where $|\cdot|_2$ is the notation for the L2 norm. Lasso, on the other hand, adds a penalty term governed by the penalty parameter λ to regulate the size of the coefficients:

$$\hat{\beta}_{lasso} = \underset{\beta}{\operatorname{argmin}}(|Y - X\beta|_2^2 + \lambda|\beta|_1), \tag{5}$$

where $|\cdot|_1$ is the notation for the L1 norm.[1] When λ is nonzero, nonzero values of β result in increases in $\lambda|\beta|_1$, and so Equation 5 reaches its minimum when both the

[1] Another alternative to lasso is ridge regression which is expressed by Equation 5 except with an L2 norm instead of an L1 norm for the regularization term. In Equation 5, the L1 norm

prediction error and the size of the elements of β are considered. A large λ value results in the coefficients in β being shrunk toward or equal to zero so fewer predictor variables are selected in the model (where "selected" means that the coefficient is nonzero in the final solution). A small λ value, on the other hand, results in less shrinkage so more predictor variables can be selected into the model. Linear regression is actually a special case of lasso regression when λ is set to zero.

While lasso has many benefits over linear regression (Hastie & Tibshirani, 2018; McNeish, 2015; Tibshirani, 1996), when applying lasso regression to models with categorical predictors, additional considerations must be made. Lasso regression models select variables based on the penalty parameter λ and the sizes of the entries in coefficient vector β. However, as we demonstrated with linear regression, using different coding strategies for a categorical predictor creates models with different coefficient vectors. This means that the choice of coding strategy may result in different variable selection in lasso regression models. The issue of coding strategies is related to the issue of variable scaling with continuous predictors, which also influences variable selection and prediction accuracy in lasso regression models. One common solution to this problem is to standardize all continuous predictors before applying lasso regression (Marquardt, 1980). In this way, the effect of scaling is excluded from the variable selection of lasso regression with continuous predictors. While dichotomous variables can be standardized, different coding strategies representing more than two categories do not result in the same standardized solution. Given this, there is reason to believe that the performance of lasso regression with categorical variables may be impacted by the choice of coding strategies for those variables.

A generalization of lasso regression which may also be impacted by coding strategy—but in different ways—is group lasso regression. Group lasso, as opposed to lasso, performs variable selection by selecting groups of variables rather than individual variables (Yuan & Lin, 2006). This is particularly valuable for the case of categorical predictors because the set of indicators for each variable forms a natural group. The mathematical formula for estimating the coefficient vector β in group lasso is

$$\hat{\beta}_{group} = \underset{\beta}{\operatorname{argmin}}(|Y - X\beta|_2^2 + \lambda \sum_{g=1}^{G} |\beta_{I_g}|_2) \tag{6}$$

where G represents the number of groups of variables, and β_{I_g} represents the coefficient vector of that corresponding group. Other notation is the same as Equation 5. Using the L2 norm within each group g is what allows group lasso to either select all or none of the variables within each group. Also, multiplying by λ after summing the L2 norms of all groups penalizes each group instead of each individual indicator variable. These differences provide group lasso with distinct properties: When all variables are considered one group, group lasso performs as ridge regression. On the other hand, when all the variables are their own group, group lasso performs as lasso regression.

penalizes the absolute value of the coefficients, used by lasso; while in ridge regression, the L2 norm penalizes the squares of all coefficients. Given this property, ridge regression is not as effective at penalizing parameters to zero compared to lasso regression (Tibshirani, 1996). Therefore, lasso regression is preferred for variable selection.

The advantage of group lasso is that when there are multiple groups of more than one variable, the result is a combination of within-group ridge regression and across-group lasso regression.

The group lasso has special properties with respect to variable selection. Within a group, group lasso typically includes or excludes all variables because of the within-group ridge regression. Given its unique properties with respect to variable selection, group lasso has been recommended as a useful alternative to lasso regression when dealing with models with categorical variables (Detmer, Cebral, & Slawski, 2020; McNeish, 2015); however, no prior research has explored the sensitivity of group lasso to different coding strategies. In group lasso, all indicators for a categorical variable are defined as a group, and the algorithm should either include all indicators associated with one categorical predictor or exclude all these indicators.

1.3 Motivation

With the increasing use of lasso techniques across scientific fields, but especially within the social and behavioral sciences, many researchers rely on their intuitions about the similarities between lasso and linear regression to understand, use, and interpret the results of lasso regression. This could be particularly problematic for models with categorical predictors. Prediction accuracy in linear regression is unaffected by the selection of coding strategy; however, lasso regression conducts regularization by minimizing regression coefficients, which differ across coding strategies. This may lead to different prediction accuracy and variable selection depending on the coding strategy used when using lasso. Since group lasso treats the variables in a group as a whole set, it seems less likely that its variable selection will be impacted by the choice of coding strategy. However, the prediction accuracy of group lasso may still be impacted by the coding strategy.

To explore the potential impacts of coding strategy on important characteristics of lasso and group lasso regression, we combine both real data analysis and simulation. First, using the COVID stress data set described previously, we demonstrate the use of lasso and group lasso regression with categorical variables, where different coding strategies of categorical variables impact two aspects of model performance: variable selection and prediction accuracy. Next, we use a Monte Carlo simulation to demonstrate a specific case where group lasso may tend to overfit the training data. In the last section, we explore other potential solutions, important future directions, and general conclusions.

2 Real Data Analysis with COVID Stress Data

We used the COVID stress data set with the same sample of 10,000 participants and the same training/test data sets used in Section 1.1 to explore how coding strategies affect models estimated by lasso and group lasso. In the models, we included six categorical predictors (where a predictor with k categories was represented by $k-1$ indicator variables): *Education* (7 categories), *Employment status* (6 categories), *Gender* (3 categories), *Isolation status* (4 categories), *Marital status* (4 categories), and *Mother's*

education (7 categories). We also included seven continuous predictors in the models. Thus, after coding all categorical variables and adding the seven continuous variables, the models predicted the outcome *Stress* with $6+5+2+3+3+6+7=32$ variables. In total, we trained eight different models using lasso and group lasso with four coding strategies: dummy, contrast, sequential, and Helmert. We used 10-fold cross-validation on the training data to select the penalty parameter from the model with the best prediction accuracy, so the penalty parameter that was selected is different across models with different coding strategies.[2] We then examined if the variable selection and prediction accuracy of these lasso and group lasso models were affected by the choice of coding strategy.

2.1 Variable Selection

We first examined differences in the variable selections of the four lasso models. Results are shown in Table 6. Focusing on the *Education* variable, we illustrate how the use of different coding strategies can result in conflicting findings. Both the dummy coding model and the sequential coding model have a predictor which represents the difference between no education and 6 years of education. After applying lasso, the dummy coding model includes this predictor, whereas the sequential coding model excludes this predictor. Based on these results, using the dummy coded model, a researcher might conclude that COVID stress differs across the no education and 6 years of education groups, whereas using a sequential coded model, the opposite conclusion would be made.

Fitting similar dummy-, contrast-, sequential-, and Helmert-coded models with group lasso, we found that the results differed notably from the traditional lasso. While lasso's variable selection was affected by the choice of coding strategy (see Table 6), the group lasso's variable selection seemed stable across different coding strategies, with all predictor variables selected to remain in all four models. Thus, based on the applied data analysis, it seems that variable selection is not impacted by the coding strategy for group lasso, though this should be subject to additional investigation. This suggests that if researchers are interested in using lasso for variable selection and have categorical predictors, using group lasso could avoid the arbitrary choice of coding strategy. However, group lasso was not successful in reducing the set of potential predictors, and thus, it may suffer from a limitation of being overly inclusive. We explore this issue more in a simulation.

2.2 Prediction Accuracy

In this section, we investigate whether prediction accuracy is affected by the choice of coding strategy using both lasso and group lasso. We examined the prediction accuracy in two ways: predicted category scores and MSE of the model applied to the test data set.

[2] Note that even with the same penalty parameter, models with different coding strategies or reference categories will still have different variable selection and prediction accuracy.

Table 6: Variable Selection for Different Coding Strategies by Lasso

	Lasso Regression			
Variable	*Dummy*	*Contrast*	*Sequential*	*Helmert*
Education	6 years - no	no - Average	**6 years - no**	no - Average(6 years and more)
	9 years - no	6 years - Average	9 years - 6 years	6 years - Average(9 years and more)
	12 years - no	**9 years - Average**	12 years - 9 years	**9 years - Average(12 years and more)**
	some college - no	12 years - Average	some college - 12 years	12 years - Average(some college, college, PhD)
	college - no	some college - Average	college - some college	some college - Average(college + PhD)
	PhD - no	**college - Average**	PhD - college	college - PhD
Employment Status	part-time - no	no - Average	part-time - no	no - Average(part-time, self-employed, student, full-time, retired)
	self-employed - no	**part-time - Average**	self-employed - part-time	**part-time - Average(self-employed, student, full-time, retired)**
	student - no	self-employed - Average	student - self-employed	self-employed - Average(student, full-time, retired)
	full-time - no	student - Average	full-time - student	student - Average(full-time, retired)
	retired - no	full-time - Average	retired - full-time	full-time - retired
Gender	man - woman	woman - Average	man - woman	woman - Average(man, other)
	other - woman	man - Average	other - man	man - other
Isolation Status	minor changes - usual	usual - Average	minor changes - usual	usual - Average(minor changes, isolated, medical isolated)
	isolated - usual	**minor changes - Average**	isolated - minor changes	minor changes - Average (isolated, medical isolated)
	medical isolated - usual	isolated - Average	medical isolated - isolated	isolated - medical isolated
Marital Status	divorced - single	single - Average	divorced - single	single - Average(divorced, married, other)
	married - single	divorced - Average	married - divorced	divorced - Average(married, other)
	other - single	married - Average	other - married	married - other
Mom's Education	6 years - no	no - Average	6 years - no	no - Average(6 years and more)
	9 years - no	6 years - Average	9 years - 6 years	6 years - Average(9 years and more)
	12 years - no	9 years - Average	12 years - 9 years	9 years - Average(12 years and more)
	some college - no	12 years - Average	some college - 12 years	12 years - Average(some college, college, PhD)
	college - no	some college - Average	college - some college	some college - Average(college, PhD)
	PhD - no	college - Average	PhD - college	college - PhD

Note. Variables with a white background color were selected to be in the model, and variables with a grey background color were not selected.

Predicted Category Scores We first examined whether the predicted *stress* score for each *Education* group is the same with different coding strategies in lasso and group lasso models. In this section, We generated the predicted score for each category using a model with only *Education* as a predictor, so the models contained 6 indicator variables in total. While this model is oversimplified, it eases the direct comparison between the true means of each group and the predicted scores.

The predicted category scores for lasso models fit using the four different coding strategies are shown in Table 7, with the final column providing the actual category means for *Education* observed in the training data. First off, it is important to note that category scores shown in the table were rounded. Thus, some category scores that were very close to the actual category scores were rounded to the same value, but there were no lasso models where the predicted scores were exactly equal to the group means like they would have been in linear regression. Also, it is evident in the table that the predicted means often differ depending on the coding strategy used. For five of the seven categories, the dummy-coded model estimated the category mean most accurately among all models.

Table 7: Predicted Category Scores for Different Coding Strategies by Lasso

	Dummy	*Contrast*	*Sequential*	*Helmert*	Training Mean	Test Mean
None	2.864	2.879	**2.864**	2.872	2.852	2.912
6 years	**2.883**	2.897	2.888	2.896	2.883	2.824
9 years	**2.962**	2.961	2.965	2.973	2.962	2.857
12 years	2.997	2.995	2.999	**2.997**	2.997	3.038
Some college	**3.013**	3.012	3.008	3.008	3.013	3.009
College	**2.990**	2.990	2.991	2.991	2.990	2.999
PhD/Doctorate	**2.991**	2.992	2.991	2.991	2.991	3.008

Note. Rows represent *Education* categories, and the middle four columns give the model predicted values with different coding strategies. The last two columns give the actual mean of each category observed in the training and test data, respectively. The closest value to the training mean is bolded and the closest value to the test mean has a grey background color in each row.

The results of the predicted category scores for the four group lasso models, shown in Table 8, are very similar to the lasso models: Group lasso estimated each category score within a categorical variable differently depending on the coding strategy used. Thus, although variable selection is not impacted by the coding strategy used for group lasso, the predicted category score *is* impacted by the choice of coding strategy. Also, among all group lasso models, the dummy-coded model generated the most accurate category scores for four of the seven categories. Thus, regardless of whether lasso or group lasso was used, the dummy-coded model estimated the majority of the category means better than the other three models. It is unclear whether this finding would remain true with other data sets, however.

The results in Table 7 and 8 show that different coding strategies result in different predicted category scores. While this is an important finding, it is equally important

Table 8: Predicted Category Means for Different Coding Strategies by Group Lasso

	Dummy	*Contrast*	*Sequential*	*Helmert*	Training Mean	Test Mean
None	2.828	**2.863**	2.879	2.868	2.852	2.912
6 years	**2.886**	2.892	2.906	2.894	2.883	2.824
9 years	2.965	**2.962**	2.954	2.963	2.962	2.857
12 years	**2.997**	2.996	2.994	2.996	2.997	3.038
Some college	**3.013**	3.012	3.011	3.012	3.013	3.009
College	**2.990**	2.990	2.991	2.990	2.990	2.999
PhD/Doctorate	2.991	2.991	**2.991**	2.990	2.991	3.008

Note. Same as Table 7.

to understand why this occurs and whether the degree of difference is predictable and understandable rather than random variability due to estimation. A core aspect of lasso and group lasso models is shrinkage: different coding strategies will result in different model intercepts and coefficients, because the degree of shrinkage is different across coding strategies.

To visualize the shrinkage effect of each coding strategy, we plotted the predicted scores from each lasso model along with each model's intercept in Figure 1. In the dummy-coded model, the predicted scores are all shrunk slightly toward the no education category score (since it is the intercept in this model) relative to the contrast-coded model, where the scores are instead all pulled closer to the grand mean (i.e., the model's intercept). The predicted scores from the sequential-coded and Helmert-coded models, on the other hand, are shrunk closer towards each other more than those from the dummy-coded or contrast-coded models, reflecting the fact that shrinkage in sequential coding and Helmert coding relies not on the intercept, but on the differences between neighboring categories or the average of multiple neighboring categories. For example, the 9 years and the some college categories are shrunk closer to the college category or Phd/Doctorate category in sequential-coded and Helmert-coded models. In summary, models fit with different coding strategies have different shrinkage patterns, and so predicted scores differ across these models, leading to different prediction accuracy. These results suggest that one way to select a coding strategy is to consider the pattern of shrinkage which seems most reasonable.

Model Fit Next, we recorded MSEs calculated from models including all six categorical variables and all seven continuous variables, to the test data set (Table 9). Model fit (MSE) differs by coding strategy for both lasso and group lasso. Contrast-coded models yielded the best MSE for both lasso and group lasso regression. This exposes uncertainty regarding which coding strategy should be used when lasso or group lasso regression is applied. While some differences in MSE are expected due to the stochastic nature of procedures like cross-validation used to choose the penalty parameter (λ), it is notable that the MSEs were more variable for the group lasso models than they were for the lasso models, suggesting that choice of coding strategy could result in a much less optimal model (possibly worse than linear regression) when using group lasso. We explore this issue more in the Monte Carlo simulation. In Appendix A, we

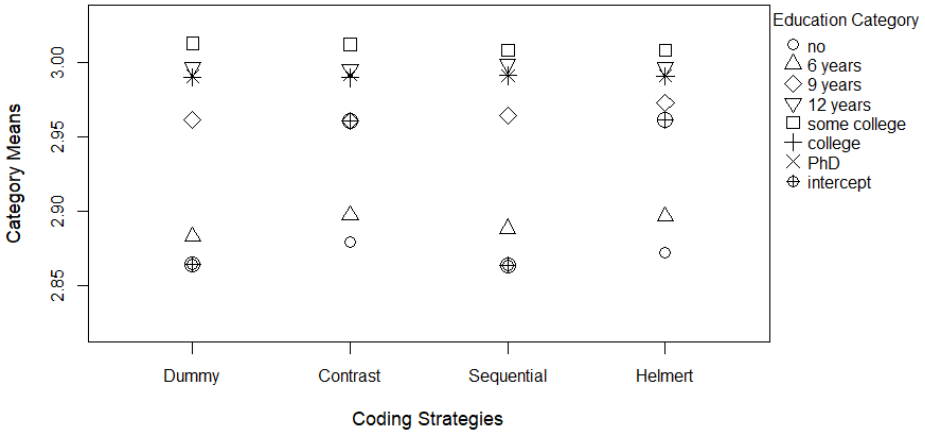

Figure 1: Graphical Presentation of Category Means for *Education* Recreated by Lasso Models with Different Coding Strategies. Intercept values are different across coding strategies. The intercept value is the estimated category mean for no education in dummy and sequential coding and the average of the category means in contrast and Helmert coding.

demonstrate similar issues with the choice of reference group or category order across different coding strategies, and in Appendix B we demonstrate that the use of singular design matrices (e.g., including dummy codes for all categories) does not ameliorate this issue.

Table 9: Model Fit (MSE) for Different Coding Strategies by Lasso and Group Lasso Regression

Coding strategies	*Dummy*	*Contrast*	*Sequential*	*Helmert*
Lasso Regression	0.13669	**0.13660**	0.13675	0.13677
Group Lasso Regression	0.13711	**0.13689**	0.13691	0.13695

Note. Rows represent different lasso methods, and columns represent models with different coding strategies. The lowest value (best prediction) in each row is bolded.

2.3 Summary

Choice of coding strategy has the potential to affect both variable selection and prediction accuracy in lasso regression models. As a result, depending on the coding strategy used, an analyst may end up with different variables included in their model, different predicted scores, and different prediction accuracy. With both the model's variable selection and predictive performance dependent on how categorical predictors are

represented in the model, it is not a choice that should be taken lightly. Ideally, there would be a method which provides the same variable selection and the same predicted scores regardless of the coding strategy chosen.

Group lasso partly addresses the issues caused by the choice of different coding strategies in lasso regression, because group lasso's variable selection is not affected by the coding strategy used. Therefore, if researchers use group lasso to select which variables contribute to the outcome variable, they do not need to worry that different coding strategies may result in different conclusions. However, coding strategies still affect the prediction accuracy of group lasso models. Therefore, if researchers aim to predict the outcome variable by using group lasso regression, they need to be aware that different coding strategies can result in different prediction accuracy. In addition, because group lasso is selecting more variables into the model, the robustness of group lasso across coding strategies may come at the cost of prediction accuracy. Comparing the MSEs between the lasso models and group lasso models, the lasso models typically have lower MSE (i.e., better prediction accuracy) than group lasso.

This trade-off between prediction accuracy and robustness leads to some additional concerns about the group lasso. There seems to be a trade-off between including a *set* of predictors in a model, as compared to when a specific predictor. For example, if the average stress for all levels of education was the same except for those with PhDs, would group lasso still select the education set of variables into the model? Will the set of indicators for the categorical variables be selected if there is only one category that differs from the other categories within that variable? If this group is selected into the model, this means that many additional parameters would also be included to capture an effect that is only attributable to one indicator variable. Alternatively, if the group is not selected, then the predictive ability of the group lasso model may suffer. This problem does not occur with lasso, as it is able to include a single indicator variable to represent one category differing from the rest. Next, we explore this specific case and examine if group lasso's ability to include groups of variables leads to issues with overfitting.

3 Monte Carlo Simulation

In this section, we use a Monte Carlo simulation to explore a potential weakness of group lasso: overfitting. Group lasso may select more variables than necessary into the model, leading to larger variance and lower prediction accuracy. We explore a particularly extreme data generation case, where across all categories within one categorical variable, only one category differs from the rest. We call this category the *dominant* category and refer to all others as *non-predictive* categories. A non-predictive category is always used as the reference category in the analysis. While the simulation is much simpler than cases that would occur in real data analysis, it provides a clear demonstration of a pattern that is likely to occur and be problematic and hard to identify in more complex situations.

Simulation Method The data was generated such that the dominant category had a nonzero category mean, while non-predictive categories all had category means of zero.

All categorical variables were encoded using dummy coding. A second predictor variable was generated to follow a standard normal distribution. The outcome variable was created by adding the category mean, the value of the continuous variable, and a random error term drawn from a standard normal distribution. For optimal prediction, both the continuous predictor and the indicator variable which estimates the difference between the dominant category and other non-predictive categories should be included in the model, while the variables associated with non-predictive categories should not.

As previously mentioned, the number of categories within categorical predictors may affect how the coefficients are estimated and how the model selects predictors in group lasso. Therefore, we varied the number of non-predictive categories (2,3,4). To examine how the effect size would affect group lasso's prediction accuracy and variable selection, we also simulated different dominant category means (0.1, 0.2, 0.3). For each combination of number of categories and effect size, we randomly generated 500 data sets with a sample size of 1200.

For each data set, we first split the data set into training and test sets randomly based on an 8:2 ratio. Then we fit lasso and group lasso models with the same training data. We selected the penalty parameter using the same cross-validation methods used in previous sections. For each model, we calculated the MSE, whether the model included the dominant category, and whether the model included the non-predictive categories. We calculated the average prediction accuracy of each method as well as the proportion of models that included the dominant category and the proportion that included non-predictive categories across each condition. For group lasso, these two proportions were always the same because group lasso either includes or excludes all categories within the categorical predictor.

Simulation Results We first found that in all conditions lasso had a higher prediction accuracy than group lasso, indicated by lower MSEs (Table 10). Though the differences in MSE of lasso and group lasso were small, they were consistent across different conditions. Secondly, for both group lasso and lasso regression, when the number of non-predictive categories increased, the probability for models to include the dominant category decreased, but the probability for lasso was consistently greater than or equal to that for group lasso (Figure 3). This means that lasso is more likely to include the dominant category than group lasso across the number of non-predictive groups. Figure 2 shows that when the number of non-predictive categories stayed the same, the probability for group lasso to include non-predictive categories increased when the effect size increased, while the probability for lasso remained relatively flat. For both models, the probability of including non-predictive categories decreased as the number of non-predictive categories increased.

Returning to the potential issue of overfitting in group lasso, consider the case where the dominant group mean is large. Figure 2 shows that when the dominant group mean was 0.3, group lasso had a higher probability than lasso of including non-predictive categories. In this case, group lasso could overfit the data because group lasso was more likely to include categories that were not supposed to be in the model. This also explains group lasso's lower prediction accuracy than lasso in Table 10 when the dominant group mean was large.

Table 10: Differences in MSE of Lasso and Group Lasso Models for Monte Carlo Simulation

	Number of Non-predictive Categories		
Dominant Category Mean	2	3	4
0.1	0.0028	0.0029	0.0003
0.2	0.0020	0.004	0.0008
0.3	0.0029	0.0029	0.0030

Note. Values larger than zero mean that the MSE for group lasso is larger than the MSE for lasso.

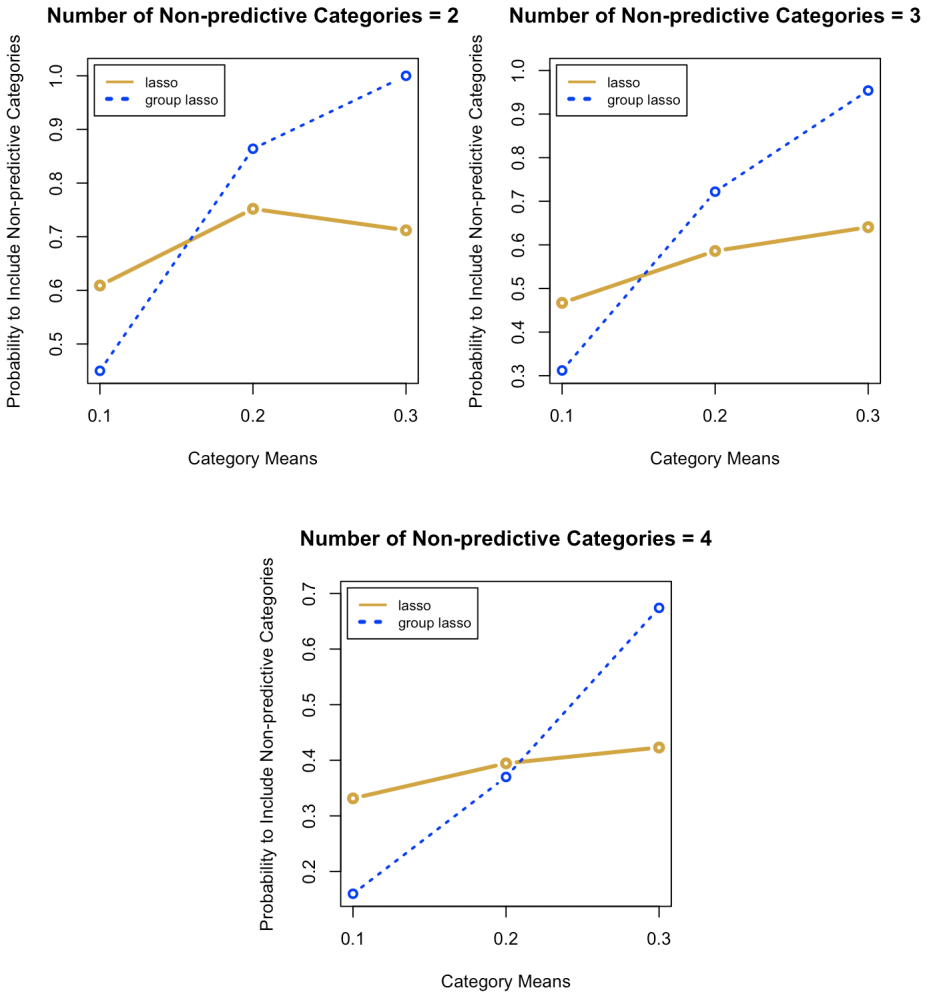

Figure 2: Comparison of Probabilities of Including Non-Predictive Categories under Different Numbers of Categories for Lasso and Group Lasso Models

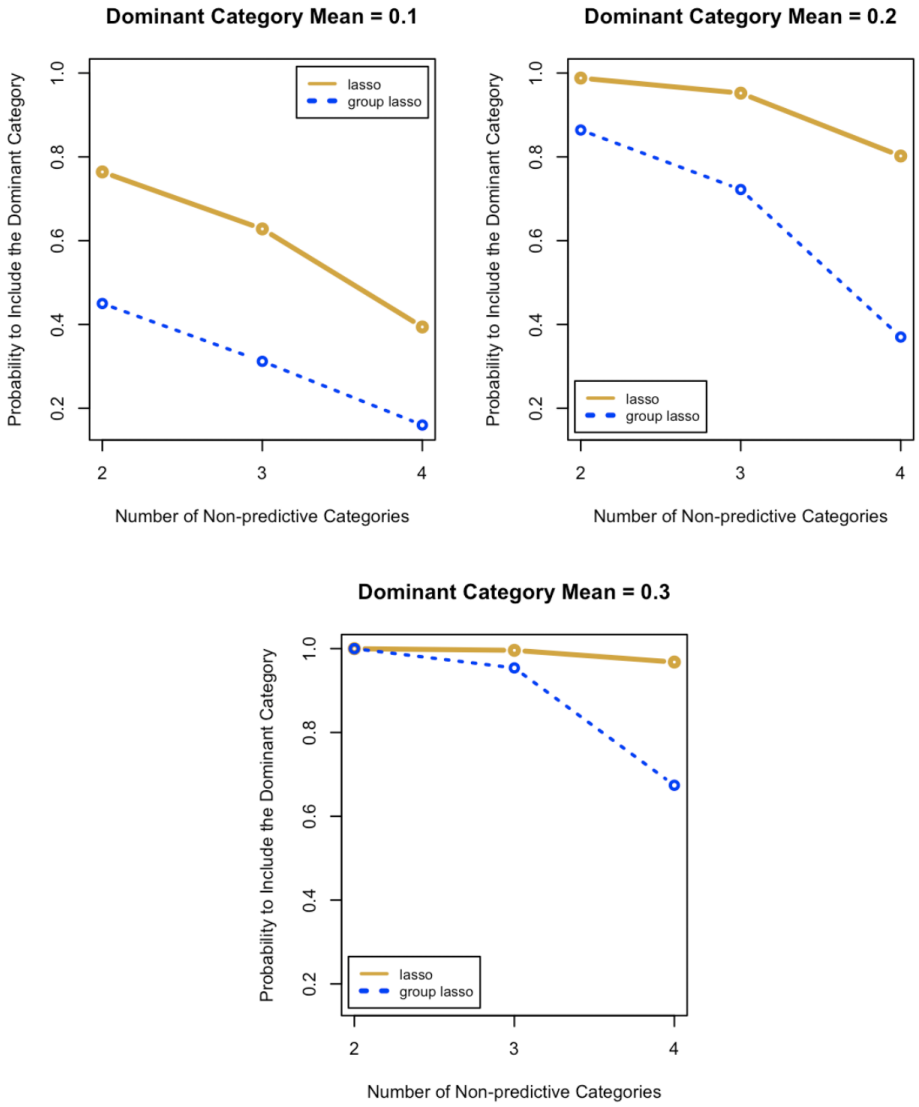

Figure 3: Comparison of Probabilities of Including the Dominant Category under Different Dominant Category Means for Lasso and Group Lasso Models

Simulation Summary Using Monte Carlo simulation, we demonstrated conditions under which group lasso may be likely to have issues with overfitting. When one or just a few categories differ from the rest, lasso may be more efficient with better prediction accuracy than group lasso. In these cases, group lasso is likely to include the categorical variable, including all non-predictive categories. Therefore, if researchers use group lasso to build predictive models, they may want to examine if one or two categories have relatively dominant means within categorical variables in advance, or if this pattern is hypothesized to occur they might prefer lasso. Looking for these effects may be particularly difficult in cases with many predictors where limited theoretical knowledge are driving the modeling, which is often the case when lasso is used. The differences must be conditional on all other variables in the data, not just examining the group means. If there are many categorical predictors in the model, exploratory analyses could be undertaken for each categorical variable, but this could a be tedious undertaking. Overall, this simulation demonstrates that there may be situations in which group lasso is not optimal for handling categorical predictors, especially if prediction accuracy is a high priority.

4 Discussion

In this paper, we demonstrate that lasso and group lasso models are sensitive to decisions about coding strategy for categorical predictors (e.g., dummy or sequential) and the choice of reference group/order of the categories (Appendix A). Linear regression does not have this problem, as the model fit and predicted values do not vary depending on the coding strategy. Group lasso presents a partial solution by having consistent variable selection across coding strategies. However, this consistency may come at a cost of reduced prediction accuracy. Ultimately, this leaves open the question of which coding strategy should be chosen. In the next section, we explore potential solutions to this issue with categorical predictors in lasso-based models.

4.1 Exploring Potential Solutions

Regardless of which of the following solutions researchers choose, one thing is always required: transparency. In searching the literature for examples of applications of lasso with categorical predictors, we found very few teams reported the coding strategy or order of categories used. Researchers using categorical variables in lasso or group lasso regression need to report how they coded the variables (both coding strategy and variable order/reference group) as this is imperative for reproducing or replicating their results. The following are a few proposed solutions, none of which seem satisfactory for all cases. As such, we weigh the pros and cons of each and consider cases when each approach might be most acceptable.

Prioritize Interpretability In cases where one coding strategy provides better interpretability of the model coefficients than another strategy, the most interpretable coding strategy could be chosen. This comes at the risk of having a worse predictive model, since the idea of interpretability is still very much rooted in the origins of inferential

rather than predictive statistical models. In particular, because the coefficient estimates in lasso regression are biased, they should not be interpreted directly. Rather, after variable selection is completed, common recommendations are to fit a linear regression model that only includes the selected variables (Hastie, Robert, & Wainwright, 2015). It would be unusual to include a coding strategy in the follow-up linear regression that is different from the strategy used in the lasso regression. Thus, researchers should choose the coding strategy for each categorical variable that would be most interpretable if that variable was selected by a variable selection procedure to remain in the model. Coding schemes like Helmert coding require the presence of all predictors to have the intended interpretation, and should perhaps only be used in concert with group lasso (ensuring all predictors are selected in or out of the model) if interpretability is the top priority. Notably, machine learning approaches are often used in cases where there are many variables included in the analysis, and relatively little theory regarding which variables should be predicting the outcome. This could make it difficult for the researcher (or analyst) to decide which coding scheme would be "most interpretable," especially considering the many possible combinations of coding schemes and variable orders/reference groups.

Prioritize Robust Variable Selection Based on the real data analysis and the simulation results, the group lasso is robust to coding strategy choices with respect to variable selection. Prediction accuracy is not necessarily optimized for the group lasso. However, when the goal is to select variables, and especially when it is conceptually useful to keep or drop all indicators for each categorical variable, group lasso seems to be an optimal choice. Nevertheless, this may come at a cost of prediction accuracy, particularly if categorical variables follow the dominant group pattern explored in the Monte Carlo simulation above, where one group is distinct from all other groups.

Prioritize Prediction Another option when estimating lasso or group lasso models would be to try many different coding strategies in order to select the one with the best overall prediction accuracy. This process should likely be completed using the training data so it does not influence the final prediction accuracy estimate acquired using an independent sample of the data. This approach can be very computationally intensive. With multiple categorical variables in the data set, trying different combinations of coding strategies would result in maximized prediction accuracy.

Notably, if prediction accuracy is of the highest priority, alternative machine learning approaches typically have higher prediction accuracy than lasso approaches, and many are robust to coding strategy. Techniques like classification and regression trees (CART) are unaffected by coding strategy because categorical predictors are treated as a single variable (Finch & Schneider, 2007). Realistically, researchers may be balancing their comfort with advanced analytic methods and their priority of prediction accuracy. CART methods do not provide the "regression-like" estimates which many behavioral scientists rely on for interpreting their results.

4.2 Future Directions

There are several future directions we believe would be particularly beneficial for improving the state of research in the area of (group) lasso regression with categorical predictors. The first is the concept of intercept penalization. The typical practice within lasso is not to penalize the intercept (Wu & Lange, 2008), but the interpretation of the intercept varies greatly depending on which coding scheme is used. For example, when dummy coding is used, the intercept is the average of the reference group. Alternatively, when contrast coding is used, the intercept is the average of all groups. Ultimately, this means that different group means have differential penalization depending on the coding strategy used (as reflected in Figure 1). Thus, it is worth investigating whether penalizing the intercept may be appropriate in certain cases, and whether this would improve prediction accuracy (just as penalizing all other regression coefficients improves prediction accuracy in lasso). This question remains largely unexplored and would be informative to researchers who are interested in improving prediction accuracy.

Current defaults in software suggest that the field norm for coding strategy is dummy coding. The current research has demonstrated that dummy coding is a potentially risky choice as a default, as the choice of reference group can greatly impact the model, and the shrinkage is toward a group mean. Alternatively, contrast coding may make it an appealing default for researchers unsure about which coding strategy to use. Because the interpretation of the intercept for contrast coding is the average across all groups, the penalization of the groups is symmetric about this average. This means that when a coefficient is dropped from the model, the group that is indicated by this predictor is assumed to be equal to the grand mean. This method contrasts with dummy coding where all estimated group means are shrunk towards the reference group score. As a result, the selection of the reference group in contrast coding has less of an impact on parameter estimates than it does in dummy coding, because by selecting a reference group in dummy coding, that group's score is not at all penalized (if the intercept is not penalized). The interpretation of the intercept from contrast coding also aligns with how intercepts would be interpreted if there were no categorical variables in the model and all continuous predictors were standardized (i.e., sample average). Thus, contrast coding stands as a reasonable default if researchers are unsure of which coding strategy to choose; however, the use of contrast coding should be studied further in a variety of contexts to assess its appropriateness as a potential default.

Another observation our team made during this investigation was that group size mattered quite a lot with respect to how much predicted group scores varied across different coding strategies. In particular, in the COVID stress data, the no education group was particularly small ($N = 77$ out of 10,000 observations). This resulted in two problems that merit further investigation. The first is how group size can impact estimates and interact with the selection of coding strategy/reference group. Previous research by Choi, Park, and Seo (2012) has already shown that variability in the number of groups that categorical predictors contain can influence whether lasso or group lasso produces better prediction accuracy and recovery of model coefficients. As can be seen in Figure 1 and Table 7, the estimated means for the no education group in the COVID stress data were very unstable and varied more across coding strategies than any other group. Similarly, in Table 12 in Appendix A, we can see that the estimates of all of the *Education*

group means have the greatest bias when no education is used as the reference group. Future research should examine how variability in the sizes of those groups can impact the fitting of lasso and group lasso models

A second issue brought up by having small groups is the difficulty of splitting test and training data sets. This may become particularly problematic when there are many categorical variables that include many groups. Previous researchers have resolved to combine groups that are particularly small (e.g., racial/ethnic minorities; Webb et al., 2019). It is unclear how this practice impacts estimates for these groups, however, and in general combining groups is actively discouraged for other analytic methods (Tarantola & Dellaportas, 2005). Methods for splitting the data such as block randomization may provide more accurate predictions for small groups if the groups can be evenly split across the training and test sets.

4.3 Conclusion

Overall, our findings suggest that researchers should be cautious and purposeful about selecting their coding strategies when using lasso or group lasso. These choices will impact both variable selection and prediction accuracy when using lasso and prediction accuracy when using group lasso. However, just because variable selection is not impacted in group lasso does not mean this method should always be preferred. In a simulation study, we demonstrated cases where group lasso may have lower prediction accuracy than lasso, particularly when there is a dominant group (one group that differs from all other groups). The choices of which method to use (lasso or group lasso), what coding strategy to use, and which group order/reference category to use should depend on the researcher's priorities. How categorical variables are represented in lasso or group lasso models must be transparently reported to maximize reproducibility and replicability. Future research should explore specific practices in this area such as penalization of the intercept, the use of contrast coding, and how small groups should be accounted for to optimize prediction accuracy for these groups.

Behavioral scientists are quickly adopting useful tools developed in statistics and computer science which fit under the broad area of machine learning and artificial intelligence. The use of these tools will likely improve the ability of behavioral researchers to predict out-of-sample data, which may be particularly important in clinical settings and precision medicine. However, it is important to acknowledge that these new tools do not necessarily perform in the same ways that many researchers expect based on their training, which is primarily in linear regression and ANOVA frameworks (Aiken, West, & Millsap, 2008). Ensuring that the differences between these more traditional statistical frameworks and the newly developed machine learning frameworks are clearly defined will improve the implementation of these new methods throughout the field of behavioral science.

References

Aiken, L. S., West, S. G., & Millsap, R. E. (2008). Doctoral training in statistics, measurement, and methodology in psychology: replication and extension

of aiken, west, sechrest, and reno's (1990) survey of phd programs in north america. *American Psychologist*, *63*(1), 32 – 50. doi: https://doi.org/10.1037/0003-066X.63.1.32

Choi, Y., Park, R., & Seo, M. (2012). *Lasso on categorical data.* Retrieved from http://cs229.stanford.edu/proj2012/ChoiParkSeo-LassoInCategoricalData.pdf

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/-correlation analysis for the behavioral sciences* (3rd ed.). Mahwah, NJ: Erlbaum. doi: https://doi.org/10.4324/9781410606266

Darlington, R., & Hayes, A. (2016). *Regression and linear models: Concepts, applications, and implementation*. New York: Guilford Press.

Detmer, F. J., Cebral, J., & Slawski, M. (2020). A note on coding and standardization of categorical variables in (sparse) group lasso regression. *Journal of Statistical Planning and Inference*, *206*, 1–11. doi: https://doi.org/10.1016/j.jspi.2019.08.003

Finch, H., & Schneider, M. K. (2007). Classification accuracy of neural networks vs. discriminant analysis, logistic regression, and classification and regression trees. *Methodology*, *3*(2), 47-57. doi: https://doi.org/10.1027/1614-2241.3.2.47

Hastie, T., Robert, T., & Wainwright, M. (2015). *Statistical learning with sparsity: The lasso and generalizations*. CRC Press. doi: https://doi.org/10.1201/b18401

Hastie, T., & Tibshirani, R. (2018). Best subset, forward stepwise, or lasso? analysis and recommendations based on extensive comparisons.. doi: https://doi.org/10.1214/19-sts733

Marquardt, D. W. (1980). Comment: You should standardize the predictor variables in your regression models. *Journal of the American Statistical Association*, *75*(369), 87-91. doi: https://doi.org/10.1080/01621459.1980.10477430

McNeish, D. (2015). Using lasso for predictor selection and to assuage overfitting: A method long overlooked in behavioral sciences. *Multivariate Behavioral Research*, *50*(5), 471 – 484. doi: https://doi.org/10.1080/00273171.2015.1036965

StataCorp. (2019). *Stata statistical software: Release 16.* College Station, TX: StataCorp LLC.

Tarantola, C., & Dellaportas, P. (2005). Model determination for categorical data with factor level merging. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, *67(2)*, 269 - 283. doi: https://doi.org/10.1111/j.1467-9868.2005.00501.x

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, *58*(1), 267 – 288. doi: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Webb, C. A., Trivedi, M. H., Cohen, Z. D., Dillon, D. G., Fournier, J. C., Goer, F., … Pizzagalli, D. A. (2019). Personalized prediction of antidepressant v. placebo response: evidence from the embarc study. *Psychological Medicine*, *49*(07), 1118-1127. doi: https://doi.org/10.1017/s0033291718001708

Wu, T. T., & Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, *1*, 224-244. doi: https://doi.org/10.1214/07-aoas147

Yamada, Y., Ćepulić, D.-B., Coll-Martín, T., Debove, S., Gautreau, G., Han, H., ... Lieberoth, A. (2021, 1). Covidistress global survey dataset on psychological and behavioural consequences of the covid-19 outbreak. *Scientific Data*, *8*(3). doi: https://doi.org/10.1038/s41597-020-00784-9

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, *68*, 49-67. doi: https://doi.org/10.1111/j.1467-9868.2005.00532.x

Appendix A Different reference categories

In addition to the analyses presented in the primary manuscript, we also examined how variable selection and prediction accuracy in lasso and group lasso models differ across choices within a specific coding strategy. These choices include reference categories (dummy and contrast coding) and the order of categories (sequential and Helmert coding). We tested whether the category chosen as the reference category in the dummy coding strategy matters for variable selection and prediction accuracy. Consider, for example, the dominant group case where all groups have the same mean except one group. If that one group is selected as the reference category, then all $k-1$ predictors should be selected into the model, because all other groups are different from the reference. If any other group is selected as the reference group, then only 1 predictor should be selected into the model (the indicator for the difference between the one deviant group and the reference). While the pattern of means is not different, the reference group may have a large impact on the size of the coefficients and the number of non-zero coefficients.

We fit lasso and group lasso models with all six dummy-coded categorical variables and seven continuous variables using the COVID stress data. To explore how choices of reference categories affect estimated coefficients, we fit seven models for each regression method with differences only in their choices of reference categories in the variable *Education*. The reference categories were chosen and fixed for all other categorical variables. Therefore, the differences between these models can only be attributed to the different choices of the reference category of the variable *Education*. While this example uses dummy coding, we believe the results would generalize to other coding strategies (e.g., choice of the reference group for contrast coding, order of groups for Helmert and sequential coding).

Appendix A.1 Variable Selection

Table 11 shows the coefficients of indicators for *Education*. The size of the coefficients varies depending on which group is the reference, which could pose a problem for lasso regression because coefficients and the penalty parameter decide whether the variable will be selected into the model, according to Equation 5. Different coefficients are not necessarily a problem by themselves; however, these results demonstrate certain asymmetries that are concerning. When coefficients vary from model to model, the variable selection can differ. For example, when "none" was the reference category, the college category was not selected into the model (i.e., the none and college categories are assumed to be equal). However, when "college" was chosen as the reference

category, the none category *was* selected into the model (i.e., the none and college categories are treated differently). This marks a particularly concerning lack of symmetry between these lasso models.

Table 11: Model Coefficients for Different Reference Categories by Lasso

	Reference Category						
Variables	*None*	*6 years*	*9 years*	*12 years*	*Some college*	*College*	*PhD/Doctorate*
Intercept	2.637	2.574	2.649	2.668	2.657	2.641	2.649
None	.	-0.013	-0.076	-0.092	-0.083	-0.068	-0.075
6 years	-0.068	.	-0.079	-0.095	-0.086	-0.071	-0.078
9 years	0.010	0.065	.	-0.006	0	0.009	0.002
12 years	0.033	0.084	0.024	.	0.017	0.032	0.024
Some college	0.017	0.067	0.008	-0.006	.	0.016	0.008
College	0	0.049	-0.008	-0.024	-0.015	.	-0.008
PhD/Doctorate	0.005	0.057	0	-0.015	-0.006	0.005	.

Note. Each column represents one model, and each row represents the coefficients for *Education* produced by each model."." is the reference category for the corresponding model, and 0 means that lasso does not select the corresponding predictor to be included in the model.

Group lasso models included all categories within the variable *Education* when different categories were chosen as the reference categories, meaning that all categories were treated as different in all group lasso models. Group lasso ensures stable performance of variable selection across reference categories.

We also explored the effect of different reference categories in education on other predictors and found that choosing different reference categories affects the coefficients and variable selection of other predictors (categorical and continuous) in lasso models. Group lasso models, on the other hand, still performed consistent variable selection for predictors that did not have their reference categories changed. In our case, group lasso models always included all categories within the other five categorical predictors and all seven continuous predictors.

Appendix A.2 Prediction Accuracy

We examined the prediction accuracy from two aspects: predicted category scores and model fit, varying the reference group used in dummy coding education.

Predicted Category Scores Predicted values for each category were different in both lasso and group lasso models from Tables 12 and 13. For the no education category, lasso models with different reference categories predicted different values, ranging from 2.982 to 2.915. Group lasso models also predicted different values for the no education category, ranging from 2.983 to 2.991. This indicates that with different choices of reference categories, predicted values vary from model to model for both lasso and group lasso.

Table 12: Predicted Category Means and Prediction Accuracy for Different Reference Categories by Lasso

	Reference Category						
Category	*None*	*6 years*	*9 years*	*12 years*	*Some college*	*College*	*PhD/Doctorate*
None	2.982	2.920	2.915	2.915	2.915	2.915	2.915
6 years	2.914	2.933	2.912	2.912	2.912	2.912	2.912
9 years	2.992	2.998	2.990	3.001	2.998	2.992	2.992
12 years	3.015	3.017	3.014	3.006	3.014	3.014	3.014
Some college	2.999	3.001	2.998	3.000	2.998	2.998	2.998
College	2.982	2.983	2.982	2.982	2.982	2.983	2.982
PhD/Doctorate	2.988	2.990	2.990	2.991	2.991	2.987	2.990
MSE	0.13669	0.13678	0.13674	0.13684	0.13675	0.13674	0.13674

Note. Each column represents one model, and each row (besides the last) represents the predicted category means for *Education* produced by each model (with all continuous predictors set to their means and all other categorical variables set to their modes). The last row contains the MSE of the corresponding model.

Table 13: Predicted Category Means and Prediction Accuracy for Different Reference Categories by Group Lasso

	Reference Category						
Category	*None*	*6 years*	*9 years*	*12 years*	*Some college*	*College*	*PhD/Doctorate*
None	2.986	2.986	2.986	2.990	2.991	2.983	2.987
6 years	2.971	2.978	2.974	2.983	2.981	2.971	2.975
9 years	2.988	2.987	2.968	2.979	2.975	2.965	2.969
12 years	3.002	3.001	3.004	2.991	2.992	2.985	2.988
Some college	2.996	2.996	2.997	2.996	3.004	3.003	3.004
College	2.983	2.983	2.983	2.983	2.983	2.996	2.997
PhD/Doctorate	2.988	2.988	2.988	2.990	2.990	2.987	2.983
MSE	0.13711	0.13719	0.13709	0.13727	0.13709	0.13708	0.13710

Note. Same as Table 12

Figure 4 visualizes the shrinkage effect when different reference categories were chosen in lasso models using *Education* to predict *Stress*. In this case, the intercept is the predicted category mean of each model's reference category because models are coded by dummy coding strategies. Similar to Figure 1, we can conclude that recreated category scores shrink towards the reference value for dummy coding.

Model Fit Model fit, measured by MSE, for both lasso and group lasso models are shown in Table 12 and 13. MSEs were generally different across reference categories. Note that MSEs in Table 12 and 13 were rounded. Although some MSEs were very close to each other and were rounded to the same value, they were not exactly the same, which would be the case if linear regression was used.

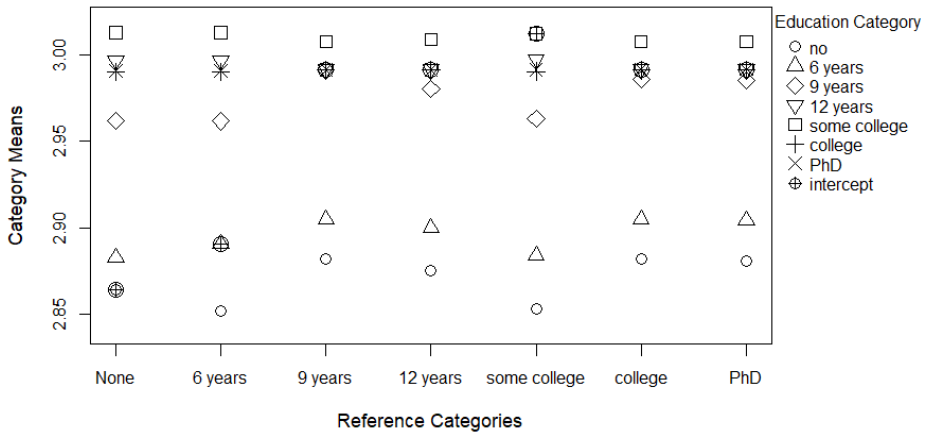

Figure 4: Graphical Presentation of Category Means for *Education* Recreated by Lasso Models with Different Reference Categories. Intercept values are different across reference categories. In dummy coding, the intercept value is the estimated category mean of the corresponding reference category.

Appendix B Singular Design Matrices

STATA is a commonly used statistical software that can implement lasso regression, and in STATA categorical predictors are handled by including a singular design matrix StataCorp (2019). In this section, we examine this alternative method for creating the design matrices for categorical variables. When we introduced categorical variables, we noted that for a variable with k categories, $k-1$ indicators are created for this variable. Different coding strategies use different matrices to represent the $k-1$ indicators and model coefficients represent differences between categories and the reference value, as this is common practice for linear regression. The researcher must then choose the reference category for analysis. However, there is another way to create the design matrix for categorical predictors where the researcher does not need to explicitly choose the reference category. Instead of using $k-1$ indicators for a categorical variable with k categories, we use k indicators. This design matrix allows lasso or group lasso to essentially select the reference values. Mathematically, this type of design matrix is defined as singular, because the matrix is not invertible. Singular design matrices cannot be used for linear regression, but lasso and group lasso regression can accommodate singular design matrices, making this a unique potential solution to the variable selection and prediction accuracy issue related to categorical variables in lasso and group lasso.

Can singular design matrices solve the inconsistency in lasso's variable selection and prediction accuracy or group lasso's prediction accuracy across coding strategies? To create singular design matrices, we appended a linearly independent column with only 1 in the first row to the matrices in Table 1 and 3, and a linearly independent column with only 1 in the last row to matrices in Table 2 and 4. If using singular design

matrices solves the issues of variable selection and prediction accuracy, these two properties should be equivalent across these four design matrices. To test this, we used the same data set and applied the same process as before to fit lasso and group lasso models with *Education* serving as the only predictor variable. Table 14 shows the coefficients of the categorical variable *Education* in lasso models as an example of lasso's variable selection. Using a singular design matrix for categorical variables, different coding strategies still lead to different lasso model's variable selection. Contrastingly, group lasso selected all categories and performed the same variable selection. For example, the contrast-coded lasso model treated the 9 years of education and PhD categories as the same, while these two categories were always treated as different in the other three lasso models and the four group lasso models. In addition, lasso and group lasso models using different coding strategies led to different prediction accuracies, shown in Table 15 and 16. This means that using singular design matrices does not solve the inconsistent variable selection or prediction accuracy for lasso, nor does it solve the inconsistency in prediction accuracy for group lasso. There are infinitely many singular design matrices that could be used, and if they all result in different solutions, this does not provide strong evidence that the identity matrix system used by StataCorp (2019) would perform optimally.

Table 14: Model Coefficients Using Singular Design Matrix with Lasso

Coding strategies	*Dummy*	*Contrast*	*Sequential*	*Helmert*
Intercept	2.991	2.961	2.873	2.961
1. no	-0.109	-0.081	0.015	0.103
2. 6 years	-0.086	-0.063	0.076	0.095
3. 9 years	-0.006	0	0.034	0.023
4. 12 years	0	0.034	0.010	0
5. some college	0.016	0.051	-0.017	-0.017
6. college degree	0	0.028	0	0
7. PhD	0	0	-0.009	0

Note. Each column represents one model, and each row represents the coefficient for an indicator of *Education* produced by the corresponding model. A 0 means that lasso does not select the corresponding category into the model.

Table 15: Predicted Category Means and Prediction Accuracy for Different Coding Strategies using Singular Design Matrices with Lasso

	Coding Strategy				
Category	*Dummy*	*Contrast*	*Sequential*	*Helmert*	Observed Mean
None	2.882	2.881	2.864	2.873	2.852
6 years	2.905	2.898	2.888	2.897	2.883
9 years	2.986	2.961	2.964	2.973	2.962
12 years	2.991	2.995	2.999	2.997	2.997
Some college	3.007	3.012	3.008	3.008	3.013
College	2.991	2.990	2.991	2.991	2.990
PhD/Doctorate	2.991	2.992	2.991	2.991	2.991
MSE	0.15630	0.15620	0.15616	0.15621	/

Note. Each column (besides the last) represents one model, and each row (besides the last) represents the predicted category means for *Education* produced by each model. The last column contains the category means observed in the training data set. The last row contains the MSE of the corresponding model.

Table 16: Predicted Category Means and Prediction Accuracy for Different Coding Strategies Using Singular Design Matrices with Group Lasso

	Coding Strategy				
Category	*Dummy*	*Contrast*	*Sequential*	*Helmert*	Observed Mean
None	2.873	2.869	2.878	2.871	2.852
6 years	2.892	2.890	2.909	2.891	2.883
9 years	2.962	2.961	2.955	2.962	2.962
12 years	2.996	2.996	2.994	2.996	2.997
Some college	3.012	3.012	3.010	3.012	3.013
College	2.990	2.990	2.991	2.990	2.990
PhD/Doctorate	2.990	2.991	2.991	2.990	2.991
MSE	0.15619	0.15619	0.15622	0.15619	/

Note. Same as Table 15

Journal of Behavioral Data Science, 2023, *3* (2), 43–63.
DOI: https://doi.org/10.35566/jbds/v3n1/li

Robust Bayesian growth curve modeling: A tutorial using JAGS

Ruoxuan Li

University of Notre Dame, Notre Dame, USA
rli23@nd.edu

Abstract. Latent growth curve models (LGCM) are widely used in longitudinal data analysis, and robust methods can be used to model error distributions for non-normal data. This tutorial introduces how to model linear, non-linear, and quadratic growth curve models under the Bayesian framework and uses examples to illustrate how to model errors using t, exponential power, and skew-normal distributions. The code of `JAGS` models is provided and implemented by the R package `runjags`. Model diagnostics and comparisons are briefly discussed.

Keywords: Robust Growth Curve Modeling · Bayesian Estimation · Structural Equation Modeling · JAGS

1 Introduction

Latent growth curve models (LGCM) are widely used in longitudinal studies, and LGCM performs well in the identification of intraindividual changes and investigation of interindividual differences in intraindividual changes (McArdle & Nesselroade, 2014). LGCM can estimate linear and nonlinear growth trajectories flexibly or freely estimate the shape of growth trajectory by observed data. Researchers may employ either the maximum likelihood estimation method or the Bayesian method to model LGSM. The Bayesian methods have advantages on handling difficulties in longitudinal data such as unequally spaced measurements, nonlinear trajectories, non-normally distributed data, and small sample sizes (Curran, Obeidat, & Losardo, 2010).

Influential outliers and non-normally distributed data can lead unreliable estimates and inferences. Conventional methods such as deleting outliers may result in underestimated standard errors (Lange, Little, & Taylor, 1989). Robust statistical modeling methods have been developed to handle the violation of the normality assumption. For example, the t-distribution is more robust to outliers, and using t-distribution to model errors is one of the robust modeling strategies (Lange et al., 1989). Robust modeling using t-distributions is easy to understand and applied in both maximum likelihood and Bayesian methods (Lange et al.,

1989; Zhang, 2016; Zhang, Lai, Lu, & Tong, 2013). The degree of freedom of t-distributions can be estimated or predetermined, and a large degree of freedom means the t-distribution approaches a normal distribution (Zhang et al., 2013). Based on simulation studies, the robust method using the t-distributions for the error term demonstrates good performance for heavy-tailed data in growth curve models, and it efficiently estimates the standard error (Zhang, 2016; Zhang et al., 2013).

This tutorial aims to present how to implement robust Bayesian growth curve models using R and the JAGS programs. To begin, it provides a brief introduction to LGCM, including the latent basis growth curve models (LBGM), the linear growth curve models, and quadratic growth curve models. Then it introduces commonly used priors and convergence diagnostic methods. Finally, a real data set is used to demonstrate how to implement robust LGCM, and how to interpret the estimated parameters.

2 Models and notations

2.1 General latent growth curve models

Latent basis growth curve models A LGCM with one variable Y can be written as:

$$\mathbf{Y}_i = \boldsymbol{\tau} + \boldsymbol{\Lambda}\boldsymbol{b}_i + \boldsymbol{\epsilon}_i \tag{1}$$

$$\boldsymbol{b}_i = \boldsymbol{\beta} + \boldsymbol{u}_i \tag{2}$$

$\mathbf{Y}_i$ is a $T \times 1$ vector in which T is the total number of measurement occasions, and $\boldsymbol{\Lambda}$ is a $T \times q$ factor loading matrix, and it decides the shape of the growth trajectory. The $\boldsymbol{\epsilon}_i$ is assumed to follow a q-variate normal distribution $\boldsymbol{\epsilon}_i \sim \text{MN}(0, \boldsymbol{\Phi})$. $\boldsymbol{b}_i$ is an $q \times 1$ vector and it represents the latent variables used to describe the change. $\boldsymbol{\beta}$ is a $q \times 1$ vector that represents the fixed effect (the means of $\boldsymbol{b}_i$) and $\boldsymbol{e}_i$ is the individual deviation from the fixed effect $\boldsymbol{\beta}$. $\boldsymbol{u}_i$ follows a multivariate normal distribution with q dimensions as $\boldsymbol{u}_i \sim \text{MN}(0, \boldsymbol{\Psi})$.

LBGM is a special case of the general LGCM. It assumes the error variance is the same for all measurements (homogeneity) by simplifying $\boldsymbol{\Phi} = \mathbf{I}\sigma_e^2$. And it also assumes measurement errors are uncorrelated. The parameters in LBGM are:

$$\boldsymbol{\Lambda} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & \lambda_1 \\ \vdots & \vdots \\ 1 & \lambda_{T-2} \end{pmatrix} \qquad \boldsymbol{b}_i = \begin{pmatrix} b_{iL} \\ b_{iS} \end{pmatrix},$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_L \\ \beta_S \end{pmatrix} \qquad \boldsymbol{\Psi} = \begin{pmatrix} \sigma_L^2 & \sigma_{LS} \\ \sigma_{LS} & \sigma_S^2 \end{pmatrix}.$$

LBGM contains two latent variables: b_L and b_S. b_L represents the intercept and b_S represents the growth slope. The specification of factor loadings of b_S determines the shape of the growth curve. Here, the first and second-factor loadings on b_S are fixed at 0 and 1 for identification purposes, while other factor loadings are freely estimated. This assumption implies that the growth unit is the difference between the first two measurements. Another common practice is to fix the first and last factor loadings at 0 and 1, respectively, with the unit representing the difference between the first and last measurements. β_L and β_S represent the average intercept and slope across all individuals, respectively. σ_L^2 and σ_S^2 represent variances, reflecting the individual differences in intercept and slope. σ_{LS} represents the covariance between the intercept and slope.

The linear growth curve model The specification of $\mathbf{\Lambda}$ decides the shape of growth. When the factor loadings of b_S are equally spaced, it becomes a linear growth curve model. A linear growth curve model assumes a linear change pattern and the slope b_S represents a linear slope. The factor loading matrix is:

$$\mathbf{\Lambda} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 1 & T-1 \end{pmatrix}.$$

The quadratic growth curve model The quadratic growth curve model estimates a nonlinear change by including the quadratic slope b_{iQ}, and the model can be presented as:

$$\mathbf{\Lambda} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 2^2 \\ \vdots & \vdots & \vdots \\ 1 & T-1 & (T-1)^2 \end{pmatrix} \qquad \boldsymbol{b}_i = \begin{pmatrix} b_{iL} \\ b_{iS} \\ b_{iQ} \end{pmatrix},$$

$$\boldsymbol{\beta} = \begin{pmatrix} b_{iL} \\ b_{iS} \\ b_{iQ} \end{pmatrix} \qquad \mathbf{\Psi} = \begin{pmatrix} \sigma_L^2 & \sigma_{LS} & \sigma_{LQ} \\ \sigma_{LS} & \sigma_S^2 & \sigma_{SQ} \\ \sigma_{LQ} & \sigma_{SQ} & \sigma_Q^2 \end{pmatrix}.$$

2.2 Robust growth curve models

The general LGCM assumes $\boldsymbol{\epsilon}_i$ follow a multivariate normal distribution ($\boldsymbol{\epsilon}_i \sim MN(0, \mathbf{\Phi})$), while robust growth curve models use other distributions to for $\boldsymbol{\epsilon}_i$. Zhang (2016) presented and summarized how to use Student's t, exponential power, and the skew normal distributions to build robust LGCM.

Student's t-distribution The robust growth curve models can be specified by modeling $\boldsymbol{\epsilon}_i$ by a student's t-distribution: $\boldsymbol{\epsilon}_i \sim MT_T(0, \Phi, k)$, where k is the degrees of freedom. The robust LGCM with a Students' t-distribution performs better than the traditional growth curve model with a multivariate normal distribution when dealing with heavy-tailed data and outliers (Zhang, 2016; Zhang et al., 2013).

The multivariate t-distribution approaches the multivariate normal distribution when k increases. In the robust Bayesian methods, k can be specified as an unknown parameter, and a prior is needed to estimate k. Alternatively, it can be fixed and some researchers suggested $k = 5$ (Zhang et al., 2013).

In JAGS, t-distribution can be specified using the function `dt()`, and this function will be explained in the following section with an example.

Exponential power distribution The exponential power distribution can model error term e_{it} with smaller kurtosis than normal distributions, and we employ the same form of density function and parameters as Zhang (2016) in this tutorial. The density of exponential power distribution is as follows:

$$p_{ep}(x) = \omega(\gamma)\sigma^{-1} exp\left[-c(\gamma)\left|\frac{x-\mu}{\sigma}\right|^{2/(1+\gamma)}\right]$$

where

$$\omega(\gamma) = \frac{\left(\Gamma\left[3(1+\gamma)/2\right]\right)^{1/2}}{(1+\gamma)\left(\Gamma\left[(1+\gamma)/2\right]\right)^{3/2}}$$

and

$$c(\gamma) = \left(\frac{\Gamma[3(1+\gamma)/2]}{\Gamma[(1+\gamma)/2]}\right)^{1/(1+\gamma)}.$$

Here μ and σ are location and scale parameters, respectively, and γ is a shape parameter that can be estimated.

Skew normal distribution Both the t-distribution and the exponential power distribution are symmetric, while the skew normal distribution offers an option to model asymmetric errors. The density function of a skew normal distribution is as follows:

$$p_{sn}(x) = \frac{2}{\omega}\phi\left(\frac{x-\mu}{\omega}\right)\Phi\left(\alpha\frac{x-\mu}{\omega}\right)$$

where μ is a location parameter, ω is a scale parameter, and α is a shape parameter which can be estimated.

3 Robust growth curve model using JAGS

The following part introduces how to build and interpret the robust LGCM in `JAGS` using a real data set, assuming homogeneity across time points.

3.1 Specification of priors

Priors of LGCM are usually specified as: $\boldsymbol{\beta} \sim N(\mu_0, \sigma_0^2)$, $\boldsymbol{\Phi} \sim W(V, m)$, $\sigma_e^2 \sim IG(\alpha, \beta)$ (assuming $\boldsymbol{\Phi} = \mathbf{I}_{T\times T}\sigma_e^2$). For the robust growth curve model with the t-distribution, the degrees of freedom k is another unknown parameter, and an uninformative prior is applied to k as follows: $k \sim U(1, 500)$. In the case of the exponential power distribution which involves an additional shape parameter γ, an uninformative prior is assigned to it as follows: $\gamma \sim U(-1, 1)$. Similarly, for the shape parameter α in the skew normal distribution, the prior is specified as $\alpha \sim U(-5, 5)$.

3.2 Convergence diagnostic

To check convergence, trace plots are visually inspected. If trace plots indicate non-convergence, then more iterations and longer burn-in periods are needed. The length of the chain should be extended until trace plots of all parameters demonstrate visual convergence.

In addition to visual inspection, various convergence diagnostic tools are available in R, including the Geweke test (Geweke, 1992), the Heidelberger and Welch test (Heidelberger & Welch, 1983), Gelman and Rubin test (Gelman & Rubin, 1992), and the Raftery and Lewis diagnostic (Raftery, Lewis, et al., 1992). In this tutorial, the Geweke diagnostic is used, which compares the mean difference between two parts of chains, typically the first and last parts. It employs a z test to compare the means of two parts, and if the z test statistic rejects the null hypothesis, it indicates a significant difference.

3.3 Autocorrelation and posterior distribution

The adjacent iterations of the Markov chain may exhibit high dependence, and serious autocorrelation can indicate problems in model estimation such as a problem with the sampling algorithm. The autocorrelation problem can be identified by visual inspections. If visual inspection shows high autocorrelation, increasing the number of iterations or implementing thinning techniques can be beneficial. Additionally, it is important to ensure that the posterior distribution makes substantive sense, taking into account factors such as the parameter's range and standard deviation. For instance, it would be unreasonable if the posterior standard deviation exceeds the parameter's scale.

4 Examples

This section includes R code and `JAGS` commands for constructing robust growth curve models. The t-distribution is offered by `JAGS` and can be directly implemented. In the following parts, t-distribution is utilized to model and compare LBGC, linear and quadratic LGCM. To illustrate different robust methods, we specify linear LGC models using the t, exponential power and skew-normal distributions.

The data used in this tutorial were obtained from the Early Childhood Longitudinal Study, Kindergarten Class of 2010-11 (ECLS-K:2011), a national longitudinal program conducted by the National Center for Education Statistics. ECLS-K:2011 collected information about children's development during their elementary school years. For this tutorial, a random subset of data consisting of $N = 200$ samples was selected from ECLS-K:2011. This subset includes math scores measured at four different occasions. Math ability assessments were conducted annually, spanning from the second grade to the fifth grade. Detailed information about ECLS-K:2011 can be found in the manual provided by Tourangeau et al. (2015).

Descriptive analysis revealed that the distributions of the observed math scores were skewed and exhibited heavier tails than normal distributions, as depicted in Figure 1. Additionally, increasing trends in math scores were observed, and the growth pattern of each individual is illustrated in Figure 2.

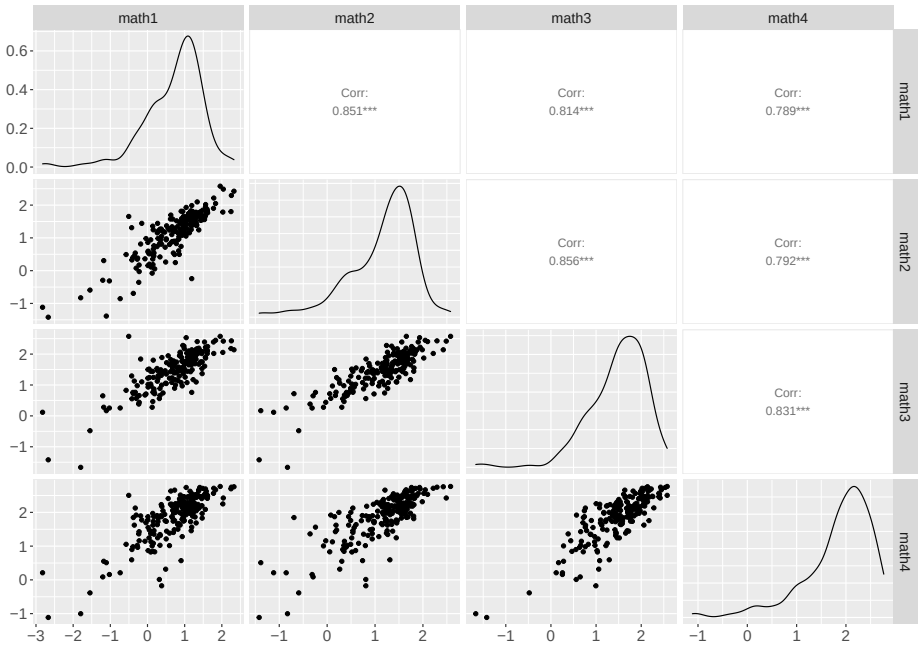

Figure 1. Descriptive plots of math scores

4.1 Specify the JAGS models

t distribution The LBGM model is specified using the JAGS notations as:

```
# models
```

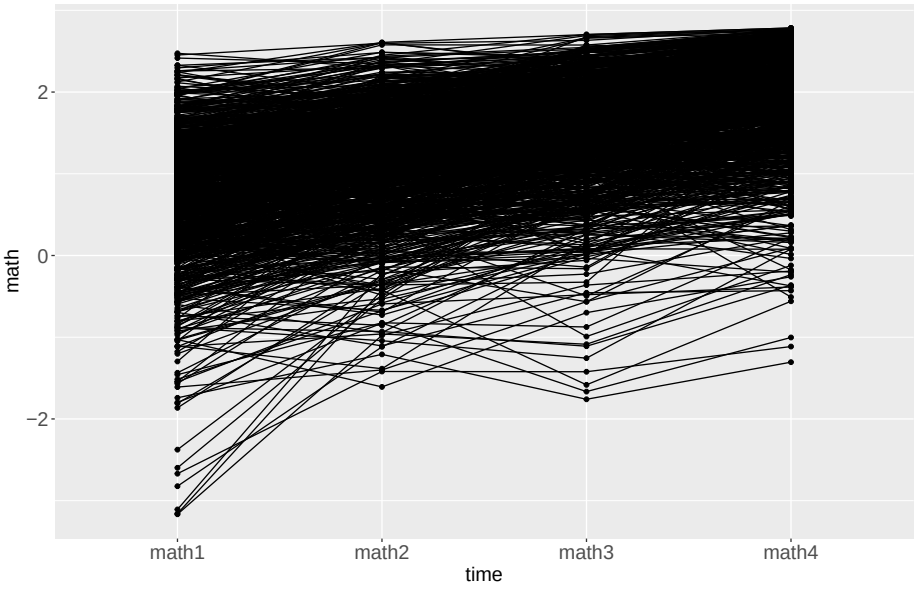

Figure 2. Growth curves of math scores in four waves

```
model1 <- "model{
# Specify the likelihood
for (i in 1:nsubj) {
    for (j in 1:ntime) {
    # t error
    y[i, j] ~ dt(mu[i, j], tauy, df)
    # normal
    # y[i, j] ~ dnorm(mu[i, j],tauy)
  }
}
for (i in 1:nsubj){
  mu[i,1] <- b[i,1]
  mu[i,2] <- b[i,1]+b[i,2]
  mu[i,3] <- b[i,1]+A3*b[i,2]
  mu[i,4] <- b[i,1]+A4*b[i,2]
  b[i,1:2] ~ dmnorm(mub[1:2], taub[1:2,1:2])
}
# Specify the growth trajectory
A3~dnorm(0,1.0E-6)
A4~dnorm(0,1.0E-6)
# specify priors
mub[1]~dnorm(0,1.0E-6)
mub[2]~dnorm(0,1.0E-6)
```

```
taub[1:2, 1:2] ~ dwish(Omega[1:2, 1:2], 2)
sigma2b[1:2, 1:2] <- inverse(taub[1:2, 1:2])
tauy ~ dgamma(0.001,0.001)
sigma2y <- 1 / tauy
df ~  dunif(1,500)
Omega[1,1] <- 1
Omega[2,2] <- 1
Omega[1,2] <- Omega[2,1]
Omega[2,1] <- 0
}
"
```

An R object `model1` is constructed using the `model` block. In the case of a four-wave of data organized with 200 rows ($N = 200$) and 4 columns ($T = 4$), we use for loops to specify the likelihood for all participants across the four measurement occasions.

This likelihood reflects the use of a robust Bayesian method. Specifically, `y[i,j]` is modeled using univariate t-distributions, which are defined by `dt()` with parameters for means `mu[i, j]`, precision `tauy`, and degrees of freedom `df`. If the data were modeled using a multivariate normal distribution, i.e., `y[i,j] ~ dnorm(mu[i,j], tauy)`, the model would represent a traditional latent growth curve model with normal assumptions.

The next part of the model involves specifying the prior distributions. $\boldsymbol{\beta}$ is assumed to follow a bivariate normally distribution with $\boldsymbol{\beta} \sim MN((0,0)^T, 1000\boldsymbol{I_2})$, and the covariance of $\boldsymbol{\epsilon}_i$ follows an inverse Wishart distribution (Zhang, 2021). The error term $\boldsymbol{\epsilon}_i$ is assumed to follow a t-distribution with an estimated k ($\boldsymbol{\epsilon}_i \sim MT_T(0, \boldsymbol{\Phi}, k)$). Here, a uniform distribution $Unif(1, 500)$ is used as the prior of k.

The latent variables and means are specified based on the hypothesized growth curve and priors. The parameter `b[i,1]` represents the latent intercept of LGCM, and the latent slope is `b[i,2]`. `A3` and `A4` are factor loadings of math scores at the third and fourth measurement occasions, which control the shape of changes.

If `A3` is set to 2 and `A4` is set to 3, then the model becomes a linear growth curve model. Quadratic LGCM involves three latent variables, within `b[i,3]` representing the quadratic shape. The coefficients `A5` and `A6` are fixed at 4 and 9, respectively.

Detailed `JAGS` models for both the linear and quadratic growth curve models can be found in the appendix.

Exponential power distribution `JAGS` does not offer exponential power distribution or the skew-normal distribution by default. However, the likelihood can be specified indirectly using the Bernoulli or the Poisson distributions (Ntzoufras, 2011).

One approach, known as the "zero trick," utilizes the Poisson distribution. A matrix with the same dimensions of the data is created, with all elements

set to zero. The likelihood is reflected in the mean of the Poisson distribution. Assuming observation y_i follows a new distribution and the log-likelihood is $l_i = logf(y_i|\theta)$. The model likelihood can be expressed as:

$$f(y|\theta) = \prod_{i=1}^{n} \frac{e^{-(-l_i+c)}(-l_i + C)^0}{0!} = \prod_{i=1}^{2} f_P(0; -l_i + C).$$

In this expression, the mean of the Poisson distribution is a constant (C) minus the log-likelihood ($C - l_i$) and C is chosen to ensure the mean of the Poisson distribution is always positive.

The one trick sets all observations to one and uses the parameter of the Bernoulli distribution to specify likelihood.

In this paper, the zero tricks were used to specify exponential power and skew-normal distributions, assuming a linear change trajectory.

The model code is provided in the appedix. In the code, the log gamma function is specified using command `loggam()`, and `dpois()` is used to sample from the Poisson distribution.

Skew normal distribution The location parameter of the skew normal distribution is reparameterized as

$$\mu = \omega \frac{\alpha}{\sqrt{(1+\alpha^2)^2}} \sqrt{(2/\pi)}$$

to ensure that the mean of the error is zero. In the code, the standard normal cumulative density function is specified by `phi()` and the log density function of the normal distribution is specified by `logdensity.norm()`.

4.2 Specify iterations,initial values, and saved parameters

After configuring the models, we can proceed by organizing the data in a list, specifying initial values, and running the `JAGS` model.

The data is organized in a wide format and stored in a list called `datalist`, which includes the number of participants (`nsubj`) and the number of measurements (`ntime`). In this setup, we use two chains (`nChains = 2`), each with a length of 20,000 iterations (`nIter = 20000`), and a burn-in period of 10,000 iterations (`burnInSteps = 10000`). Monitored parameters encompass the means and variances of intercepts and slopes, and the shape parameters such as the degrees of freedom. These parameters' posterior draws will be saved.

```
# create data set for \texttt{JAGS} model
nsubj = nrow(data)
ntime = ncol(data)
datalist = list(nsubj=nsubj,ntime=ntime,y=data)
# set parameters, adaption, and MCMC chains
parameters = c("mub","sigma2b","sigma2y","df","A3","A4")
adaptSteps =5000           # Adaptive period
```

```
burnInSteps = 10000       # Burn-in period
nChains = 2     # The number of chains
nIter =20000    # The number of kept iterations
```

Two chains are used in this tutorial, and two sets of initial values are specified:

```
# specify initial values
inits <- list(list(mub=c(0.7,0.4),
                   taub=structure(.Data=c(1,0,0,10),
                                  .Dim=c(2,2)),
                   tauy=10,df=3),
              list(mub=c(0.8,0.5),
                   taub=structure(.Data=c(2,0,0,8),
                                  .Dim=c(2,2)),
                   tauy=15,df=5))
```

4.3 Run JAGS models

The package `runjags` is used in this tutorial and the function `run.jags()` is used to read, compile, and run the model, and the model results are saved for later analysis.

```
# run JAGS model
set.seed(1234)
out <- run.jags(model=model,
                monitor=parameters,
                data=datalist, n.chains=2,
                inits=inits, method="simple",
                adapt=adaptSteps,
                burnin = burnInSteps,
                sample=nIter,
                keep.jags.files=TRUE,
                tempdir=TRUE)
```

4.4 Convergence diagnostic

For convergence checking, we examine both trace plots and Geweke's test. A visual inspection of the trace plots reveals that all parameters have converged after the adaptation and burn-in period. Figure 3 displays the plots of the latent intercept and slope in the LBGM.

If Geweke's test values exceeded 2, we doubled the number of iterations and reran the model. In this particular example, we found no clear evidence of non-convergence, however, some models exhibited autocorrelation issues in the slope, as shown in the autocorrelation plots. To address this, longer iterations or thinning techniques may be employed.

Additionally, posteriors make practical sense by checking the shape and range in the posteriors plots. For example, the range of possible values for math ability

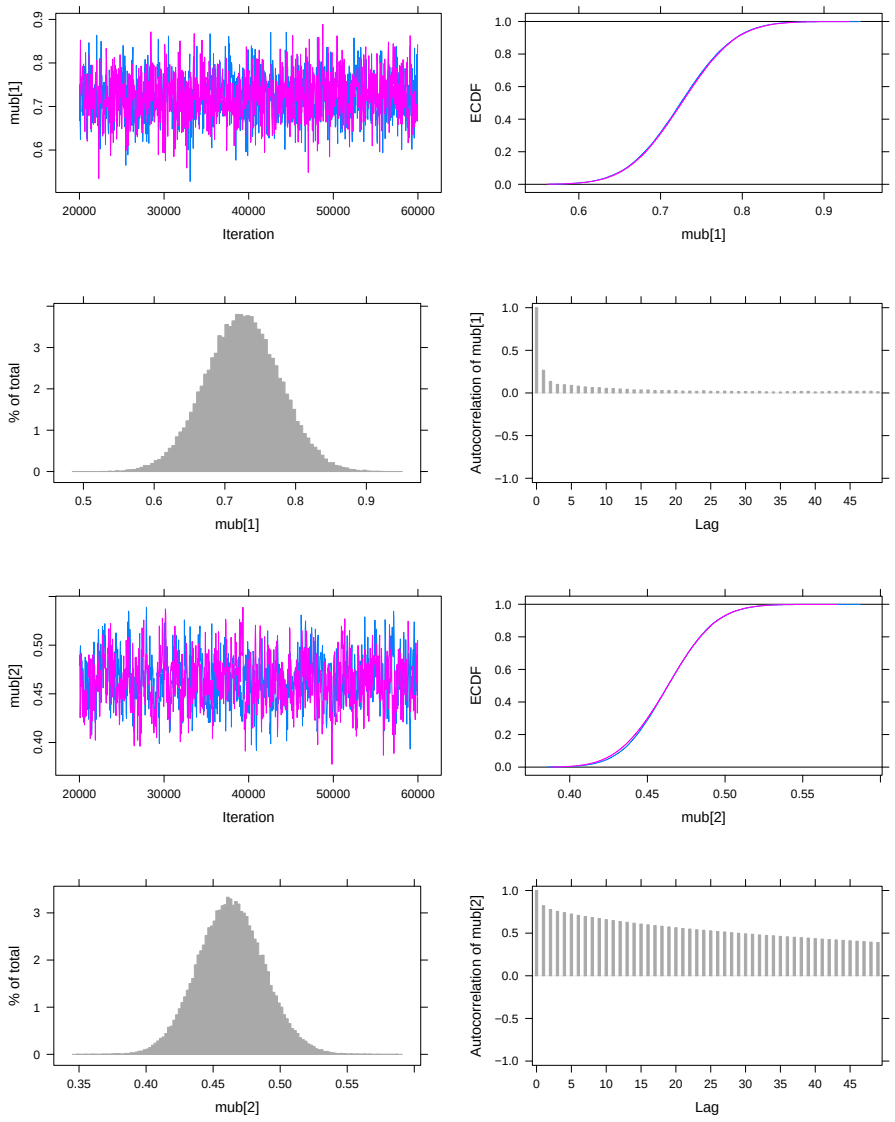

Figure 3. Trace, ECDF, posterior and autocorrelation plots of the intercept and the slope in LBGM with a t distribution

is from -4.0 to 4.0, and the posterior mean of the intercept was close to the mean of the observed math score in the second grade.

```
# Geweke diagnostic
geweke.diag(out$mcmc)
# Trace plots and autocorrelation plots
plot(out)
```

4.5 Model comparison

Results of LBGM, the linear and quadratic LGC models are summarized in Table 1. To compare these models, we used the deviance information criterion (DIC). When the t distribution was employed, the quadratic LGCM exhibited the lowest DIC.

Table 1. Results for the LBGM, linear and quadratic LGC models with t distribution

	LBGM			Linear LGCM			Quadratic LGCM		
	Mean	L	U	Mean	L	U	Mean	L	U
b_{iL}	0.73	0.62	0.83	0.76	0.65	0.85	0.74	0.64	0.85
b_{iS}	0.46	0.42	0.51	0.37	0.34	0.39	0.42	0.35	0.48
b_{iQ}							-0.02	-0.04	0.01
σ_L^2	0.49	0.38	0.59	0.48	0.38	0.59	0.54	0.42	0.66
σ_{LS}	-0.05	-0.07	-0.02	-0.04	-0.06	-0.02	-0.11	-0.18	-0.05
σ_{LQ}							0.02	0.00	0.04
σ_{LS}	-0.05	-0.07	-0.02	-0.04	-0.06	-0.02	-0.11	-0.18	-0.05
σ_S^2	0.03	0.02	0.03	0.02	0.02	0.03	0.09	0.05	0.13
σ_{SQ}							-0.02	-0.04	-0.01
σ_{LQ}							0.02	0.00	0.04
σ_{SQ}							-0.02	-0.04	-0.01
σ_Q^2							0.01	0.01	0.02
σ_e^2	0.03	0.02	0.04	0.03	0.02	0.04	0.03	0.02	0.04
k	3.44	2.30	4.76	3.53	2.31	4.94	3.58	2.04	5.42
A3	1.64	1.51	1.77	2.00			2.00		
A4	2.44	2.26	2.64	3.00			3.00		
A5							4.00		
A6							9.00		
DIC	370.79			374.78			283.36		

Note. k represents the degrees of freedom. L: 2.5% HPD; U: 97.5% HPD.

The estimated means of the intercept b_{iL} from the three models were close. The estimated factor loadings in LBGM were 1.64 and 2.44 in LBGM, which suggests the estimated growth shape was different from a linear trend.

The estimated degrees of freedom were smaller than 5 in the three models. This aligns with the observation that the observed data had heavier tails than the normal distribution, as shown in Figure 1(Tong & Zhang, 2017). Therefore,

the estimated degrees of freedom (k) are consistent with descriptive statistics, affirming that the robust growth curve models are suitable for handling this dataset.

When dealing with models that use exponential power and skew normal distributions, it's important to interpret the DIC (deviance information criterion) values from JAGS with caution. In these models, the DIC is calculated separately based on likelihood and posteriors. The deviance, denoted as $D(\theta; y)$, is defined as $-2\log(p(x|\theta))$. The effective model parameters is defined as $p_D = \bar{D} - \hat{D}$, and the DIC is calculated as $DIC = \bar{D} + p_D$. The model using the skew normal distribution exhibited the lowest DIC value, making it the preferred choice over the t and exponential power distributions.

Table 2. Results for linear models with t, exponential power, and skew normal distributions

	Mean	2.5% HPD	97.5% HPD	DIC
t-distribution				
b_{iL}	0.76	0.65	0.85	
b_{iS}	0.37	0.34	0.39	
σ_L^2	0.48	0.38	0.59	
σ_{LS}	-0.04	-0.06	-0.02	374.78
σ_{LS}	-0.04	-0.06	-0.02	
σ_S^2	0.02	0.02	0.03	
σ_e^2	0.03	0.02	0.04	
k	3.53	2.31	4.94	
Exponential power distribution				
b_{iL}	0.76	0.66	0.86	
b_{iS}	0.37	0.34	0.4	
σ_L^2	0.49	0.39	0.6	
σ_{LS}	-0.04	-0.06	-0.02	392.10
σ_{LS}	-0.04	-0.06	-0.02	
σ_S^2	0.02	0.02	0.03	
σ_e^2	0.07	0.06	0.08	
γ	0.91	0.76	1	
Skew normal distribution				
b_{iL}	0.74	0.64	0.83	
b_{iS}	0.37	0.35	0.4	
σ_L^2	0.46	0.36	0.56	
σ_{LS}	-0.04	-0.06	-0.02	360.41
σ_{LS}	-0.04	-0.06	-0.02	
σ_S^2	0.02	0.02	0.03	
σ_e^2	0.18	0.15	0.21	
α	-4.17	-5	-3.01	

4.6 Summary of posteriors

The posterior means of most parameters were almost the same for linear models using t, exponential power, and skew-normal distributions, see Table 2. For the linear LGCM with the exponential power error, the estimated shape parameter γ was 0.91, which suggested a fatter tail of the errors than the normal distribution. The estimated α in the model using the skew-normal distribution was -4.17 which indicates the distribution was left-skewed.

5 Summary

LGCM is widely used in longitudinal studies, and the Bayesian approach can be applied to handle complex conditions. Bayesian approaches can handle the conditions that data are not normally distributed or the sample size is small. The robust Bayesian method offers an operable solution for data with heavy tails or outliers.

This tutorial introduces how to implement robust LGCM with three distributions in JAGS and R in steps. It also covers the model diagnostics and comparison, and interpretations of posterior estimations. This tutorial offers some guidelines for researchers who are interested in robust Bayesian growth curve models.

References

Curran, P. J., Obeidat, K., & Losardo, D. (2010). Twelve frequently asked questions about growth curve modeling. *Journal of cognition and development*, *11*(2), 121–136. doi: https://doi.org/10.1080/15248371003699969

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 457–472. doi: https://doi.org/10.1214/ss/1177011136

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculations of posterior moments. *Bayesian statistics*, *4*, 641–649.

Heidelberger, P., & Welch, P. D. (1983). Simulation run length control in the presence of an initial transient. *Operations Research*, *31*(6), 1109–1144. doi: https://doi.org/10.1287/opre.31.6.1109

Lange, K. L., Little, R. J., & Taylor, J. M. (1989). Robust statistical modeling using the t distribution. *Journal of the American Statistical Association*, *84*(408), 881–896. doi: https://doi.org/10.2307/2290063

McArdle, J. J., & Nesselroade, J. R. (2014). *Longitudinal data analysis using structural equation models.* American Psychological Association. doi: https://doi.org/10.1037/14440-000

Ntzoufras, I. (2011). *Bayesian modeling using winbugs.* John Wiley & Sons. doi: https://doi.org/10.1080/09332480.2012.685377

Raftery, A. E., Lewis, S., et al. (1992). How many iterations in the gibbs sampler. *Bayesian statistics*, *4*(2), 763–773.

Tong, X., & Zhang, Z. (2017). Outlying observation diagnostics in growth curve modeling. *Multivariate Behavioral Research*, *52*(6), 768–788. doi: https://doi.org/10.1080/00273171.2017.1374824

Tourangeau, K., Nord, C., Lê, T., Sorongon, A. G., Hagedorn, M. C., Daly, P., & Najarian, M. (2015). Early childhood longitudinal study, kindergarten class of 2010-11 (ecls-k: 2011). user's manual for the ecls-k: 2011 kindergarten data file and electronic codebook, public version. nces 2015-074. *National Center for Education Statistics*.

Zhang, Z. (2016). Modeling error distributions of growth curve models through bayesian methods. *Behavior research methods*, *48*, 427–444. doi: https://doi.org/10.3758/s13428-015-0589-9

Zhang, Z. (2021). A note on wishart and inverse wishart priors for covariance matrix. *Journal of Behavioral Data Science*, *1*(2), 119–126. doi: https://doi.org/10.35566/jbds/v1n2/p2

Zhang, Z., Lai, K., Lu, Z., & Tong, X. (2013). Bayesian inference and application of robust growth curve models using student's t distribution. *Structural Equation Modeling: A Multidisciplinary Journal*, *20*(1), 47–78. doi: https://doi.org/10.1080/10705511.2013.742382

Appendix A Data

```
data=read.csv("example_data.csv",header = T)
colnames(data)=c("ID",paste("math",rep(1:4),sep=''))
data=data[,-1]
```

Appendix B Using the t-distribution for error

```
# The latent basis growth curve model
model1 <- "model{
# specify the likelihood
for (i in 1:nsubj) {
    for (j in 1:ntime) {
    # t error
    y[i, j] ~ dt(mu[i, j], tauy, df)
    # normal
    # y[i, j] ~ dnorm(mu[i, j],tauy)
  }
}
for (i in 1:nsubj){
  mu[i,1] <- b[i,1]
  mu[i,2] <- b[i,1]+b[i,2]
  mu[i,3] <- b[i,1]+A3*b[i,2]
```

```
  mu[i,4] <- b[i,1]+A4*b[i,2]
  b[i,1:2] ~ dmnorm(mub[1:2], taub[1:2,1:2])
}
# specify the growth trajectory
A3~dnorm(0,1.0E-6)
A4~dnorm(0,1.0E-6)
# specify priors
mub[1]~dnorm(0,1.0E-6)
mub[2]~dnorm(0,1.0E-6)
taub[1:2, 1:2] ~ dwish(Omega[1:2, 1:2], 2)
sigma2b[1:2, 1:2] <- inverse(taub[1:2, 1:2])
tauy ~ dgamma(0.001,0.001)
sigma2y <- 1 / tauy
df ~  dunif(1,500)
Omega[1,1] <- 1
Omega[2,2] <- 1
Omega[1,2] <- Omega[2,1]
Omega[2,1] <- 0
}
"
# write model out
writeLines(model1, "model1.txt")


# set parameters, adaption, and MCMC chains
parameters = c("mub","sigma2b","sigma2y","df",
"A3","A4","dic")# Specify the estimated parameters
adaptSteps =10000           # Adaptive period
burnInSteps = 10000         # Burn-in period
nChains = 2
nIter =40000    # The number of kept iterations

nsubj = nrow(data)
ntime = ncol(data)

# create data set for JAGS model
datalist = list(nsubj=nsubj,ntime=ntime,y=as.matrix(data))

# specify initial values
inits <- list(list(mub=c(0.7,0.4),
  taub=structure(.Data = c(1,0,0,10),.Dim=c(2,2)),
  tauy=10,df=3),
  list(mub=c(0.7,0.5),
  taub=structure(.Data = c(2,0,0,8),.Dim=c(2,2)),
  tauy=15,df=5))
```

```
# run jags model
set.seed(1234)
out <- run.jags(model=model,
                monitor=parameters,
                data=datalist, n.chains=2,
                inits=inits, method="simple",
                adapt=adaptSteps,
                burnin = burnInSteps,
                sample=nIter,
                keep.jags.files=TRUE,
                tempdir=TRUE)
# diagnostic
geweke.diag(out$mcmc)
# plots
# trace plots and autocorrelation plots
plot(out)
# Summarize posterior distributions
mcmcChain = as.matrix(out$mcmc)
sum = summary(out$mcmc)

# The linear LGCM
model2 <- "model{
# likelihood
for (i in 1:nsubj) {
    for (j in 1:ntime) {
    # t error
    y[i, j] ~ dt(mu[i, j], tauy, df)
  }
}
# growth trajectory
for (i in 1:nsubj){
  mu[i,1] <- b[i,1]
  mu[i,2] <- b[i,1]+b[i,2]
  mu[i,3] <- b[i,1]+A3*b[i,2]
  mu[i,4] <- b[i,1]+A4*b[i,2]
  b[i,1:2] ~ dmnorm(mub[1:2], taub[1:2,1:2])
}

A3 <- 2 # linear change
A4 <- 3
mub[1]~dnorm(0,1.0E-6)
mub[2]~dnorm(0,1.0E-6)
taub[1:2, 1:2] ~ dwish(Omega[1:2, 1:2], 2)
```

```
sigma2b[1:2, 1:2] <- inverse(taub[1:2, 1:2])
tauy ~ dgamma(0.001,0.001)
sigma2y <- 1 / tauy
df ~  dunif(1,500)
Omega[1,1] <- 1
Omega[2,2] <- 1
Omega[1,2] <- Omega[2,1]
Omega[2,1] <- 0
}
"


# Quadratic LGCM
model3 <- "model{
# likelihood
for (i in 1:nsubj) {
    for (j in 1:ntime) {
    # t error
    y[i, j] ~ dt(mu[i, j], tauy, df)
  }
}
# growth trajectory
for (i in 1:nsubj){
  mu[i,1] <- b[i,1]
  mu[i,2] <- b[i,1]+b[i,2]+b[i,3]
  mu[i,3] <- b[i,1]+A3*b[i,2]+A5*b[i,3]
  mu[i,4] <- b[i,1]+A4*b[i,2]+A6*b[i,3]
  b[i,1:3] ~ dmnorm(mub[1:3], taub[1:3,1:3])
}
# linear change
A3 <- 2
A4 <- 3
# quadratic change
A5 <- 4
A6 <- 9
mub[1]~dnorm(0,1.0E-6)
mub[2]~dnorm(0,1.0E-6)
mub[3]~dnorm(0,1.0E-6)
taub[1:3,1:3] ~ dwish(Omega[1:3, 1:3], 3)
sigma2b[1:3, 1:3] <- inverse(taub[1:3,1:3])
tauy ~ dgamma(0.001,0.001)
sigma2y <- 1 / tauy
df ~  dunif(1,500)
Omega[1,1] <- 1
Omega[2,2] <- 1
```

```
Omega[3,3] <- 1
Omega[1,2] <- Omega[2,1]
Omega[1,3] <- Omega[3,1]
Omega[2,3] <- Omega[3,2]
Omega[2,1] <- 0
Omega[3,1] <- 0
Omega[3,2] <- 0
}
"
```

Appendix C Using exponential power distribution for error

```
# A linear LGCM
model4 <- "model{
C <- 100000
lomega <- 0.5*loggam(3*(1+gamma)/2)-log(1+gamma)
   -3/2*loggam((1+gamma)/2)
cgamma <- (exp(loggam(3*(1+gamma)/2))
    /exp(loggam((1+gamma)/2)))^(1/(1+gamma))
for (i in 1:nsubj) {
    for (j in 1:ntime) {
    # Exponential power
    zeros[i,j] ~ dpois(zeros.mean[i,j])
    zeros.mean[i,j] <- C-le[i,j]
    le[i,j] <- lomega-log(sqrt(sigma2y))
    -cgamma*abs((y[i,j]-mu[i,j])
        /sqrt(sigma2y))^(2/(1+gamma))
  }
}
# growth trajectory
for (i in 1:nsubj){
  mu[i,1] <- b[i,1]
  mu[i,2] <- b[i,1]+b[i,2]
  mu[i,3] <- b[i,1]+A3*b[i,2]
  mu[i,4] <- b[i,1]+A4*b[i,2]
  b[i,1:2] ~ dmnorm(mub[1:2], taub[1:2,1:2])
}
A3 <- 2 # linear change
A4 <- 3
mub[1]~dnorm(0,1.0E-6)
mub[2]~dnorm(0,1.0E-6)
taub[1:2, 1:2] ~ dwish(Omega[1:2, 1:2], 2)
sigma2b[1:2, 1:2] <- inverse(taub[1:2, 1:2])
```

```
tauy ~ dgamma(0.001,0.001)
sigma2y <- 1 / tauy
gamma ~  dunif(-1,1)
Omega[1,1] <- 1
Omega[2,2] <- 1
Omega[1,2] <- Omega[2,1]
Omega[2,1] <- 0
}
"
```

Appendix D Using the skew normal distribution

```
# A linear LGCM
model5 <- "model{
C <- 100000
xi <- -sqrt(sigma2y)*(alpha/sqrt(1+alpha^2))
    *sqrt(2/3.1415)
for (i in 1:nsubj) {
    for (j in 1:ntime) {
    # Exponential power
    zeros[i,j] ~ dpois(zeros.mean[i,j])
    zeros.mean[i,j] <- C-le[i,j]
    e[i,j] <- y[i,j]-mu[i,j]
    # phi(): standard normal cdf
    # the log density of x is given by
    le[i,j] <- log(2)-log(sqrt(sigma2y))
    +logdensity.norm((e[i,j]-xi)/sqrt(sigma2y),0,1)
    +log(phi(alpha*(e[i,j]-xi)/sqrt(sigma2y)))
  }
}
# growth trajectory
for (i in 1:nsubj){
  mu[i,1] <- b[i,1]
  mu[i,2] <- b[i,1]+b[i,2]
  mu[i,3] <- b[i,1]+A3*b[i,2]
  mu[i,4] <- b[i,1]+A4*b[i,2]
  b[i,1:2] ~ dmnorm(mub[1:2], taub[1:2,1:2])
}
A3 <- 2 # linear change
A4 <- 3
mub[1]~dnorm(0,1.0E-6)
mub[2]~dnorm(0,1.0E-6)
taub[1:2, 1:2] ~ dwish(Omega[1:2, 1:2], 2)
sigma2b[1:2, 1:2] <- inverse(taub[1:2, 1:2])
```

```
tauy ~ dgamma(0.001,0.001)
sigma2y <- 1 / tauy
alpha ~  dunif(-5,5)
Omega[1,1] <- 1
Omega[2,2] <- 1
Omega[1,2] <- Omega[2,1]
Omega[2,1] <- 0
}
"
```

Journal of Behavioral Data Science, 2023, *3* (2), 64–126.
DOI:https://doi.org/10.35566/jbds/v3n2/wang

Conducting Meta-analyses of Proportions in R

Naike Wang

Texas A&M University, College Station, TX 77843, USA
wangnaike@tamu.edu

Abstract. Meta-analysis of proportions has been widely adopted across various scientific disciplines as a means to estimate the prevalence of phenomena of interest. However, there is a lack of comprehensive tutorials demonstrating the proper execution of such analyses using the R programming language. The objective of this study is to bridge this gap and provide an extensive guide to conducting a meta-analysis of proportions using R. Furthermore, we offer a thorough critical review of the methods and tests involved in conducting a meta-analysis of proportions, highlighting several common practices that may yield biased estimations and misleading inferences. We illustrate the meta-analytic process in five stages: (1) preparation of the R environment; (2) computation of effect sizes; (3) quantification of heterogeneity; (4) visualization of heterogeneity with the forest plot and the Baujat plot; and (5) explanation of heterogeneity with moderator analyses. In the last section of the tutorial, we address the misconception of assessing publication bias in the context of meta-analysis of proportions. The provided code offers readers three options to transform proportional data (e.g., the double arcsine method). The tutorial presentation is conceptually oriented and formula usage is minimal. We will use a published meta-analysis of proportions as an example to illustrate the implementation of the R code and the interpretation of the results.

Keywords: Meta-analysis of proportions · Heterogeneity · Meta-regression · Double arcsine transformation · Baujat plot

1 Introduction

A meta-analysis is a statistical approach that synthesizes quantitative findings from multiple studies investigating the same research topic. Its purpose is to provide a numerical summary of a particular research area, aiming to inform future work in that area. Meta-analyses of proportions are commonly conducted across diverse scientific fields, such as medicine (e.g., Gillen, Schuster, Meyer Zum Bschenfelde, Friess, & Kleeff, 2010), clinical psychology (e.g., Fusar-Poli et al., 2015), epidemiology (e.g., Wu, Long, Lin, & Liu, 2016), and public health (e.g., Keithlin, Sargeant, Thomas, & Fazil, 2014), etc. The outcomes derived

from these studies are often used for decision models (Hunter et al., 2014). Each individual study included in a meta-analysis of proportions contributes a specific number of "successes" and a corresponding total sample size (Hamza, van Houwelingen, & Stijnen, 2008). While the majority of meta-analyses primarily focus on effect-size metrics that measure a relationship between a treatment group and a control group–such as standardized mean difference and odds ratio– the effect-size metric in meta-analyses of proportions is an estimate of the overall proportion related to a particular condition or event across all included studies (Barendregt, Doi, Lee, Norman, & Vos, 2013). For instance, a meta-analysis can be conducted to provide an overall prevalence estimate of homeless veterans affected by both post-traumatic stress disorder and substance use disorder.

The purpose of this tutorial is to provide an introduction to conducting a meta-analysis of proportions using the R software (R Core Team, 2022). We discuss two distinct benefits of choosing R as your primary meta-analysis tool. First, R is freely available open-source software that offers a comprehensive collection of R packages, which are extensions developed for specialized applications, including meta-analysis. This remarkable feature provides researchers with diverse possibilities and flexibility when it comes to data manipulation and analysis. Two widely used R packages for meta-analysis are *metafor* (Viechtbauer, 2010) and *meta* (Schwarzer, Carpenter, & Rücker, 2015). Second, R offers more convenient options for transforming proportional data than other statistical software. The two commonly adopted data transformation methods are the logit and the double arcsine transformations (though not transforming data is also appropriate under certain circumstances). Both the *metafor* and *meta* packages are capable of performing these transformations. In contrast, other meta-analysis software such as Comprehensive Meta-Analysis (CMA) (Borenstein, Hedges, Higgins, & Rothstein, 2005) and MedCalc (Schoonjans, 2017) can only perform one of these transformations. Additionally, while CMA and MedCalc automatically transform data, R allows meta-analysts to make a decision on whether to apply data transformation.

To the best of our knowledge, this is the first tutorial that illustrates the implementation of such analyses. The tutorial offers an overview of the fundamental statistical concepts related to meta-analysis of proportions and provides hands-on code examples to guide readers through the process in R.[1] We use a dataset from a published meta-analytic study to detail the steps involved. Moreover, we've rigorously tested the code in R and validated it using CMA, ensuring identical results from both software.

Last but not least, this tutorial will explain why common publication bias assessment procedures aren't recommended for meta-analyses of proportions.

[1] Throughout this tutorial, we'll present generic code templates for all transformation methods. However, the main text of this tutorial will focus on code examples for the logit transformation, given the similarity in coding across all methods. For R code related to other transformation methods and their associated datasets, please refer to the supplementary files.

2 Preparation of the R environment

2.1 R and RStudio

The first step is to download R. The base R program can be downloaded for free from the Comprehensive R Archive Network (https://cran.r-project.org/). R provides a basic graphical user interface (GUI), but we recommend that readers use a more productive code editor that interfaces with R, known as RStudio (RStudio Team, 2022). This is a development environment built to make using R as effective and efficient as possible, which is freely available at https://www.rstudio.com/. It adds much more functionality above and beyond R's bare-bones GUI.

Once RStudio is successfully installed on your computer and opened, the first step is to create a new R Script. To do this, navigate to the "File" menu. Click on "File", and in the dropdown menu, select "New File", then choose "R Script". A new tab will open in the top-left pane of RStudio, known as the source editor. This space is where you'll write your R code.

2.2 Setting up the working directory

To ensure proper organization of your R files and data, it's crucial to establish a working directory for the current R session. A working directory serves as a centralized location where you can store all your work, including the R code you've written and data files (e.g., .csv files) you wish to import into R for analysis. To set up a working directory, start by creating a folder named "data" in your preferred location on the computer, such as the D drive. After doing so, enter the following code into the source editor:

```
setwd("D:/data")
```

3 Overview of the example data set

3.1 Illustrative example: Prevalence and epidemiological characteristics of congenital cataract (Wu et al., 2016)

The data set we will use for this tutorial is extracted from a published meta-analytic study conducted by Wu et al. (2016). They estimated the prevalence of congenital cataracts (CC) and their main epidemiological traits. CC refers to the opacity of the lens detected at birth or at an early stage of childhood. It is the primary cause of treatable childhood blindness worldwide. Current studies have not determined the etiology of this condition. The few large-scale epidemiological studies on CC also have limitations: they involve specific regions, limited populations, and partial epidemiological variables. Wu et al. (2016) aimed to explore its etiology and estimate its population-based prevalence and major epidemiological characteristics, morphology, associated comorbidities and etiology. The

original dataset consists of 27 published studies that were published from 1983 to 2014, among which 17 contained data on the population-based prevalence of CC, 2 were hospital-based studies and 8 were CC-based case reviews. Samples investigated in the studies were from different regions of the world, including Europe, Asia, the USA, Africa, and Australia. The sample sizes of the included studies ranged from 76 to 2,616,439 patients, with a combined total of 8,302,708 patients. The diagnosed age ranged from 0 to 18 years of age. The proportions were transformed using the logit transformation, which is commonly employed when dealing with proportional data. This transformation results in a sampling distribution that is more normal, with a mean of zero and a standard deviation of 1.83. The authors coded five moderators, including world region (China vs. the rest of the world), study design (birth cohort vs. other), sample size (less vs. more than 100,000), diagnosed age (older vs. younger than 1 year old), and research period (before vs. after the year 2000). All of these potential moderators are categorical variables. Due to page limits, we will work with only a subset of the provided moderating variables, including study design and sample size.

3.2 Recommended format for organizing data

Prior to performing a meta-analysis in R, it is important to first organize the data properly. Table 1 shows an excerpt of the example dataset. Each row in this table represents the data extracted from a primary study included in the current meta-analysis. The columns contain variables that will be used to compute effect sizes, create plots, and conduct further analyses.

Table 1. Data from Wu et al. (2016)

author	year	authoryear	cases	total	studesg	studydesign	size	samplesize
Stewart-Brown	1988	Stewart-Brown 1988	7	12853	0	Birth cohort	0	< 100000
Bermejo	1998	Bermejo 1998	71	1124654	0	Birth cohort	1	> 100000
SanGiovanni	2002	SanGiovanni 2002	73	53639	0	Birth cohort	0	< 100000
Haargaard	2004	Haargaard 2004	773	2616439	0	Birth cohort	1	> 100000
Stayte	1993	Stayte 1993	4	6687	0	Birth cohort	0	< 100000
Stoll	1997	Stoll 1997	57	212479	0	Birth cohort	1	> 100000
Rahi	2001	Rahi 2001	248	734000	1	Others	1	> 100000
Wirth	2002	Wirth 2002	421	1870000	1	Others	1	> 100000
Hu	1987	Hu 1987	77	207319	1	Others	1	> 100000
Abrahamsson	1999	Abrahamsson 1999	136	377334	1	Others	1	> 100000
Bhatti	2003	Bhatti 2003	199	982128	1	Others	1	> 100000
Nie	2008	Nie 2008	15	15398	1	Others	0	< 100000
Chen	2014	Chen 2014	6	9246	1	Others	0	< 100000
Yang	2014	Yang 2014	8	6299	1	Others	0	< 100000
Pi	2012	Pi 2012	3	3079	1	Others	0	< 100000
Holmes	2003	Holmes 2003	10	33021	1	Others	0	< 100000
Halilbasic	2014	Halilbasic 2014	51	38133	1	Others	0	< 100000

In this data set, we have separate columns for authors' names and the year of publication, which will be useful when sorting studies according to the year of publication in R. Additionally, if we decide to use the *forest()* function in the *meta* package to create forest plots, we need to create a column that combines

both variables. In this case, we label the column as "authoryear". It's important to note that when importing a data file into R, column names with uppercase letters will be converted to lowercase. Therefore, we cannot use uppercase or lowercase letters to differentiate between different columns. Moreover, we cannot leave a blank space between two words when naming a column. As seen in the table, we use "authoryear" instead of "author year", "studydesign" instead of "study design", and "samplesize" instead of "sample size".

The variable "cases" represents the number of the event of interest in the sample of each study. By dividing "cases" by "total", we can obtain the proportions needed to compute effect sizes, which are labeled as "yi" in R. R will also calculate the sampling variance for each "yi" and label them as "vi". The remaining variables in the dataset are potential moderators, which will be examined in either a subgroup analysis or a meta-regression. For instance, "study design" is a potential moderator with two categories or levels: "birth cohort" and "others". We have coded each category as either 1 or 0 in the column labeled "studesg". For continuous moderators, readers can create columns to store continuous values, such as the "year" column. This dataset is saved as a comma-separated values (.csv) file named "data.csv" and is included in the online supplemental materials for this tutorial. To import it into R, ensure the .csv file is stored in the working directory.

4 Computation of effect sizes

4.1 Fixed-effect and random-effects model

Before combining effect sizes in a meta-analysis, we need to make a choice between two modeling approaches for calculating the summary effect size:[2] the fixed-effect and random-effects model (Hedges & Vevea, 1998; Hunter & Schmidt, 2000). The fixed-effect model assumes that studies included in a meta-analysis are functionally equivalent, sharing a common true effect size. Put differently, the true effect size is identical across studies, and any observed variation in effect size estimates is solely due to random sampling error within each study, known as within-study variance. The random-effects model allows the included studies to have true effect sizes that are not identical or "fixed" but follow a normal distribution. In other words, the random-effects model accounts for both within-study and between-study variances, while the fixed-effect model assumes that the between-study variance is zero (i.e., between-study heterogeneity does not exist).

The fixed-effect model applies when participants in the studies are drawn from a single common population and undergo the same experimental procedures conducted by the same researchers under identical conditions. For instance, a series of studies with the same protocol conducted in the same lab and sampling from the same population (e.g., school children from the same class) may fit the fixed-effect model. However, these conditions rarely hold in reality. In fact,

[2] The "summary effect size" and "overall effect size" are interchangeable terms.

the majority of meta-analyses are conducted based on studies collected from the literature. In such cases, we can generally assume that the true effect varies from study to study. Even when a group of studies focuses on a common topic, they are often conducted using different methods (Borenstein, 2019). Consequently, the true effect size is assumed to follow a normal distribution under the random-effects model.

An additional limitation of the fixed-effect model is that its conclusions are limited to the specific set of studies included in the meta-analysis and cannot be generalized to multiple populations. However, most social scientists aim to make inferences that extend beyond the selected set of studies in their meta-analyses. As a general rule of thumb, the random-effects model will be more plausible than the fixed-effect model in most meta-analytic studies because the random-effects model allows more generalizable conclusions beyond a specific population (Borenstein, 2019; Borenstein, Hedges, Higgins, & Rothstein, 2009). However, we discourage the practice of switching to the random-effects model from the fixed-effect model based solely on the results of heterogeneity tests. We will discuss the reasons in more depth later.

The random-effects model can be estimated by several methods (although other methods exist, we will focus on the most popular ones here): the method of moments or the DerSimonian and Laird method (DL; DerSimonian & Laird, 1986) and the restricted maximum likelihood method (REML; Raudenbush & Bryk, 1985). In all cases, the summary effect size (i.e., the summary proportion) is estimated as the weighted average of the observed effect sizes extracted from primary studies. The weighting for each observed effect size is the inverse of the total variance of a study, which is the sum of the within-study variance and the between-study variance (Ma, Chu, & Mazumdar, 2016). These two methods differ mainly in the estimation of the between-study variance, commonly denoted as τ^2 in the meta-analytic literature. The technical differences between these methods have been summarized elsewhere (e.g., Knapp, Biggerstaff, & Hartung, 2006; Thorlund, Wetterslev, Awad, Thabane, & Gluud, 2011; Veroniki et al., 2016) and will not be discussed here.

4.2 Transformation of proportions: the logit transformation and the double arcsine transformation

When the observed proportions are around 0.5 and the number of studies is sufficiently large, the proportions follow an approximately symmetrical binomial distribution. Under such circumstances, the normal distribution is a good approximation of the binomial distribution, and using the raw proportion as the effect-size metric for analysis is appropriate (Barendregt et al., 2013; Box, Hunter, & Hunter, 2005; Wang & Liu, 2016). Additionally, based on their simulation study, Lipsey and Wilson (2001) suggested that when observed proportions derived from primary studies fall between 0.2 and 0.8, and the focus is solely on the mean proportion across the studies, the raw proportion can be adequately employed as the effect-size metric. The procedure for calculating the effect size,

sampling variance, and inverse variance weight for an individual study using the raw proportion is as follows (Lipsey & Wilson, 2001):

The raw proportion is given by:

$$ES_p = p = \frac{k}{n} \tag{1}$$

with its sampling variance:

$$Var_p = SE_p^2 = \frac{p\,(1-p)}{n} \tag{2}$$

and the inverse variance weight:

$$w_p = \frac{1}{Var_p} = \frac{1}{SE_p^2} = \frac{n}{p\,(1-p)} \tag{3}$$

where p is the proportion, k is the number of individuals or cases in the category of interest, and n is the sample size. *ES*, *SE*, *Var*, and w stand for effect size, standard error, sampling variance, and inverse variance weight, respectively.

However, when collecting studies for a meta-analysis of proportions, it is observed that proportional data are rarely centered around 0.5 and often exhibit significant skewness (Hunter et al., 2014). As the proportions deviate further from 0.5 and approach closer to the boundaries (particularly when they are below 0.2 or above 0.8), they become less likely to be normally distributed (Lipsey & Wilson, 2001). Additionally, using the raw proportion as the effect-size metric in such situations may underestimate the coverage of the confidence interval around the weighted average proportion and overestimate the level of heterogeneity among the observed proportions (Lipsey & Wilson, 2001). Consequently, relying on the assumption of normality may lead to biased estimation and potentially misleading or invalid inferences (Feng et al., 2014; Ma et al., 2016).

To address the skewness in the distribution of observed proportions, it is common practice to apply transformations to the observed proportions collected for a meta-analysis. This is done to ensure that the transformed proportions conform as closely as possible to a normal distribution, thus enhancing the validity of subsequent statistical analyses (Barendregt et al., 2013). More specifically, all computations and analyses are performed based on the transformed proportions (e.g., the natural logarithm of the proportion) and their inverted variances (i.e., the study weight). The results, such as the summary proportion and its confidence interval, are presented in the original effect-size metric (i.e., proportion) for ease of presentation and interpretation (Borenstein et al., 2009).

In practice, the approximate likelihood approach (Agresti & Coull, 1998) is arguably the predominant framework for modeling proportional data (Hamza et al., 2008; Nyaga, Arbyn, & Aerts, 2014). There are two main ways to transform observed proportions within this framework: the logit or log odds transformation (Sahai & Ageel, 2012) and the Freeman-Tukey double arcsine transformation (Freeman & Tukey, 1950; Miller, 1978). For the logit transformation, the

observed proportions are first converted to their natural logarithm of the proportions (i.e., the logit). Following the transformation, the logit transformed proportions are assumed to follow a normal distribution, and all analyses are conducted on the logit scale. Subsequently, the logits are converted back into proportions for reporting and interpretation purposes. The procedure for calculating the logit, its standard error and inverse variance weight for primary studies, as well as the formula for back-transformation, are as follows (Lipsey & Wilson, 2001).

The logit is calculated by:

$$ES_l = \log_e\left(\frac{p}{1-p}\right) = \ln\left(\frac{p}{1-p}\right) \tag{4}$$

with its sampling variance:

$$Var_l = SE_l^2 = \frac{1}{np} + \frac{1}{n(1-p)} \tag{5}$$

and the inverse variance weight:

$$w_l = \frac{1}{SE_l^2} = np\,(1-p)\,. \tag{6}$$

To convert the transformed values into proportions, use:

$$p = \frac{e^{logit}}{e^{logit}+1}. \tag{7}$$

Being widely employed in meta-analyses of proportions, the logit transformation still has its limitations in certain situations. Two limitations are particularly noteworthy.

First, the issue of variance instability persists even after applying the logit transformation (Barendregt et al., 2013; Hamza et al., 2008). The purpose of data transformation is to bring the skewed data closer to a normal distribution or at least to achieve more consistent variance. While the logit transformation generates a sampling distribution that approximates normality to a greater extent, it fails to stabilize the variance, potentially placing undue weight on studies. According to the equation for sampling variance (Eq. 5), for a fixed value of n, the variance changes with p. For instance, consider a situation with two studies of the same sample size, where an observed proportion close to 0 or 1 yields grossly magnified variance, while an observed proportion around 0.5 yields squeezed variance, leading to variance instability (Barendregt et al., 2013).

Second, when the event of interest is extremely rare (i.e. $p = 0$) or extremely common (i.e., $p = 1$), the logits and their sampling variances become undefined. In practice, the common solution is to add an arbitrary constant 0.5 correction to the *np* and *n(1-p)* for all studies (Hamza et al., 2008). However, this approach has been shown to introduce additional bias to the results (Lin & Xu, 2020; Ma et al., 2016).

Both of the aforementioned problems can be elegantly solved by employing the variance-stabilizing transformation known as the double arcsine transformation (Freeman & Tukey, 1950), which is accomplished with the following equation[3]:

$$ES_t = \sin^{-1}\sqrt{\frac{k}{n+1}} + \sin^{-1}\sqrt{\frac{k+1}{n+1}} \tag{8}$$

The sampling variance is computed by:

$$Var_t = \frac{1}{n+0.5} \tag{9}$$

The back-transformation is computed by the equation as proposed by Miller (1978):

$$p = \frac{1}{2}\left[1 - sgn\left(\cos t\right)\left[1 - \left(\sin t + \frac{\sin t - \frac{1}{\sin t}}{n'}\right)^2\right]^{\frac{1}{2}}\right] \tag{10}$$

where t denotes the double arcsine transformed value or the confidence interval around it with sgn being the sign operator. In Eq. (10), the total sample size denoted by n' is calculated as the harmonic mean of individual sample sizes (Miller, 1978). The harmonic mean is defined as:

$$n' = m(\sum_{i}^{m} n_i^{-1})^{-1} \tag{11}$$

where n_i denotes the sample size of each included study and m denotes the number of included studies. Miller (1978) gives an example in his paper: a meta-analysis of proportions includes four studies with sample sizes being 11, 17, 21, and 6, respectively. The harmonic mean of the four sample sizes will be:

$$n' = \frac{4}{\frac{1}{11} + \frac{1}{17} + \frac{1}{21} + \frac{1}{6}} = 10.9885. \tag{12}$$

Barendregt et al. (2013) found that Eq. (10) becomes numerically unstable when $\sin t$ is close to 0 or 1, leading to potentially misleading results. This phenomenon has also been documented by recent publications (Evangelou & Veroniki, 2022; Lin & Xu, 2020; Schwarzer, Chemaitelly, Abu-Raddad, & Rücker, 2019). Instead of the harmonic mean, Barendregt et al. (2013) and Xu et al. (2021) recommend using $1/\bar{v}$ as the estimate for the total sample size. They propose that the double arcsine back-transformation be implemented as follows:

$$\bar{p} = \frac{1}{2}\left[1 - sgn(\cos \bar{t})\left[1 - \left(\sin \bar{t} + \frac{\sin \bar{t} - \frac{1}{\sin \bar{t}}}{\frac{1}{\bar{v}}}\right)^2\right]^{\frac{1}{2}}\right] \tag{13}$$

[3] The *metafor* package uses different definitions of Eq.8 and 9. For more details, see https://www.metafor-project.org/doku.php/faq.

where $\bar{p}$ is the pooled proportion on the natural scale and $\bar{v}$ is the pooled variance on the transformed scale. Notice that Eq. (13) uses $1/\bar{v}$ instead of the harmonic mean.

In summary, raw proportions are adequate when the observed proportions from primary studies fall between 0.2 and 0.8. When observed proportions are less than 0.2 or greater than 0.8, the logit or double arcsine transformation is recommended. It is worth noting that some simulation studies have shown that the double arcsine method slightly outperforms the logit transformation in terms of relative bias, mean squared error, and 95% coverage (Barendregt et al., 2013; Xu et al., 2021). Furthermore, the double arcsine method would be a more appropriate choice when extreme proportions need to be addressed. Last but not least, we recommend Eq. (13) when applying the back-transformation of the double arcsine method.

4.3 Calculating the summary effect size in R

In a meta-analysis, effect sizes are weighted by the inverse of their sampling variances, giving greater weight to larger studies and allowing their effect sizes to have a greater impact on the overall mean. The weighted average proportion (i.e., the summary proportion) can be computed as follows (Barendregt et al., 2013):

$$ES_P = P = \frac{\sum (w_i p_i)}{\sum w_i} = \frac{\sum \frac{p_i}{Var_{p_i}}}{\sum \frac{1}{Var_{p_i}}} \tag{14}$$

with its sampling error:

$$SE_p = \sqrt{\sum w_i} = \sqrt{\sum \frac{1}{Var_{p_i}}}. \tag{15}$$

The confidence interval of the weighted average proportion can be expressed as follows:

$$\begin{aligned} P_L &= P - Z_{(1-\alpha)}\,(SE_P) \\ P_U &= P + Z_{(1-\alpha)}\,(SE_P) \end{aligned} \tag{16}$$

where $Z_{(1-\alpha)} = 1.96$ when $\alpha = 0.05$.

We will now proceed with the first step of our meta-analysis. First, readers need to install and download the necessary R packages. These packages are developed to run within R and contain a collection of functions that are essential for conducting meta-analyses. In this tutorial, we will install two packages: *metafor* (Viechtbauer, 2010) and *meta* (Schwarzer et al., 2015). We will primarily rely on *metafor* and use *meta* to create forest plots. To install these packages, execute the following command:

```
install.packages(c("metafor", "meta"))
```

Once readers have installed a package, it becomes permanently available for use in R on this specific computer. To use the installed packages, one needs to

execute the *library()* function each time you run R. To load *metafor* and *meta* into the current R session, type the following R code:

```
library(metafor)
library(meta)
```

We then need to import data.csv into R and create a data frame named "dat". This can be achieved by using the read.csv() function and running the following code:

```
dat <- read.csv("data.csv", header = TRUE, sep = ",")
```

The code above represents a standard approach to importing .csv files. It instructs R to read a .csv file, interpreting the first row as column names, and recognizing commas as the separators between values.

To estimate the weighted average proportion, we will use the following functions in *metafor*: *escalc()*, *rma()*, and *predict()*. These functions, in conjunction with a range of arguments to be specified within them, provide instructions to R on how to calculate effect sizes. Note that certain arguments have default values, such as *weighted = TRUE*, so users don't need to specify them. The *escalc()* function estimates an effect size and its standard error for every primary study included in a meta-analysis. Users have the flexibility to decide whether to transform these effect sizes and, if so, which transformation method to employ, by using the *measure* argument. We will now create a data frame named "ies" (short for individual effect size) to store calculated effect sizes and standard errors using the following generic code:

```
#Only choose one of the three transformation methods
ies <- escalc(xi = cases, ni = total, data = dat,
   measure = "PR")
```

Here, the variable "cases" contains the number of events. The variable "total" contains the sample size. We use the argument *data* to inform R that these variables are contained in the data frame "dat". By using the argument *measure*, we can specify which computational method to employ for transforming the raw proportions:

```
measure = "PR" #No transformation
measure = "PLO" #The logit transformation
measure = "PFT" #The double arcsine transformation
```

We will then use the function *rma()* to pool the derived effect sizes. The function will yield a summary proportion, its standard error, and a 95% confidence interval. Additionally, it will also conduct heterogeneity tests. We can execute the following code to achieve this:

```
pes <- rma(yi, vi, data = ies, method = "REML")
```

Although naming an object in R is arbitrary, we strongly recommend that readers assign meaningful names to objects. In this case, if we decide not to perform a transformation, we will name this object "pes", which stands for pooled effect size. If we decide to perform a transformation with either the logit or the double arcsine, we will name it "pes.logit" or "pes.da", which stands for logit or double-arcsin transformed pooled effect size, respectively. The object will store all of the outcomes. The *method* argument dictates which of the following between-study variance estimators will be used (the default method is REML):

```
method = "DL" #The DL estimator
method = "REML" #The REML estimator
```

If unspecified, *rma()* estimates the variance component using the REML estimator. Even though *rma()* stands for random-effects meta-analysis, the function can perform a fixed-effect meta-analysis with the code:

```
method = "FE"
```

The object "pes.logit" or "pes.da" now contains the estimated transformed summary proportion. To convert it back to its original, non-transformed scale (i.e., proportion) and yield an estimate for the true summary proportion, we can use the *predict()* function:

```
#Inverse of logit transformation
pes <- predict(pes.logit, transf = transf.ilogit)
#Inverse of double arcsine transformation
pes <- predict(pes.da, transf = transf.ipft.hm, targ =
   list(ni = dat$total))
```

The argument *transf* dictates how to convert the transformed proportion back to proportion. As mentioned earlier, we can follow two methods for back-transformation (Eq. 10 or Eq. 13). In either case, we set the *transf* argument to *transf.ipft.hm* (the "hm" stands for the harmonic mean). If we opt for the harmonic mean ($n^{'}$) in Eq. (10) as the estimate for the total sample size, the sample sizes of primary studies are specified by setting the *targ* argument to *list(ni = dat$total)*. If we opt to use $1/\bar{v}$ as the total sample size estimate, then we specify the total sample size as $1/(pes.da\$se)^2$ within the *targ* argument and use the following code for back-transformation:

```
pes <- predict(pes.da, transf = transf.ipft.hm, targ =
   list(ni=1/(pes.da$se)^2))
```

Finally, to see the output for the estimated summary proportion and its 95% CI, we can use the *print()* function:

```
print(pes)
```

For the sake of readers' convenience, we provide readers with generic code for calculating the summary proportion under the random-effects model using three different transformation methods:

```
# Option 1: no transformation
   ies <- escalc(xi = cases, ni = total, data = dat,
      measure = "PR")
   pes <- rma(yi, vi, data = ies)
   print(pes)

# Option 2: the logit transformation
   ies.logit <- escalc(xi = cases, ni = total, data =
      dat, measure = "PLO")
   pes.logit <- rma(yi, vi, data = ies.logit)
   pes <- predict(pes.logit, transf = transf.ilogit)
   print(pes)

# Option 3: the double arcsine transformation
# targ can also be set to list(ni = 1/(pes.da$se)^2)
   ies.da <- escalc(xi = cases, ni = total, data =
      dat, measure = "PFT", add = 0)
   pes.da <- rma(yi, vi, data = ies.da)
   pes <- predict(pes.da, transf = transf.ipft.hm,
      targ = list(ni = dat$total))
   print(pes)
```

Note the use of *add = 0* in Option 3. When a study contains proportions equal to 0, the *escalc()* function will automatically add 0.5 to the observed data (i.e., the "cases" variable). Since the double arcsine transformation does not require any adjustments to be made to the data in such a situation, we can explicitly switch *add = 0.5* to *add = 0* to prevent the default adjustment.

Returning to the running example, we chose Option 2 (i.e., the logit transformation) to calculate the summary proportion because all of the observed proportions in the dataset are far below 0.2:

```
ies.logit <- escalc(xi = cases, ni = total, measure =
   "PLO", data = dat)
pes.logit <- rma(yi, vi, data = ies.logit, method =
   "DL", level = 95)
pes <- predict(pes.logit, transf = transf.ilogit)
print(pes, digits = 6)
```

The argument *digits* specifies the number of decimal places to which the printed results should be rounded, with the default value being 4. The argument *level* specifies the confidence interval, with the default value set to 95%.[4]

[4] In this particular case, the estimates of τ, τ^2, and I^2 will fall outside of the 95% CI for unknown reasons (though the summary proportion will not). The original authors did not discover this issue. One way to address this issue is by switching to the 99% CI. However, for the sake of consistency, we will continue to use the 95% CI throughout this tutorial.

The estimated summary proportion and its 95% CI are shown in Figure 1. Interpreting these summary statistics, we find that the summary proportion is estimated to be 0.000424 and its 95% CI is between 0.000316 and 0.000569.

```
    pred      ci.lb      ci.ub      cr.lb      cr.ub
0.000424   0.000316   0.000569   0.000133   0.001347
```

Figure 1. Summary proportion and its 95% CI

5 Quantification of heterogeneity

Meta-analysis aims to synthesize studies and estimate a more precise summary effect. An important decision that all meta-analysts face is whether it is appropriate to combine a set of identified studies in a meta-analysis, given the inevitable differences in their characteristics to varying degrees. Combining studies with substantially different effect estimates can result in an inaccurate summary effect and an unwarranted conclusion. For example, in a meta-analysis of proportions regarding re-offending rates among juvenile offenders in a city, the summary proportion may fall within a medium range (around 0.5). However, considerable variation exists among these proportions, with some studies conducted in certain boroughs reporting small proportions (e.g., under 0.1), while others report very large proportions (e.g., above 0.9). Simply reporting a moderately large mean proportion would be misleading, as it fails to acknowledge the significant variation or inconsistency in effect sizes across the studies. This variation is known as heterogeneity (Del Re, 2015). We will introduce three quantifying statistics for heterogeneity in this section: τ^2, Q, and I^2.

5.1 The between-study variance: τ^2

Heterogeneity can be quantified by dividing it into two distinct components: the between-study variance, which arises from the true variation among a body of studies, and the within-study variance, resulting from the sampling error. The true variation can be attributed to clinical and/or methodological diversity, in other words, the systematic differences between studies beyond what would be expected by chance, such as experimental designs, measurements, sample characteristics, interventions, study settings, and combinations thereof (Lijmer, Bossuyt, & Heisterkamp, 2002; Thompson & Higgins, 2002). In this tutorial, we focus on the true variation in effect sizes, namely the between-study heterogeneity.

We characterize between-study heterogeneity by the variance of the true effect size underlying the data, τ^2, a statistic called tau-squared. Under the assumption of normality, 95% of the true effects are expected to fall within $\pm$

$1.96 \times \tau$ of the point estimate of the summary effect size (Borenstein, Hedges, Higgins, & Rothstein, 2010). τ^2 reflects the total amount of systematic differences in effects across studies. The total variance of a study consists of the between- and within-study heterogeneity and is used to assign weights under the random-effects model (i.e., the inverse of the total variance).

In classic inverse variance meta-analysis, τ^2 can be estimated by numerous methods, as mentioned in Section 4 (e.g., REML, DL). Review and simulation studies have shown that both methods perform satisfactorily well across various situations; the differences between their results are negligible and rarely significant enough to impact the qualitative conclusions (e.g., Hamza et al., 2008; Thorlund et al., 2011; Veroniki et al., 2016). Nevertheless, it is advisable to obtain the 95% confidence interval around the point estimate of τ^2, especially when the number of included studies is small (less than 5) (Veroniki et al., 2016).

In practice, the DerSimonian and Laird estimator is arguably the most commonly used statistical method for meta-analyses of proportions and has become the conventional and default method for assessing the amount of between-study heterogeneity in many software packages, such as CMA (Cornell et al., 2014; Schwarzer et al., 2015). All estimations in this tutorial are based on the DL method.

5.2 Test of heterogeneity: Cochran's *Q*

Using formal tests, the presence of between-study heterogeneity is generally examined using a χ^2 test with a statistic Q (Cochran, 1954) under the null hypothesis that all studies share the same true effect (Hedges & Olkin, 1985). In other words, the Q-test and its p-value serve as a test of significance to address the null hypothesis: $H_0 : \tau^2 = 0$. If the value of the Q-statistic is above the critical χ^2 value, we will reject the null hypothesis and conclude that the effect sizes are heterogeneous. Under such circumstances, you may consider taking the random-effects model route. If Q does not exceed this value, then we fail to reject the null hypothesis.

It is important to exercise caution when interpreting a non-significant p-value and drawing the conclusion of homogeneous true effects. The statistical power of the Q-test heavily relies on the number of studies included in a meta-analysis, and as a result, it may fail to detect heterogeneity due to limited power when the number of included studies is small (less than 10) or when the included studies are of small size (Huedo-Medina, Snchez-Meca, Marn-Martnez, & Botella, 2006). Therefore, a non-significant result should not be taken as showing empirical evidence for homogeneity (Hardy & Thompson, 1998). This issue warrants serious attention, considering that a significant proportion of meta-analyses in Cochrane reviews involve only five or fewer studies (Davey, Turner, Clarke, & Higgins, 2011).

Furthermore, it is important to note that the Q-test, in addition to its aforementioned limitation, only assesses the viability of the null hypothesis and does not provide a quantification of the magnitude of the true heterogeneity in effect sizes (Card, 2015).

5.3 I^2 statistic

Higgins, Thompson, Deeks, and Altman (2003) proposed a statistic for measuring heterogeneity, denoted as I^2, that remains unaffected by the number of included studies. In essence, it reflects the ratio of the observed heterogeneity, representing the true between-study variance, to the total observed heterogeneity (i.e., the sum of between- and within-study variance). As a result, it facilitates the comparison of heterogeneity estimates across meta-analyses, regardless of the original scale used in the meta-analyses themselves.

I^2 can take values from 0% to 100%. A value of 0% indicates that all heterogeneity is caused by sampling error alone, requiring no further explanation. Conversely, when I^2 equals 100%, the entire heterogeneity can be attributed exclusively to genuine differences between studies, thus justifying the application of subgroup analyses or meta-regressions to identify potential moderating factors. The thresholds of 25%, 50%, and 75% are commonly used to indicate low, medium, and high heterogeneity, respectively (Higgins et al., 2003) . Note that these thresholds only serve as tentative benchmarks for I^2. The 95% CI around the I^2 statistic should also be calculated (Cuijpers, 2016; Ioannidis, Patsopoulos, & Evangelou, 2007).

Relying solely on the value of I^2 can be misleading because a 0% I^2, accompanied by a 95% CI ranging from 0% to 80%, does not necessarily indicate homogeneity in a small meta-analysis study. Rather, the degree of heterogeneity remains uncertain in such cases.

An important caveat

Together, the Q-statistic, τ^2, and I^2 can inform us if the effects are homogeneous, or consistent. When the effect sizes are reasonably consistent, it is appropriate to combine them and present a summary effect size in reports. In cases where moderate and substantial heterogeneity is present, the summary effect size becomes less informative or even of no value. In such cases, we strongly suggest that researchers conduct moderator analyses to thoroughly explore the possible sources of heterogeneity in observed effect sizes rather than relying solely on the mechanistic calculation of a single mean effect estimate (Egger, Schneider, & Smith, 1998). We will discuss moderator analysis in more detail later.

However, it is important to note that the methods used to estimate the amount of heterogeneity and conduct significance tests for heterogeneity are not always reliable, potentially leading to misleading interpretations of the variability of the true effect size. Relying solely on the Q-test is ill-advised due to its inadequate power to detect low heterogeneity (Chung, Rabe-Hesketh, & Choi, 2013; Rücker, Schwarzer, Carpenter, & Schumacher, 2008). Furthermore, the rules of thumb benchmarks for I^2 only hold true when the within-study error is relatively constant (Borenstein, Higgins, Hedges, & Rothstein, 2017). Underestimating between-study heterogeneity or failing to detect any heterogeneity due to inadequate statistical power can result in authors fitting the wrong model (i.e., the fixed-effect model),

leading to inaccurate inferences about the overall effect (Higgins & Thompson, 2002; Thompson, 1994; Thompson & Sharp, 1999).

Heterogeneity tests provide only a single piece of evidence when deciding between the fixed- and random-effects models. The choice of model should consider a range of factors, including the sampling frame, the desired type of inference, expectations about the distribution of the true effect, and the statistical significance of the heterogeneity tests, among others. Borenstein (2019) suggested that when studies in a meta-analysis are collected from the literature, a random-effects model is almost always preferable. This is because the true effect size is likely to vary across studies unless they were conducted by the same lab, following identical protocols, and using consistent materials on the same population. Furthermore, if we intend to make an inference to comparable populations, as is common in social sciences, the random-effects model becomes the only appropriate choice.

5.4 Viewing results of the heterogeneity test and statistics in R

To view the results of the heterogeneity test (Cochran's Q) and the estimates of between-study variance (τ^2) and I^2, we still use the *print()* function:

```
# Note, if you selected other transformation methods,
# then type pes.logit or pes.da in print()
print(pes)
```

The *confint()* function computes and displays the confidence intervals for τ^2 and I^2:

```
# If you selected other transformation methods,
# then type pes.logit or pes.da in confint()
confint(pes)
```

To display the output of heterogeneity-related results for the running example, we can type:

```
print(pes.logit, digits = 4)
confint(pes.logit, digits = 4)
```

The output appears in Figure 2. It reveals that τ^2 is 0.3256 (95% CI = 0.3296, 1.4997), I^2 is 97.24% (95% CI = 97.28, 99.39), and the Q-statistic is 580.5387 (p <.001), all of which suggests high heterogeneity in the observed proportions.[5]

[5] Again, the values of τ, τ^2, and I^2 have fallen out their 95% CIs. Readers can fix this problem by switching to the 99% CI.

```
Random-Effects Model (k = 17; tau^2 estimator: DL)

tau^2 (estimated amount of total heterogeneity): 0.3256 (SE
    = 0.2033)
tau (square root of estimated tau^2 value):      0.5707
I^2 (total heterogeneity / total variability):   97.24%
H^2 (total variability / sampling variability):  36.28

Test for Heterogeneity:
Q(df = 16) = 580.5387, p-val < .0001

Model Results:

estimate      se      zval    pval    ci.lb    ci.ub
 -7.7650  0.1502  -51.7147  <.0001  -8.0593  -7.4707  ***

---
Signif. codes:  0 ***  0.001  **  0.01  *  0.05  .  0.1  1

       estimate    ci.lb     ci.ub
tau^2    0.3256   0.3296    1.4997
tau      0.5707   0.5741    1.2246
I^2(%)  97.2439  97.2758   99.3884
H^2     36.2837  36.7079  163.4972
```

Figure 2. A random-effects model analysis of heterogeneity

6 Visualization of heterogeneity

This section is dedicated to visualization tools and a few formal diagnostic tests pivotal for heterogeneity analyses. We introduce two essential tools for readers: the forest plot and the Baujat plot. The forest plot allows for a visual assessment of the homogeneity across studies, while the Baujat plot can pinpoint studies that exert a significant impact on the overall effect, heterogeneity, or both. It's crucial to introduce the forest plot at this point. It lays the foundation for our in-depth demonstration of its application in subgroup analyses, which we will discuss in Section 7.

6.1 Forest plots

A forest plot (as shown in Figure 3) is a graphical representation that effectively displays the point estimates of study effects along with their corresponding confidence intervals (Lewis & Clarke, 2001). It is composed of a vertical reference line, an x-axis, and graphical representations of effect size estimates and their 95% CIs. The x-axis of the forest plot represents the scale of the outcome measure (in our case, the proportion) and can range from 0 to 1.

Typically, the vertical reference line is positioned at the point estimate of the pooled proportion. At the bottom of the reference line lies a colored diamond shape with its length representing the 95% confidence interval of the pooled proportion. Each study effect plotted in a forest plot consists of two components: a colored square symbolizing the point estimate of the study effect size and a horizontal line through the square representing the confidence interval around the point estimate. I refer to the horizontal lines as the squares' "wings", if you will.

The size of a square corresponds to the study's weight; a larger square signifies a larger sample size and, therefore, a greater weight. An effect size with a greater weight carries more influence on the summary effect size and is therefore depicted by a larger square with a shorter horizontal line (Anzures-Cabrera & Higgins, 2010).

In a forest plot, study effects are determined as homogeneous if all the horizontal lines of the squares overlap (Petrie, Bulman, & Osborn, 2003; Ried, 2006). The forest plot also allows us to identify potential outliers. This can be achieved by examining studies whose 95% confidence intervals do not overlap with the confidence interval of the summary effect size (Harrer, Cuijpers, A, & Ebert, 2021). Furthermore, it is worth noting that if large studies are identified as outliers, it may suggest that the overall heterogeneity is high.

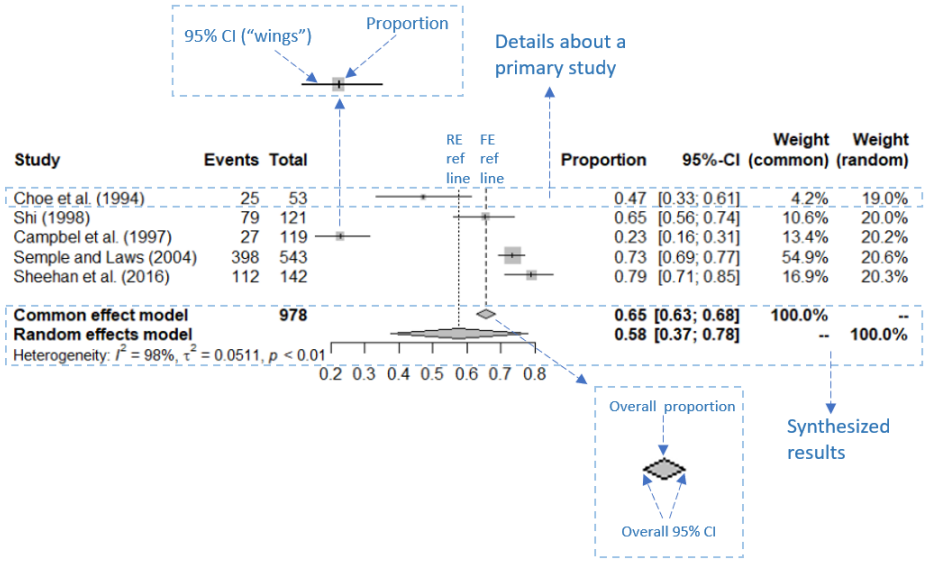

Figure 3. An anatomy of a basic forest plot

6.2 Creating forest plots in R

In this section, we will begin by explaining how to create a basic forest plot using the *meta* package. We will also show readers how to create a more sophisticated, publication-ready forest plot.

We can create a simple forest plot using the following generic code (assuming that we have loaded the *meta* package):

```
pes.summary <- metaprop(cases, total, authoryear, data
   = dat, sm = "PRAW")
forest(pes.summary)
```

Using the *metaprop()* function, we conduct a meta-analysis of proportions and save the results in an object named "pes.summary". We then feed these results into the *forest()* function to automatically generate a forest plot. The *sm* argument in the *metaprop()* function dictates which transformation method will be used to convert the original proportions:

```
PRAW # no transformation
PLO  # the logit transformation
PFT  # the double arcsine transformation
```

Forest plots created by the generic code are bare-boned and often fail to meet publishing standards. The following code can produce publication-quality forest plots for the running example:

```
pes.summary <- metaprop(cases, total, authoryear, data
   = dat, sm = "PLO", method.tau = "DL", method.ci =
   "NAsm")
forest(pes.summary,
       common = FALSE,
       print.tau2 = TRUE,
       print.Q = TRUE,
       print.pval.Q = TRUE,
       print.I2 = TRUE,
       rightcols = FALSE,
       pooled.totals = FALSE,
       weight.study = "random",
       leftcols = c("studlab", "event", "n", "effect",
          "ci"),
       leftlabs = c("Study", "Cases", "Total",
          "Prevalence", "95% C.I."),
       xlab = "Prevalence of CC (%)",
       smlab = "",
       xlim = c(0,4),
       pscale = 1000,
       squaresize = 0.5,
       fs.hetstat = 10,
```

```
        digits = 2,
        col.square = "navy",
        col.square.lines = "navy",
        col.diamond = "maroon",
        col.diamond.lines = "maroon")
```

The generated forest plot is shown in Figure 4.

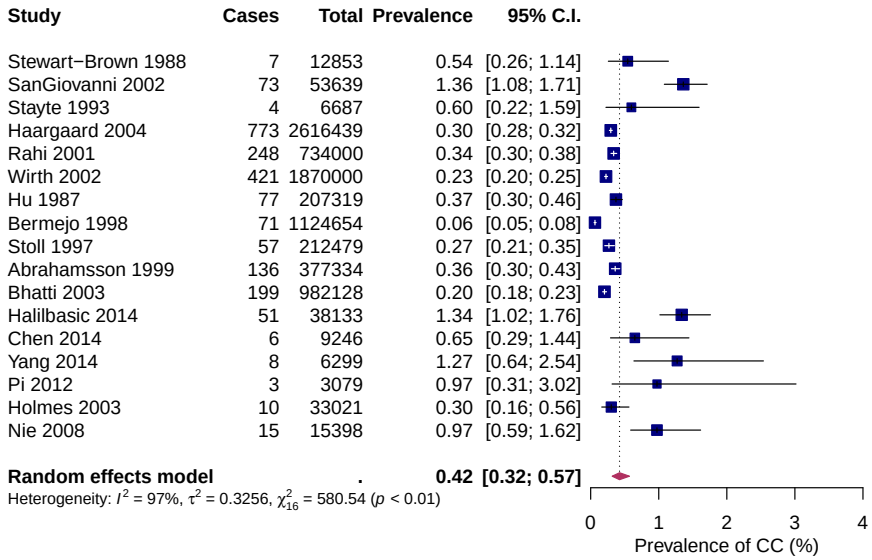

Figure 4. A publication-quality forest plot

The arguments in *forest()* provided above are mostly self-explanatory. They determine which components of the forest plot are displayed, as well as their colors, sizes, and positions on the graph. The *pscale* argument is particularly noteworthy. Setting "pscale = 1000" means that the prevalence is expressed as events per 1,000 observations. Consequently, the combined proportion under the random-effects model is displayed as 0.42‰ in the forest plot[6]. It should be mentioned that due to space constraints, we have only listed the most essential arguments in the *forest()* function. Readers are encouraged to refer to the documentation that comes with the *meta* package (type ?meta::forest() in R) to explore additional useful arguments for customizing their own forest plots.

[6] Readers should note that showing the permille symbol (‰) within code snippets in LaTeX can be challenging. Consequently, the "%" is used in the *xlab* argument purely for illustrative purposes. For accurate representation, readers can substitute the "%" with "‰" in R.

We can sort the individual studies by precision to help us visually inspect the data. This can be achieved by sorting the included studies using SE or the inverse of SE:

```
precision <- sqrt(ies.logit$vi)
```

We then add the *sortvar* argument in the *forest()* function:

```
sortvar = precision
```

The new forest plot is shown in Figure 5. This forest plot clearly shows that the prevalence of CC is higher in smaller studies (those with longer "wings"). In meta-analyses of comparative studies, a forest plot without indications of publication bias will exhibit an even spread of studies with varying precision on both sides of the mean effect size. However, in a meta-analysis of observational data, an uneven spread of studies may actually reflect a genuine pattern in effect sizes rather than publication bias, especially when small studies fall to the right side of the mean. It is also possible that some small studies are not published due to valid reasons, such as the use of inadequate research methods. Thus, this uneven distribution of effects warrants further investigation as it may provide new insights into the topic of interest.

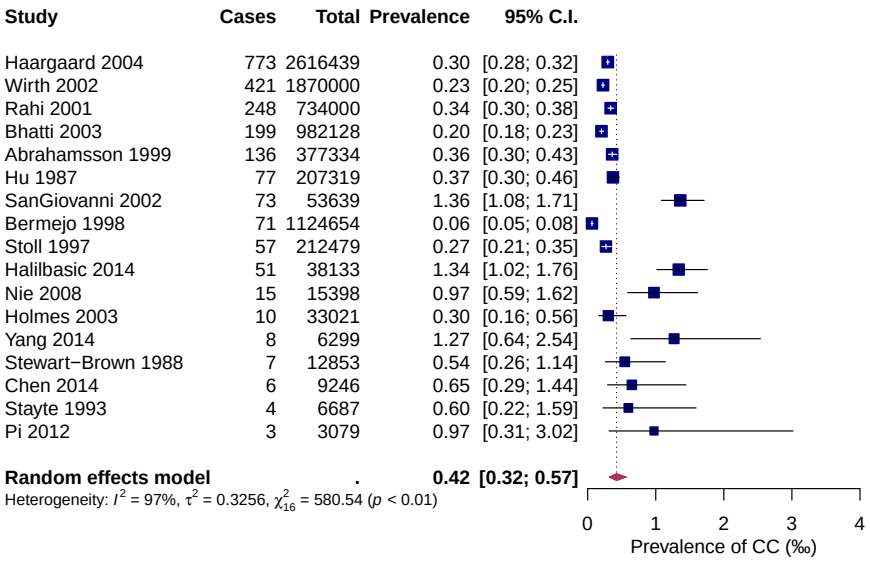

Figure 5. A forest plot with sorted studies by precision

A visual inspection of the forest plot identifies several potential outlying studies, including Wirth (2002), Bhatti (2003), SanGiovanni (2002), Bermejo

(1998), Halilbasic (2014), Nie (2008), and Yang (2014). Their 95% CIs do not overlap with that of the summary proportion. In the next step, we will cross-validate these potential outliers using the Baujat plot.

6.3 Identifying outlying and influential studies with diagnostic tools

When dealing with high between-study heterogeneity in a meta-analysis, one approach is to identify and exclude outliers, and then reassess the robustness of the summary effect size. In this section, we will introduce some diagnostic tools that can identify outlying and influential studies.

A basic Baujat plot is depicted in Figure 6. The horizontal axis of the Baujat plot quantifies each study's contribution to the overall heterogeneity or the Cochran Q-test, while the vertical axis measures the impact of each study on the summary effect size. We've divided the Baujat plot into four quadrants with light blue dotted lines for illustration purposes. Studies situated far to the right on the horizontal axis (in Quadrants 2 and 3) are significant contributors to heterogeneity. Those positioned far up on the vertical axis (in Quadrants 1 and 2) substantially influence the overall meta-analysis result. A study's influence is deemed substantial if its removal would lead to a drastically different overall effect.

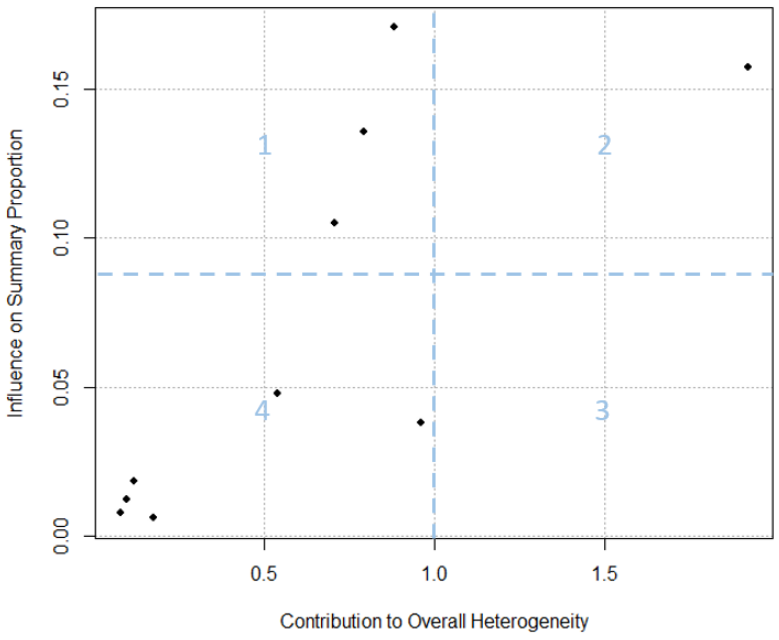

Figure 6. An anatomy of a basic Baujat plot

It can sometimes be challenging to differentiate between the concepts of an "outlier" and an "influential effect size" in the context of meta-analysis. While an outlying effect size can often be influential, it isn't always so. Conversely, an effect size that is influential doesn't necessarily have to be an outlier (Harrer et al., 2021). The Baujat plot helps distinguish between outliers that are influential and those that are not:

- Small studies with effect sizes similar to others typically fall into the lower left corner of Quadrant 4, indicating they are neither outliers nor influential.
- Small studies with notably different effect sizes than others often appear in the lower right corner of Quadrant 3. They may be outliers, but their small sample sizes prevent them from heavily impacting the overall effect size.
- Large studies with effect sizes similar to the majority of effect sizes tend to populate the upper left corner of Quadrant 1. While these studies have influential effects, they may not be outliers. Their influence on the pooled effect size is pronounced because of their extensive sample sizes.
- Large studies with dramatically different effect sizes than the rest tend to appear in the upper right corner of Quadrant 2. These studies are influential outliers, exerting the most substantial impact on both the overall effect and heterogeneity.

It is crucial to conduct several formal diagnostic tests to determine if the outlying effect sizes identified in the forest plot and Baujat plot are truly outliers. If deemed outliers, further investigation is required to determine their actual influence on the overall effect size. Viechtbauer and Cheung (2010) have proposed a set of case deletion diagnostics derived from linear regression analyses to identify influential studies, such as difference in fits values (DFFITS), Cook's distances, leave-one-out estimates for the amount of heterogeneity (i.e., τ^2) as well as the test statistic for heterogeneity (i.e., Q-statistic). In leave-one-out analyses, each study is removed sequentially, and the summary proportion is re-estimated based on the remaining n-1 studies. This approach allows for the assessment of each study's influence on the summary proportion.

Outlying effect sizes can also be identified by screening for externally studentized residuals exceeding an absolute value of 2 or 3 (Tabachnick, Fidell, & Osterlind, 2013; Viechtbauer & Cheung, 2010).

As a final note, instead of simply removing outlying effect sizes, meta-analysts should investigate these outliers and influential cases to understand their occurrence. They sometimes reveal valuable study characteristics that may serve as potential moderating variables.

6.4 Identifying outlying and influential studies in R

In this section, we will use the Baujat plot and diagnostic tests introduced above to detect outliers and influential studies. The generic code for Baujat plot is provided below:

```
baujat(pes) # or pes.logit, pes.da
```

For the running example, use the following code to create a customized Baujat plot:

```
# Create a Baujat plot
bjplot <- baujat(pes.logit,
                 symbol=19,
                 xlim=c(0,15),
                 xlab="Contribution to Overall
                    Heterogeneity",
                 ylab="Influence on Summary
                    Proportion")
# Label those studies located in the upper quadrants
bjplot <- bjplot[bjplot$x >= 10 | bjplot$y >= 0.4,]
text(bjplot$x, bjplot$y, bjplot$slab, pos=1)
```

The generated plot can be seen in Figure 7. In this customized Baujat plot, we have labeled only a few of the more "extreme" studies, specifically: SanGiovanni (2002) (Study 2), Bermejo (1998) (Study 8), and Halilbasic (2014) (Study12). We observe that both Study 2 and Study 12 may be considered influential, though they might not contribute heavily to the overall heterogeneity. In contrast, Study 8 stands out as an influential outlier, as it has a large impact on both the pooled proportion and heterogeneity.

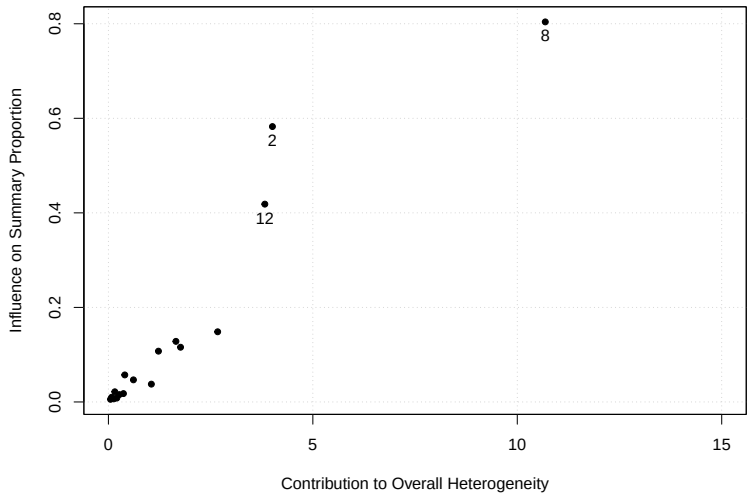

Figure 7. A basic Baujat plot

Next, we screen for large externally studentized residuals (ESR). The code below calculates the ESR for each study in the current dataset, then sorts them in descending order based on the absolute values of the z-scores tied to their respective ESRs:

```
# Calculate ESR
stud.res <- rstudent(pes.logit) # or pes, pes.da
# Sort ESR by z-values in descending order
abs.z <- abs(stud.res$z)
stud.res[order(-abs.z)]
```

The test outcome appears in Figure 8. The key here is to locate studies with z-values that exceed an absolute value of 2 or 3. Since we only have 17 studies in the running example, we will set the threshold at 2. Therefore, the second, eighth, and twelfth studies are chosen. They match the studies we previously identified through the Baujat plot.

```
     resid     se       z
8  -2.0265 0.5183 -3.9101
2   1.2701 0.5183  2.4505
12  1.2415 0.5541  2.2407
14  1.1563 0.6831  1.6928
17  0.8840 0.6382  1.3853
11 -0.7967 0.6198 -1.2854
6  -0.6895 0.6576 -1.0485
15  0.8618 0.8254  1.0441
9  -0.4925 0.6177 -0.7973
13  0.4459 0.7182  0.6209
4  -0.4063 0.7250 -0.5604
16 -0.3563 0.6727 -0.5297
3   0.3579 0.7743  0.4622
5  -0.2520 0.6444 -0.3911
1   0.2627 0.7021  0.3741
10 -0.1790 0.6231 -0.2872
7  -0.1447 0.6162 -0.2348
```

Figure 8. Externally studentized residuals results

The following code performs a set of leave-one-out diagnostic tests:

```
# Option 1: no transformation
# L1O stands for leave-one-out
L1O <- leave1out(pes); print(L1O)
# Option 2: the logit transformation
L1O <- leave1out(pes.logit, transf = transf.ilogit)
print(L1O)
```

```
# Option 3: the double arcsine transformation
# targ can also be set to list(ni = 1/(pes.da$se)^2)
L1O <- leave1out(pes.da, transf = transf.ipft.hm, targ
   = list(ni = dat$total))
print(L1O)
```

Using the current data set, we execute the following code:

```
L1O <- leave1out(pes.logit, transf = transf.ilogit)
print(L1O, digits = 6)
```

The output is shown in Figure 9. The numbers in the first column are the leave-one-out estimates for the summary proportion, which are derived by excluding one study at a time from the included studies. For instance, the first estimate in this column (i.e., 0.000419) is the summary proportion estimate when the first study in the included studies is removed.

	estimate	zval	pval	ci.lb	ci.ub	Q	Qp	tau2	I2	H2
1	0.000419	-50.492057	0.000000	0.000310	0.000566	577.938615	0.000000	0.326294	97.404569	38.529241
2	0.000383	-58.124097	0.000000	0.000293	0.000499	405.001830	0.000000	0.236593	96.296313	27.000122
3	0.000418	-50.760057	0.000000	0.000310	0.000565	578.562279	0.000000	0.325980	97.407366	38.570819
4	0.000443	-41.417189	0.000000	0.000308	0.000639	580.526132	0.000000	0.489631	97.416137	38.701742
5	0.000435	-46.319695	0.000000	0.000313	0.000603	575.730710	0.000000	0.383340	97.394615	38.382047
6	0.000449	-45.217145	0.000000	0.000321	0.000626	540.974670	0.000000	0.400959	97.227227	36.064978
7	0.000429	-48.854385	0.000000	0.000315	0.000586	576.473491	0.000000	0.341576	97.397972	38.431566
8	0.000479	-56.505027	0.000000	0.000367	0.000624	404.914535	0.000000	0.236229	96.295515	26.994302
9	0.000439	-48.899481	0.000000	0.000322	0.000598	579.956198	0.000000	0.338978	97.413598	38.663747
10	0.000431	-47.992385	0.000000	0.000314	0.000591	574.985048	0.000000	0.354815	97.391237	38.332337
11	0.000449	-47.824077	0.000000	0.000328	0.000616	548.816035	0.000000	0.353117	97.266844	36.587736
12	0.000387	-55.147300	0.000000	0.000293	0.000511	461.941616	0.000000	0.267048	96.752836	30.796108
13	0.000416	-50.664580	0.000000	0.000308	0.000562	576.843109	0.000000	0.325434	97.399640	38.456207
14	0.000400	-51.312960	0.000000	0.000297	0.000539	563.535902	0.000000	0.318164	97.338235	37.569060
15	0.000412	-51.101371	0.000000	0.000305	0.000555	576.283345	0.000000	0.324438	97.397114	38.418890
16	0.000432	-50.008941	0.000000	0.000319	0.000586	580.534075	0.000000	0.328479	97.416172	38.702272
17	0.000403	-51.136341	0.000000	0.000298	0.000543	559.149838	0.000000	0.317149	97.317356	37.276656

Figure 9. Results of leave-one-out diagnostic meta-analyses

A leave-one-out forest plot can visualize the change in the summary effect size. The generic code is given below:

```
# Option 1: no transformation
l1o <- leave1out(pes)
yi <- l1o$estimate; vi <- l1o$se^2
forest(yi,
       vi,
       slab = paste(dat$author, dat$year, sep = ","),
       refline = pes$b,
       xlab = "Leave-one-out summary proportions")

# Option 2: the logit transformation
l1o <- leave1out(pes.logit)
yi <- l1o$estimate; vi <- l1o$se^2
```

```
forest(yi,
       vi,
       transf = transf.ilogit,
       slab = paste(dat$author, dat$year, sep = ","),
       refline = pes$pred,
       xlab = "Leave-one-out summary proportions")

# Option 3: the double arcsine transformation
# targ can also be set to list(ni = 1/(pes.da$se)^2)
l1o <- leave1out(pes.da)
yi <- l1o$estimate; vi <- l1o$se^2
forest(yi,
       vi,
       transf = transf.ipft.hm,
       targ = list(ni = dat$total),
       slab = paste(dat$author, dat$year, sep = ","),
       refline = pes$pred,
       xlab = "Leave-one-out summary proportions")
```

To generate a customized leave-one-out forest plot for the current data set, use the following code:

```
l1o=leave1out(pes.logit)
yi=l1o$estimate; vi=l1o$se^2
forest(yi,
       vi,
       transf=transf.ilogit,
       slab=paste(dat$author,dat$year,sep=", "),
       xlab="Leave-one-out summary proportions",
       refline=pes$pred,
       digits=6)
abline(h=0.1)
```

The generated forest plot is shown in Figure 10. Each black square represents a leave-one-out summary proportion. The reference line indicates where the original summary proportion lies. The further a box deviates from the reference line, the more pronounced the impact of the corresponding excluded study will be on the original summary proportion. For instance, if we exclude the study by SanGiovanni et al. (2002), the new summary proportion becomes 0.00038. If we exclude Stayte et al. (1993), the new summary proportion becomes 0.000418. Apparently, excluding the former study has a larger impact on the original summary proportion than the latter study.

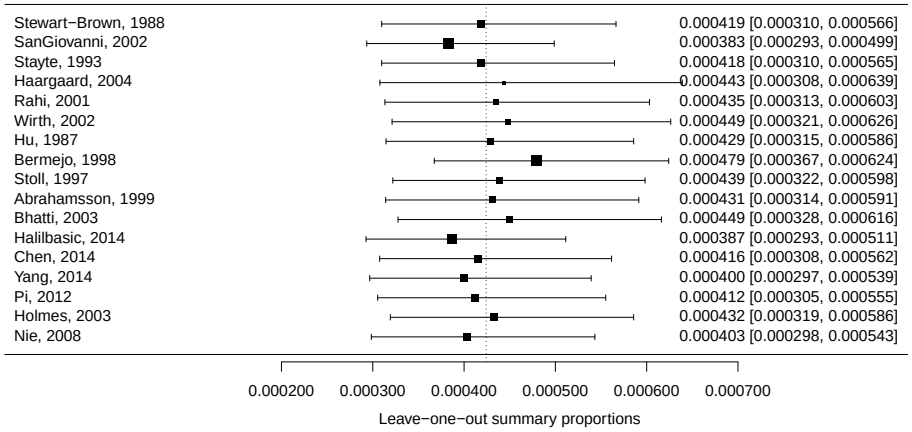

Figure 10. A leave-one-out forest plot

With these potential influential studies in mind, we now conduct a few more leave-one-out diagnostics with the *influence()* function in *metafor* to verify our guesses:

```
inf <- influence(pes.logit)
print(inf, digits=3)
plot(inf)
```

In Figure 11, studies marked with an asterisk are potential influential studies:

	rstudent	dffits	cook.d	cov.r	tau2.del	QE.del	hat	weight	dfbs	inf
1	0.374	0.083	0.007	1.052	0.326	577.939	0.048	4.811	0.083	
2	2.451	0.801	0.474	0.813	0.237	405.002	0.066	6.643	0.791	*
3	0.462	0.093	0.009	1.042	0.326	578.562	0.039	3.915	0.093	
4	-0.560	-0.242	0.088	1.541	0.490	580.526	0.069	6.896	-0.247	
5	-0.391	-0.151	0.027	1.239	0.383	575.731	0.068	6.839	-0.152	
6	-1.049	-0.336	0.139	1.289	0.401	540.975	0.069	6.873	-0.339	
7	-0.235	-0.077	0.006	1.117	0.342	576.473	0.067	6.658	-0.077	
8	-3.910	-0.941	0.653	0.812	0.236	404.915	0.066	6.636	-0.929	*
9	-0.797	-0.223	0.052	1.109	0.339	579.956	0.066	6.569	-0.224	
10	-0.287	-0.103	0.011	1.156	0.355	574.985	0.068	6.770	-0.103	
11	-1.285	-0.369	0.148	1.152	0.353	548.816	0.068	6.818	-0.371	
12	2.241	0.674	0.377	0.900	0.267	461.942	0.065	6.530	0.669	
13	0.621	0.136	0.019	1.047	0.325	576.843	0.046	4.578	0.136	
14	1.693	0.395	0.153	1.031	0.318	563.536	0.050	5.001	0.395	
15	1.044	0.197	0.039	1.032	0.324	576.283	0.034	3.420	0.198	
16	-0.530	-0.128	0.017	1.064	0.328	580.534	0.053	5.296	-0.128	
17	1.385	0.350	0.120	1.036	0.317	559.150	0.057	5.746	0.350	

Figure 11. Results of the influential study test

The diagnostics plots in Figure 12 show that the second and eighth studies are colored in red, indicating that they fulfill the criteria as influential studies.

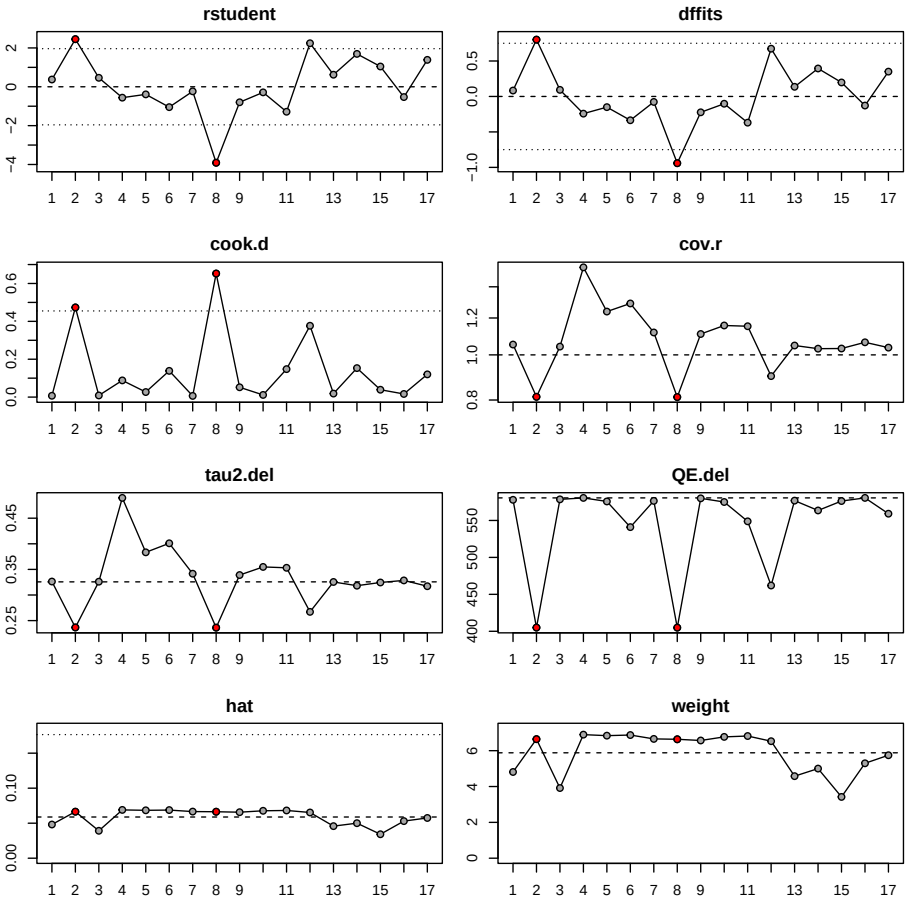

Figure 12. Influential study diagnostics

Based on the Baujat plot and the outcomes of the diagnostic tests, we determine that all three studies (Study 2, 8, and 12) can be considered outliers, but only Study 2 and 8 are deemed influential.

6.5 Removing outlying studies in R

Once all possible outliers are identified, we can remove them with the following generic code:

```
# Depending on the transformation method,
# measure = "PLO" or measure = "PFT"
# Remember to add "add = 0" when using the
# double arcsine transformation
ies.noutlier <- escalc(xi = cases, ni = total, measure
   = "PR", data = dat[-c(study1, study2,),])
```

If we were to exclude Study 2 and Study 8 in the current data set, we would execute the following code:

```
# Remove the two studies and calculate individual
# effect sizes
ies.logit.noutlier <- escalc(xi = cases, ni = total,
   measure = "PLO", data = dat[-c(2, 8),])
# Conduct meta-analysis with no outliers
pes.logit.noutlier <- rma(yi, vi, data =
   ies.logit.noutlier, method = "DL")
pes.noutlier <- predict(pes.logit.noutlier, transf =
   transf.ilogit)
print(pes.noutlier, digits = 5)
```

7 Explanation of heterogeneity with moderator analyses

We've determined that our data shows significant heterogeneity. Furthermore, we identified several outlying studies that notably impact both the overall effect and the variability of the observed effect sizes. When substantial heterogeneity remains even after excluding these outliers, one commonly employed strategy to unearth additional sources of heterogeneity is through moderator analyses. In fact, a thorough moderator analysis can often yield deeper insights than a mere estimate of summary effect size. This analysis helps identify and quantify the extent to which certain study-level characteristics contribute to the observed heterogeneity.

Subgroup analysis and meta-regression are two major forms of moderator analysis. Subgroup analysis can be seen as a special case of meta-regression, which examines the impact of a single categorical variable (Thompson & Higgins, 2002). In fact, meta-regression can accommodate both categorical and continuous moderators of desired numbers. For instance, a meta-regression can include a series of continuous variables or a mix of both continuous and categorical variables. In this tutorial, our focus will be on subgroup analysis and meta-regression with a continuous moderator.

7.1 Meta-regression with a categorical moderator: Subgroup analysis

When we want to explain heterogeneity with a categorical moderator in a meta-analysis, subgroup analysis is the method of choice. This approach mirrors the

logic of ANOVA in primary research (Littell, Corcoran, & Pillai, 2008). In a subgroup analysis, studies are partitioned into two or more subgroups according to the categories within the moderator. This moderator represents a specific study characteristic that can potentially explain a portion of the variability observed between studies (Hamza et al., 2008). If a subgroup has a unique characteristic absent in other subgroups (e.g., exposure to a new treatment vs. an old treatment), and the effect sizes between the subgroups show significant differences, it suggests that the variation in effect sizes (i.e., the true heterogeneity) can be attributed to this unique characteristic. In essence, the purpose of subgroup analysis is to ascertain if the chosen moderator accounts for a significant portion of the true heterogeneity.

To evaluate the influence of a proposed moderator, we apply a weighted least squares (WLS) regression. In this approach, effect sizes (e.g., those transformed using logit or double arcsine methods) are regressed against the moderator (Harrer et al., 2021):

$$ES_i = \beta_0 + \beta_1 C + \delta_i + e_i \tag{17}$$

where ES_i is the observed effect size for the primary study i, C is the dummy variable representing the moderator (or predictor), β_1 is the regression coefficient (or slope), and β_0 is the model intercept. δ_i and e_i are error terms. Specifically, δ_i is the between-study error for the primary study i, with its variance being the between-study variance, τ^2; e_i is the sampling error for the primary study i, with its variance being the within-study variance. The goal of the meta-regression model is to estimate the parameters, β_0 and β_1.

The categorical moderator is introduced in the analysis through dummy coding (e.g., the "studesg" variable in our data set). Let's say we have two categories within this predictor: Subgroup A and Subgroup B. If Subgroup A is chosen as the reference group, then all primary studies in Subgroup A would be coded as 0, while those in Subgroup B would be coded as 1. Mathematically, this can be represented as $C = 0$ for Subgroup A and $C = 1$ for Subgroup B. The regression coefficient of C, β_1, quantifies the effect size difference between the two subgroups. When $C = 0$, β_0 becomes the true overall effect of Subgroup A. When $C = 1$, the overall effect of Subgroup B is captured by the sum β_0 and β_1. In summary, the observed effect size for the study i, ES_i, is an estimator of the study's true effect size, $\beta_0 + \beta_1 C + \delta_i$, burdened by the sampling error, e_i.

Eq. (17) is a mixed-effects meta-regression model, a standard choice for meta-regression. In subgroup analyses, this model combines the study effects within each subgroup using a random-effects model, while a fixed-effect model is used to combine subgroups and yield the overall effect (Borenstein et al., 2009). A Wald-type test is used in meta-regression to determine if the slope of the model is statistically significant, using the Z-score. In subgroup analyses, a statistically significant slope suggests that Subgroups A and B exhibit statistically significant differences between their overall effect sizes. In other words, the subgroup membership can explain some or all of the between-study heterogeneity. Another method to assess a moderator's impact in meta-regression is through Cochran's Q. In subgroup analyses, if the Q-statistic for the predictor is statistically sig-

nificant, it means that the subgroup membership explains some or the entirety of the variability observed in the effect sizes. The R^2 index can be employed in meta-regression to quantify the proportion of the true heterogeneity across all studies (i.e., the between-study heterogeneity) that can be accounted for by moderators.

7.2 Meta-regression with a continuous moderator

In a meta-regression model with a single continuous moderator, as shown in Eq. (18) (Harrer et al., 2021),

$$ES_i = \beta_0 + \beta_1 x_i + \delta_i + e_i \tag{18}$$

x_i represent a continuous moderator, β_1 is the regression slope. δ_i and e_i are the between- and within-study error terms for the study i, respectively. β_0 is still the model intercept, but it now represents the overall true effect size when $x = 0$. In summary, ES_i represents the observed effect size for the study i, which is an estimator of the study's true effect size, $\beta_0 + \beta_1 x_i + \delta_i$, burdened by the sampling error, e_i.

As summarized by Harrer et al. (2021), meta-regression analyzes the relationship between predictors and observed effects to identify a consistent pattern between them, in the form of a regression line. By accounting for both sampling error and between-study differences, meta-regression seeks to fit a model that can generalize across all possible studies relevant to the topic. A well-fitting meta-regression model can predict effect sizes close to the observed data.

An important caveat

Moderator analysis is subject to several limitations that should be taken into consideration. A primary issue is that both the subgroup analysis and meta-regression require a large ratio of studies to moderators. It is generally recommended that moderator analysis should only be conducted when there are at least 10 studies available for each moderator included in the analysis. This is particularly crucial in multivariate models where the number of studies might be small, leading to reduced statistical power (Higgins & Green, 2006; Littell et al., 2008).

Another significant limitation is that the significant differences observed between subgroups of studies cannot be seen as causal evidence. We may fail to identify moderators that are truly responsible for the heterogeneity in effect sizes. Consequently, causal conclusions cannot be drawn solely from moderator analyses (Cuijpers, 2016; Littell et al., 2008). We strongly recommend that researchers select moderators based on solid theoretical reasoning and only test those moderators with a strong theoretical basis. This approach helps prevent erroneously attributing heterogeneity to spurious moderators (Schmidt & Hunter, 2014).

7.3 Conducting subgroup analyses and recalculating the overall summary proportion in R

In a mixed-effects model meta-regression, the summary effect size for each subgroup is computed using a random-effects model. Instead of estimating τ^2 across all studies, it's estimated within these subgroups. In other words, each subgroup has its own estimated τ^2. These τ^2 estimates may vary across subgroups. We can choose to pool them or keep them separate when we compute the overall and within-subgroup summary proportions, depending on our assumptions (Borenstein et al., 2009).

If we attribute the differences in these observed within-group τ^2 estimates solely to sampling error, then we anticipate a common τ^2 across subgroups. In such a scenario, pooling a common τ^2 estimate and applying it universally to all studies is appropriate. Conversely, if systematic factors, beyond just sampling errors, are believed to influence the varying values of the observed within-group τ^2 estimates, then employing distinct τ^2 estimates for each subgroup is justified. Essentially, using a separate estimate for between-study variance is equal to conducting an independent meta-analysis for each subgroup. It's important to emphasize that the pooled proportion across all subgroups is likely to differ from the summary proportion derived from pooling across all studies without subgrouping. Nevertheless, any differences in these estimates are generally negligible.

When we assume that τ^2 is the same for all subgroups, we can use the R^2 index to represent the proportion of the between-study variance across all studies that can be explained by the subgroup membership (Borenstein et al., 2009).

We have developed the following generic code to help readers perform subgroup analyses and compute the overall and within-subgroup summary proportions. It is essential for readers to gain a thorough understanding of their data's characteristics to choose the appropriate computational option.

In the first situation, we do not assume a common between-study variance component across subgroups and thus do not pool within-group τ^2 estimates. In R, we first fit a random-effects model for each subgroup, and then we combine the estimated statistics into a data frame. In the next step, we fit a fixed-effect model to compare the two estimated logit transformed proportions and recalculate the summary proportion. The generic code is provided below:

```
# Assumption 1:
# Do not assume a common between-study variance
# component (not pooling within-group estimates of
# between-study variance)
# Option 1: no transformation
# Conduct a random-effects model meta-analsis for each
# subgroup defined by the moderator variable
pes.subgroup1 <- rma(yi, vi, data = ies, subset =
   moderator == "subgroup1")
pes.subgroup2  <- rma(yi, vi, data = ies, subset =
   moderator == "subgroup2")
```

```
# Create a dataframe to store effect size estimates,
# standard errors, heterogeneity for both subgroups
# Add an object named moderator to distinguish two
# subgroups. It will be used in the next step.
dat.diffvar  <- data.frame(estimate =
   c(pes.subgroup1$b, pes.subgroup2$b), stderror =
   c(pes.subgroup1$se, pes.subgroup2$se), moderator =
   c("subgroup1", "subgroup2"), tau2 =
   round(c(pes.subgroup1$tau2, pes.subgroup2$tau2),
   3))
# Fit a fixed-effect meta-regression to compare the
# subgroups
subganal.moderator  <- rma(estimate, sei = stderror,
   mods = ~ moderator, method = "FE", data =
   dat.diffvar)
# Recalculate summary effect size assuming different
# heterogeneity components
pes.moderator  <- rma(estimate, sei = stderror, method
   = "FE", data = dat.diffvar)
pes.moderator  <- predict(pes.moderator)
# Display subgroup 1 summary effect size
print(pes.subgroup1)
# Display subgroup 2 summary effect size
print(pes.subgroup2)
# Display subgroup analysis results
print(subganal.moderator)
# Display recomputed summary effect size
print(pes.moderator)

# Option 2: the logit transformation
# Conduct a random-effects model meta-analsis for each
# subgroup defined by the moderator variable
pes.logit.subgroup1  <- rma(yi, vi, data = ies.logit,
   subset = moderator == "subgroup1")
pes.logit.subgroup2  <- rma(yi, vi, data = ies.logit,
   subset = moderator == "subgroup2")
pes.subgroup1  <- predict(pes.logit.subgroup1, transf
   = transf.ilogit)
pes.subgroup2  <- predict(pes.logit.subgroup2, transf
   = transf.ilogit)
# Create a dataframe to store effect size estimates,
# standard errors, heterogeneity for both subgroups
# Add an object named moderator to distinguish two
# subgroups.
```

```
dat.diffvar <- data.frame(estimate =
   c(pes.logit.subgroup1$b, pes.logit.subgroup2$b),
   stderror = c(pes.logit.subgroup1$se,
   pes.logit.subgroup2$se), moderator =
   c("subgroup1", "subgroup2"), tau2 =
   round(c(pes.logit.subgroup1$tau2,
   pes.logit.subgroup2$tau2), 3))
# Fit a fixed-effect meta-regression to compare the
# subgroups
subganal.moderator <- rma(estimate, sei = stderror,
   mods = ~ moderator, method = "FE", data =
   dat.diffvar)
# Recalculate summary effect size assuming different
# heterogeneity components
pes.logit.moderator <- rma(estimate, sei = stderror,
   method = "FE", data = dat.diffvar)
pes.moderator <- predict(pes.logit.moderator, transf =
   transf.ilogit)
# Display subgroup 1 summary effect size
print(pes.subgroup1); print(pes.logit.subgroup1)
# Display subgroup 2 summary effect size
print(pes.subgroup2); print(pes.logit.subgroup2)
# Display subgroup analysis results
print(subganal.moderator)
# Display recomputed summary effect size
print(pes.moderator)

# Option 3: the double arcsine transformation
# Conduct a random-effects model meta-analsis for each
# subgroup defined by the moderator variable
# targ can also be set to list(ni = 1/(pes.da$se)^2)
pes.da.subgroup1 <- rma(yi,vi,data = ies.da, subset =
   moderator == "subgroup1")
pes.da.subgroup2 <- rma(yi,vi,data = ies.da, subset =
   moderator == "subgroup2")
pes.subgroup1 <- predict(pes.da.subgroup1, transf =
   transf.ipft.hm,targ = list(ni = dat$total))
pes.subgroup2 <- predict(pes.da.subgroup2, transf =
   transf.ipft.hm,targ = list(ni = dat$total))
# Create a dataframe to store effect size estimates,
# standard errors, heterogeneity for both subgroups
# Add an object named moderator to distinguish two
# subgroups.
dat.diffvar <- data.frame(estimate =
   c(pes.da.subgroup1$b, pes.da.subgroup2$b),
```

```
    stderror = c(pes.da.subgroup1$se,
    pes.da.subgroup2$se), moderator = c("subgroup1",
    "subgroup2"), tau2 =
    round(c(pes.da.subgroup1$tau2,
    pes.da.subgroup2$tau2), 3))
# Fit a fixed-effect meta-regression to compare the
# subgroups
subganal.moderator <- rma(estimate, sei = stderror,
    mods = ~ moderator, method = "FE", data =
    dat.diffvar)
# Recalculate summary effect size assuming different
# heterogeneity components
# targ can also be set to list(ni = 1/(pes.da$se)^2)
pes.da.moderator <- rma(estimate, sei = stderror,
    method = "FE", data = dat.diffvar)
pes.moderator <- predict(pes.da.moderator, transf =
    transf.ipft.hm, targ = list(ni = dat$total))
# Display subgroup 1 summary effect size
print(pes.subgroup1); print(pes.da.subgroup1)
# Display subgroup 2 summary effect size
print(pes.subgroup2); print(pes.da.subgroup2)
# Display subgroup analysis results
print(subganal.moderator)
# Display recomputed summary effect size
print(pes.moderator)
```

In the second situation, we assume a common between-study variance component across subgroups and pool within-group τ^2 estimates. Generally speaking, unless there is a substantial number of studies available within each subgroup (i.e., more than five studies) or compelling evidence suggesting within-group variances vary from one subgroup to the next, it is sufficient to calculate summary proportions and create forest plots with a pooled τ^2 (Borenstein et al. (2009)). In this case, we can directly use the *rma()* function and fit a mixed-effects model to evaluate the potential moderator. In R, we still need to combine the estimated statistics into a new data frame for us to calculate a new overall summary proportion using a pooled τ^2 across all studies.

```
# Assumption 2: Assume a common between-study variance
# component (pool within-group estimates of
# between-study variance)
# Option 1: no transformation
# Conduct moderator analysis
subganal.moderator <- rma(yi, vi, data = ies, mods = ~
    moderator)
pes.subg.moderator <- predict(subganal.moderator)
# Obtain estimates for each subgroup
```

```
pes.subgroup1 <- rma(yi, vi, data = ies, mods = ~
   moderator == "subgroup2")
pes.subgroup2 <- rma(yi, vi, data = ies, mods = ~
   moderator == "subgroup1")
# Create a dataframe to store effect size estimates,
# standard errors, heterogeneity for both subgroups
dat.samevar <- data.frame(estimate =
   c((pes.subgroup1$b)[1], (pes.subgroup1$b)[1]),
   stderror = c((pes.subgroup2$se)[1],
   (pes.subgroup2$se)[1]), tau2 =
   subganal.moderator$tau2)
# Recalculate summary effect size assuming a common
# heterogeneity component
pes.moderator <- rma(estimate, sei = stderror, method
   = "FE", data = dat.samevar)
pes.moderator <- predict(pes.moderator)
# Display subgroup 1 summary effect size
print(pes.subg.moderator[study label 1])
# Display subgroup 2 summary effect size
print(pes.subg.moderator[study label 2])
# Display subgroup analysis results
print(subganal.moderator)
# Display recomputed summary effect size
print(pes.moderator)

# Option 2: the logit transformation
# Conduct moderator analysis
subganal.moderator <- rma(yi, vi, data = ies.logit,
   mods = ~ moderator)
pes.subg.moderator <- predict(subganal.moderator,
   transf=transf.ilogit)
# Obtain estimates for each subgroup
pes.logit.subgroup1 <- rma(yi, vi, data = ies.logit,
   mods = ~ moderator == "subgroup2")
pes.logit.subgroup2 <- rma(yi, vi, data = ies.logit,
   mods =~ moderator == "subgroup1")
# Create a dataframe to store effect size estimates,
# standard errors, heterogeneity for both subgroups
dat.samevar <- data.frame(estimate =
   c((pes.logit.subgroup1$b)[1],(pes.logit.subgroup2$b)[1]),
   stderror =
   c((pes.logit.subgroup1$se)[1],(pes.logit.subgroup2$se)[1]),
   tau2 = subganal.moderator$tau2)
# Recalculate summary effect size assuming a common
# heterogeneity component
```

```
pes.logit.moderator <- rma(estimate, sei = stderror,
   method = "FE", data = dat.samevar)
pes.moderator <- predict(pes.logit.moderator, transf =
   transf.ilogit)
# Display subgroup 1 summary effect size
print(pes.subg.moderator[study lable 1])
# Display subgroup 2 summary effect size
print(pes.subg.moderator[study lable 2])
# Display subgroup analysis results
print(subganal.moderator)
# Display recomputed summary effect size
print(pes.moderator)

# Option 3: the double arcsine transformation
# Conduct moderator analysis
# targ can also be set to list(ni = 1/(pes.da$se)^2)
subganal.moderator <- rma(yi, vi, data = ies.da, mods
   = ~ moderator)
pes.subg.moderator <- predict(subganal.moderator,
   transf = transf.ipft.hm, targ = list(ni=dat$total))
# Obtain estimates for each subgroup
pes.da.subgroup1 <- rma(yi, vi, data = ies.da, mods =
   ~ moderator == "subgroup2")
pes.da.subgroup2 <- rma(yi, vi, data = ies.da, mods =
   ~ moderator == "subgroup1")
# Create a dataframe to store effect size estimates,
# standard errors, heterogeneity for both subgroups
dat.samevar <- data.frame(estimate =
   c((pes.da.subgroup1$b)[1],
   (pes.da.subgroup2$b)[1]), stderror =
   c((pes.da.subgroup1$se)[1],
   (pes.da.subgroup2$se)[1]), tau2 =
   subganal.moderator$tau2)
# Recalculate summary effect size assuming a common
# heterogeneity component
# targ can also be set to list(ni = 1/(pes.da$se)^2)
pes.da.moderator <- rma(estimate, sei = stderror,
   method = "FE", data = dat.samevar)
pes.moderator <- predict(pes.da.moderator, transf =
   transf.ipft.hm, targ = list(ni = dat$total))
# Display subgroup 1 summary effect size
print(pes.subg.moderator[study lable 1])
# Display subgroup 2 summary effect size
print(pes.subg.moderator[study lable 2])
# Display subgroup analysis results
```

```
print(subganal.moderator)
# Display recomputed summary effect size
print(pes.moderator)
```

To help readers better understand how to use the code templates, we will now illustrate their implementation with the running example. For demonstrative purposes, we will use the variable "study design" (Birth cohort vs. Others) as the moderator and conduct the analysis with the logit transformation under both assumptions.

In the first situation, we do not assume a common between-study variance component across subgroups:

```
# Conduct a random-effects model meta-analsis for each
# subgroup defined by the moderator studydesign
pes.logit.birthcohort <- rma(yi, vi, data=ies.logit,
    subset=studydesign == "Birth cohort", method="DL")
pes.logit.others <- rma(yi, vi, data=ies.logit,
    subset=studydesign == "Others", method = "DL")
pes.birthcohort <- predict(pes.logit.birthcohort,
    transf = transf.ilogit, digits = 5)
pes.others <- predict(pes.logit.others, transf =
    transf.ilogit, digits = 5)
# Create a dataframe to store effect size estimates,
# standard errors, heterogeneity for both subgroups
# Add an object named studydesign to distinguish two
# subgroups.
dat.diffvar <- data.frame(estimate =
    c(pes.logit.birthcohort$b, pes.logit.others$b),
    stderror = c(pes.logit.birthcohort$se,
    pes.logit.others$se), studydesign = c("Birth
    cohort", "Others"), tau2 =
    round(c(pes.logit.birthcohort$tau2,
    pes.logit.others$tau2), 3))
# Fit a fixed-effect meta-regression to compare the
# subgroups
subganal.studydesign <- rma(estimate, sei = stderror,
    data = dat.diffvar, mods = ~ studydesign, method =
    "FE")
# Recalculate summary effect size assuming different
# heterogeneity components
pes.logit.studydesign <- rma(estimate, sei = stderror,
    method = "FE", data = dat.diffvar)
pes.studydesign <- predict(pes.logit.studydesign,
    transf = transf.ilogit)
# Display summary effect sizes of the two subgroups
```

```
print(pes.birthcohort, digits = 6);
   print(pes.logit.birthcohort, digits = 3)
print(pes.others, digits = 6); print(pes.logit.others,
   digits = 3)
# Display subgroup analysis results
print(subganal.studydesign, digits = 3)
# Display recomputed summary effect size
print(pes.studydesign, digits = 6)
```

The outcomes of the subgroup analysis appear in Figure 13.

```
     pred    ci.lb    ci.ub    pi.lb    pi.ub
 0.000352 0.000158 0.000782 0.000045 0.002737


Random-Effects Model (k = 6; tau^2 estimator: DL)

tau^2 (estimated amount of total heterogeneity): 0.932 (SE = 0.866)
tau (square root of estimated tau^2 value):      0.966
I^2 (total heterogeneity / total variability):   98.55%
H^2 (total variability / sampling variability):  68.92

Test for Heterogeneity:
Q(df = 5) = 344.594, p-val < .001

Model Results:

estimate     se     zval   pval   ci.lb   ci.ub
  -7.952  0.408  -19.501  <.001  -8.752  -7.153  ***

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

     pred    ci.lb    ci.ub    pi.lb    pi.ub
 0.000472 0.000341 0.000653 0.000169 0.001317


Random-Effects Model (k = 11; tau^2 estimator: DL)

tau^2 (estimated amount of total heterogeneity): 0.247 (SE = 0.175)
tau (square root of estimated tau^2 value):      0.497
I^2 (total heterogeneity / total variability):   95.76%
H^2 (total variability / sampling variability):  23.59

Test for Heterogeneity:
Q(df = 10) = 235.944, p-val < .001

Model Results:

estimate     se     zval   pval   ci.lb   ci.ub
  -7.658  0.166  -46.161  <.001  -7.984  -7.333  ***

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

Fixed-Effects with Moderators Model (k = 2)

I^2 (residual heterogeneity / unaccounted variability): 0.00%
H^2 (unaccounted variability / sampling variability):   1.00
R^2 (amount of heterogeneity accounted for):            NA%

Test for Residual Heterogeneity:
QE(df = 0) = 0.000, p-val = 1.000
```

```
Test of Moderators (coefficient 2):
QM(df = 1) = 0.445, p-val = 0.505

Model Results:
                   estimate     se     zval   pval   ci.lb   ci.ub
intrcpt              -7.952  0.408  -19.501  <.001  -8.752  -7.153  ***
studydesignOthers     0.294  0.440    0.667  0.505  -0.569   1.157

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


     pred    ci.lb    ci.ub
 0.000453 0.000335 0.000611
```

Figure 13. A subgroup analysis assuming different between-study variance components

From the output above, we can derive that the summary effect estimates are 0.00035 (95% CI = 0.00016, 0.00078), 0.00047 (95% CI = 0.00034, 0.00065), and 0.00045 (95% CI = 0.00034, 0.00061) for the two subgroups and the overall group of studies, respectively. Note that the subgroup summary effect estimates are derived by taking the exponential of the model results (e.g., exp(-7.952) = 0.00035). When we fit separate random-effects models in the two subgroups, we decide to allow the amount of variance within each set of studies to be different, which results in two different within-group estimates of τ^2 (0.93 and 0.25 for studies using the birth cohort design and other study designs, respectively). In other words, studies within each subgroup share the same estimate of τ^2 .

The results reveal that the difference between the two subgroup summary estimates is not statistically significant ($QM(1) = 0.45$, $p = 0.51$). Note that the sum of the within-group heterogeneity across the subgroups in the fixed-effect model is equal to $QE(0) = 0$, $p = 1$. This is because the within-group heterogeneity has been accounted for in each subgroup ($Q(df = 5) = 344.594$, $p < 0.001$; $Q(df = 10) = 235.944$, $p < 0.01$, respectively) in the random-effects model, thus there is no heterogeneity left to be accounted for.

In the second situation where we assume a common between-study variance component across subgroups, execute the following code:

```
# Conduct a subgroup analysis based on studydesign
subganal.studydesign <- rma(yi, vi, data = ies.logit,
    mods = ~ studydesign, method = "DL")
pes.subg.studydesign <- predict(subganal.studydesign,
    transf = transf.ilogit)
# Obtain estimates for each subgroup
pes.logit.birthcohort <- rma(yi, vi, data = ies.logit,
    mods = ~ studydesign == "Others", method = "DL")
pes.logit.others = rma(yi, vi, data = ies.logit, mods
    = ~ studydesign == "Birth cohort", method = "DL")
```

```
# Create a dataframe to store effect size estimates,
# standard errors, heterogeneity for both subgroups
dat.samevar <- data.frame(estimate =
   c((pes.logit.birthcohort$b)[1],
   (pes.logit.others$b)[1]), stderror =
   c((pes.logit.birthcohort$se)[1],
   (pes.logit.others$se)[1]), tau2 =
   subganal.studydesign$tau2)
# Recalculate summary effect size assuming a common
# heterogeneity component
pes.logit.studydesign = rma(estimate, sei = stderror,
   method = "FE", data = dat.samevar)
pes.studydesign = predict(pes.logit.studydesign,
   transf = transf.ilogit)
# Display subgroup summary effect sizes
print(pes.subg.studydesign[1], digits = 6)
print(pes.subg.studydesign[17], digits = 6)
# Display subgroup analysis results
print(subganal.studydesign, digits = 4)
# Display recomputed summary effect size
print(pes.studydesign, digits = 6)
```

The outcome of the subgroup analysis appears in Figure 14. This output is fairly self-explanatory. Based on this output, we can derive that we have fitted a mixed-effects model, meaning a random-effects model is used to combine studies within each subgroup and a fixed-effect model is used to combine the subgroups and estimate the summary effect size. The amount of within-group heterogeneity across the two subgroups is assumed to be the same ($\tau^2 = 0.44$ in this case). This combined estimate is derived by pooling the two within-group variance estimates as displayed earlier ($\tau^2 = 0.93$ and $\tau^2 = 0.25$). Once we have the pooled estimate, we then apply it to each study across the two subgroups, meaning every study now shares the same estimate of τ^2 (i.e., 0.44).

```
Mixed-Effects Model (k = 17; tau^2 estimator: DL)

tau^2 (estimated amount of residual heterogeneity):     0.4427 (SE = 0.2518)
tau (square root of estimated tau^2 value):             0.6654
I^2 (residual heterogeneity / unaccounted variability): 97.42%
H^2 (unaccounted variability / sampling variability):   38.70
R^2 (amount of heterogeneity accounted for):            0.00%

Test for Residual Heterogeneity:
QE(df = 15) = 580.5386, p-val < .0001

Test of Moderators (coefficient 2):
QM(df = 1) = 0.9202, p-val = 0.3374

Model Results:

                   estimate      se      zval    pval    ci.lb    ci.ub
intrcpt             -7.9742  0.2892  -27.5726  <.0001  -8.5411  -7.4074  ***
studydesignOthers    0.3452  0.3599    0.9593  0.3374  -0.3601   1.0506

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


      pred    ci.lb    ci.ub    pi.lb    pi.ub
1 0.000344 0.000195 0.000606 0.000083 0.001425

       pred    ci.lb    ci.ub    pi.lb    pi.ub
17 0.000486 0.000319 0.000739 0.000124 0.001910

     pred    ci.lb    ci.ub
 0.000430 0.000307 0.000602
```

Figure 14. A subgroup analysis assuming a common between-study variance component

The test of moderators suggests that the study design does not have a moderating effect ($QM(1) = 0.92$, $p = 0.34$). That is, when we divide the included studies according to their study designs, we fail to find any significant differences between the two subgroups of effect sizes. This conclusion is also supported by the results of the test for residual heterogeneity: there is significant unexplained heterogeneity left in the effect sizes ($QE(15) = 580.54$, $p < 0.01$), which can also explain why R^2 shows 0%. Finally, the estimates for the two subgroup summary proportions and the overall summary proportion are displayed at the bottom of the output. They are 0.00034 (95% CI = 0.0002, 0.00061), 0.00049 (95% CI = 0.00032, 0.00074), and 0.00043 (95% CI = 0.00031, 0.0006), respectively.

There are several other points that are worth noting. Under the framework of the mixed-effect model, the residual heterogeneity estimate here ($QE(15) = 580.54$) is the sum of the two within-group heterogeneity estimates we have obtained above in the random-effects model ($Q(df = 5) = 344.59$, $Q(df = 10) = 235.94$, respectively). When we dummy-code a moderator with two categories, the subset of studies coded as 0 in a dummy variable will function as the reference group, represented by the intercept of the fitted mixed-effects regression model. The other subset of studies coded as 1 will be compared against the reference group. In the running example, the "Birth cohort" group is the reference group,

while the "Others" group is compared against it. The estimate of the intercept (i.e., -7.97) is the logit-transformed summary effect size of the reference group (i.e, logit(0.00034)). The slope is estimated to be 0.35. The sum of the slope and the intercept is equal to -7.629, which is the logit-transformed summary effect size of the "Others" group (i.e., logit(0.00049)).

When calculating the summary effect estimate across the subgroups, the outcomes may vary depending on the specific τ^2 estimate applied. However, even with this variation, the two computational models may reach the same qualitative conclusions. For instance, in the given example, both models agree that the study design doesn't significantly influence the results. In general, Borenstein et al. (2009) recommend pooling the separate τ^2 when the number of studies in a subgroup is small (i.e., less than five studies). In doing so, we can obtain a more accurate estimate of τ^2. In contrast, if we decide not to pool them, each subgroup should ideally consist of at least five studies to ensure moderately stable estimates of τ^2.

7.4 Creating forest plots in the presence of subgroups in R

Many authors conducting meta-analyses of proportions did not construct forest plots correctly for their subgroup analyses. Specifically, numerous published meta-analytic studies did not present the appropriate estimates for either the overall or subgroup summary proportions in their forest plots. These authors failed to consider the two possible assumptions about τ^2 that we have discussed in Section 7.3.

In this section, we will construct forest plots with subgroups under different assumptions (i.e., separate between-study variance components vs. a common between-study variance component). We have obtained the estimates for subgroup and overall summary proportions in the previous section, which can be used to create our forest plots. The following code is used to construct forest plots under the first assumption:

```
# Assumption 1: Do not assume a common between-study
# variance component (use separate within-group
# estimates of between-study variance).

# Option 1: no transformation
ies.summary <- summary(ies, ni = dat$total)
forest(ies.summary$yi, ci.lb = ies.summary$ci.lb,
    ci.ub = ies.summary$ci.ub, rows = c(d:c, b:a))

# Option 2: the logit transformation
ies.summary <- summary(ies.logit, transf =
    transf.ilogit)
forest(ies.summary$yi, ci.lb = ies.summary$ci.lb,
    ci.ub = ies.summary$ci.ub, rows = c(d:c, b:a))
```

```
# Option 3: the double arcsine transformation
ies.summary <- summary(ies, transf = transf.ipft, ni =
   dat$total)
forest(ies.summary$yi,
       ci.lb = ies.summary$ci.lb,
       ci.ub = ies.summary$ci.ub,
       rows = c(d:c, b:a))
```

The code above merely builds the "bones" of a forest plot. More components need to be added to it (e.g., texts, headers, labels, etc.). We also have to manually adjust its appearance to make it look more professional. Dividing a set of included studies into several subgroups in a forest plot using *metafor* has to be done manually with the *rows* argument. Readers may have noticed that the parameters in the argument (a, b, c, and d denotes a particular position on the Y-axis) are ordered from right to left. a specifies the vertical position for plotting the first study in the first subgroup; b specifies the vertical position for plotting the last study in the first subgroup; c specifies the vertical position for plotting the first study in the second subgroup; d specifies the vertical position for plotting the last study in the second subgroup. Mathematically speaking, $b - a + 1$ and $d - c + 1$ should be equal to the number of studies in their corresponding subgroups. c and b do not need to be consecutive numbers. If we order these parameters from left to right, studies will be displayed in reverse order with the first study being displayed at the bottom of the plot and the last study being displayed at the top of all the studies.

To illustrate, we can execute the following code to create a forest plot using the study design as the moderator:

```
# Run the subgroup analysis code with the assumption
# of separate within-group estimates of between-study
# variance components first, then run the following
# code
ies.summary <- summary(ies.logit, transf =
   transf.ilogit)
# par() function specifies font parameters
par(cex = 1, font = 6)
# Set up forest plot
# order= argument ensures that studies are divided by
# the subgroup variable
forest(ies.summary$yi,
       order = ies.summary$studesg,
       ci.lb = ies.summary$ci.lb,
       ci.ub = ies.summary$ci.ub,
       ylim = c(-5, 23),
       xlim = c(-0.005, 0.005),
       slab = paste(dat$author, dat$year, sep = ","),
       ilab = cbind(data = dat$cases, dat$total),
```

```
        ilab.xpos = c(-0.0019, -0.0005),
        ilab.pos = 2,
        rows = c(19:14, 8.5:-1.5),
        at = c(seq(from = 0, to = 0.004, by = 0.001)),
        refline = pes.studydesign$pred,
        main = "",
        xlab = "Proportion",
        digits = 4)
# Add summary polygons for the subgroup and overall
# proportions
par(cex = 1.2, font = 7)
addpoly(pes.birthcohort$pred, ci.lb =
   pes.birthcohort$ci.lb, ci.ub =
   pes.birthcohort$ci.ub, row = 12.8, digits = 5)
addpoly(pes.others$pred, ci.lb = pes.others$ci.lb,
   ci.ub = pes.others$ci.ub, row = -2.7, digits = 5)
addpoly(pes.studydesign$pred, ci.lb =
   pes.studydesign$ci.lb, ci.ub =
   pes.studydesign$ci.ub, row = -4.6, digits = 5)
# Add column headings to the plot
par(cex = 1.1, font = 7)
text(-0.005, 21.8, pos = 4, "Study")
text(c(-0.0026, -0.0014), 21.8, pos = 4, c("Cases",
   "Total"))
text(0.0025, 21.8, pos = 4, "Proportion [95% CI]")
# Add text for the subgroups
text(-0.005, c(9.7, 20.2), pos = 4, c("Others", "Birth
   cohort"))
# Add text for the subgroup and overall proportions
par(cex = 1, font = 7)
text(-0.005, -4.6, pos = 4, c("Overall proportion"))
text(-0.005, 12.8, pos = 4, c("Subgroup proportion"))
text(-0.005, -2.7, pos = 4, c("Subgroup proportion"))
abline(h = -3.7)
```

The generated forest plot is shown in Figure 15. Notice that the overall summary proportion is 0.00045 (95% CI = 0.00033, 0.00061) under the given assumption, which is different than the one derived in the absence of subgroups (0.00042).

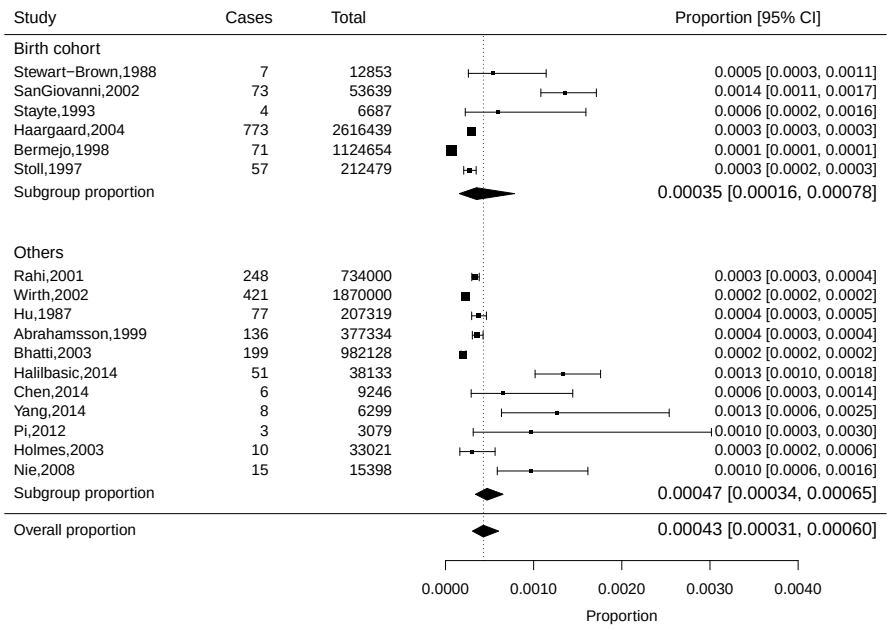

Figure 15. A forest plot with subgroups assuming different τ^2 generated by *metafor*

Under the assumption of a common τ^2, we employ the *rma()* function in *metafor* in conjunction with the *metaprop()* and *forest()* functions in *meta* to produce a forest plot with subgroups. The inclusion of predictors is set by the *mods* argument in *metafor* and the *byvar* argument in *meta*. In the *metaprop()* function, two arguments are particularly noteworthy: *tau.common* determines whether a common τ^2 estimate is applied across subgroups, while *tau.preset* sets the value of τ. Given our assumption, we set *tau.common* to TRUE and *tau.preset* to the pooled τ estimate obtained from the previous section.

```
# Assumption 2: Assume a common between-study variance
# component (pooling within-group estimates of
# between-study variance)
# data= could also be set to ies.logit or ies.da
subganal.moderator <- rma(yi, vi, data = ies, mods = ~
   moderator, method = "DL")
# sm= could also be set to "PLO" or "PFT"
# tau.common= must be TRUE and tau.preset must be
# sqrt(subganal.moderator$tau2)
```

```
pes.summary <- metaprop(cases, total, authoryear, data
   = dat, sm = "PRAW", byvar = moderator,
   tau.common=TRUE, tau.preset =
   sqrt(subganal.moderator$tau2))
# resid.hetstat= must be FALSE
forest(pes.summary, resid.hetstat = FALSE)
```

Assuming that we apply a common τ^2 across subgroups, the following code creates a customized forest plot using the study design as the moderator:

```
subganal.studydesign <- rma(yi, vi, data = ies.logit,
   mods = ~ studydesign, method = "DL")
pes.summary <- metaprop(cases, total, authoryear, data
   = dat, sm = "PLO", method.tau = "DL", method.ci =
   "NAsm", byvar = studydesign, tau.common=TRUE,
   tau.preset = sqrt(subganal.studydesign$tau2))
forest(pes.summary,
       common = FALSE,
       overall = TRUE,
       overall.hetstat = TRUE,
       resid.hetstat = FALSE,
       subgroup.hetstat = TRUE,
       test.subgroup = FALSE,
       fs.hetstat = 10,
       print.tau2 = TRUE,
       print.Q = TRUE,
       print.pval.Q = TRUE,
       print.I2 = TRUE,
       rightcols = FALSE,
       xlim = c(0 ,4),
       leftcols = c("studlab", "effect", "ci"),
       leftlabs = c("Study", "Proportion", "95% C.I."),
       text.random.w = "Subgroup proportion",
       text.random = "Overall proportion",
       xlab = "Prevalence of CC (%)",
       pscale = 1000,
       smlab = " ",
       weight.study = "random",
       squaresize = 0.5,
       col.square = "navy",
       col.diamond = "maroon",
       col.diamond.lines = "maroon",
       digits = 2)
```

The generate forest plot is presented in Figure 16:

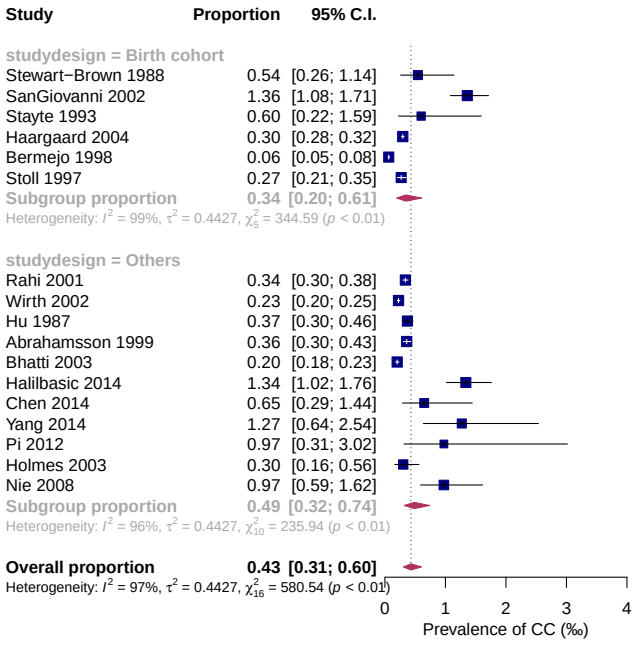

Figure 16. A forest plot with subgroups assuming a common τ^2

Notice that the estimates of τ^2 are identical (0.4427) across two subgroups. The overall summary proportion and its 95% CI (0.43; 95% CI = 0.31, 0.6) are calculated across two subgroups based on the same τ^2 estimate, as well.

7.5 Conducting meta-regression with different types of predictors in R

When we want to evaluate the influence of a continuous moderator, the R code is identical to what we used for subgroup analyses:

```
#data= could also be set to ies.logit or ies.da
metareg.moderator <- rma(yi, vi, data = ies, mods = ~
   moderator)
```

As mentioned above, a mix of continuous and categorical moderators can be regressed on the effect sizes in a meta-regression model. This can be achieved by using the plus sign in the *mods* argument:

```
#data= could also be set to ies.logit or ies.da
metareg.moderators <- rma(yi, vi, data = ies, mods =
~moderatorA + moderatorB + moderatorC + ...)
```

7.6 Visualizing moderator analyses with scatter plots in R

Scatter plots serve as an invaluable visualization tool when assessing potential moderator variables. Such plots, as depicted in Figure 17, are constructed with a regression line, flanked by two curved dotted lines that represent the 95% confidence interval bounds, with studies represented by circles drawn proportional to their study weights (i.e., larger studies appear as larger circles). What's important in scatter plots is the slope of the regression line. Specifically, if the regression line is horizontal or nearly so, it suggests there's no significant association between the moderator and the effect sizes. Conversely, if the regression line has a noticeable slope, it indicates the effect sizes change in relation to the value of the moderator. To determine the significance of this relationship, one can look at the slope and its significance test. A notably positive or negative slope indicates that the predictor plays a significant moderating role, potentially explaining a significant portion of the observed heterogeneity.

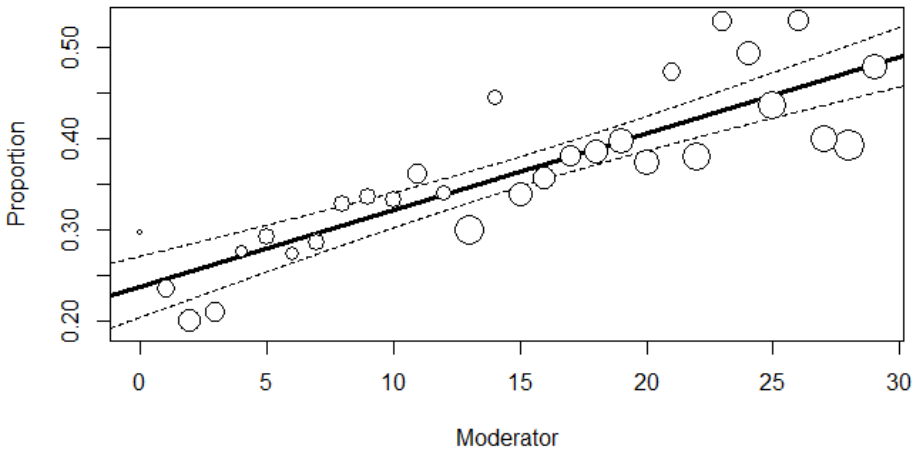

Figure 17. A basic scatter plot

In this section, we will employ the *regplot()* function in the *metafor* package to create scatter plots. *regplot()* offers a distinct advantage over R's native *plot()* function. It simplifies the coding process, making it more user-friendly, especially for those less familiar with R. It helps users to customize their scatter plots with ease.

The following generic code creates weighted scatter plots for subgroup analyses. In a weighted scatter plot, a study is represented by a circle. The weight of a study is depicted by the size of the circle, with a larger circle indicating a greater study weight. In an unweighted scatter plot, the circles are of equal size. Additionally, it is necessary to use dummy variables for categorical moderators (e.g., variables labeled as "studesg" in the running example).

```
# Option 1: no transformation
regplot(subganal.dummyvar, mod = "dummyvar")

# Option 2: the logit transformation
regplot(metareg.dummyvar, mod = "dummyvar",
   transf=transf.ilogit)

# Option 3: the double arcsine transformation
# targ can also be set to list(ni = 1/(pes.da$se)^2)
regplot(subganal.dummyvar, mod = "dummyvar",
   transf=transf.ipft.hm, targ=list(ni=dat$total))
```

Using the running example, we can create a customized scatter plot with a regression line and corresponding 95% CI bounds for "studesg" with the following code:

```
# Conduct a subgroup analysis based on the dummy
# variable "studesg"
subganal.studesg=rma (yi, vi, data = ies.logit, mods =
   ~ studesg, method = "DL")
# Create a scatter plot
regplot(subganal.studesg, mod = "studesg",
        xlab = "Study Design",
        transf=transf.ilogit,
        legend = FALSE,
        label = TRUE,
        shade = "white",
        bg = "transparent",
        lcol = "navy",
        digits = 4)
```

The generated scatter plot is shown in Figure 18.

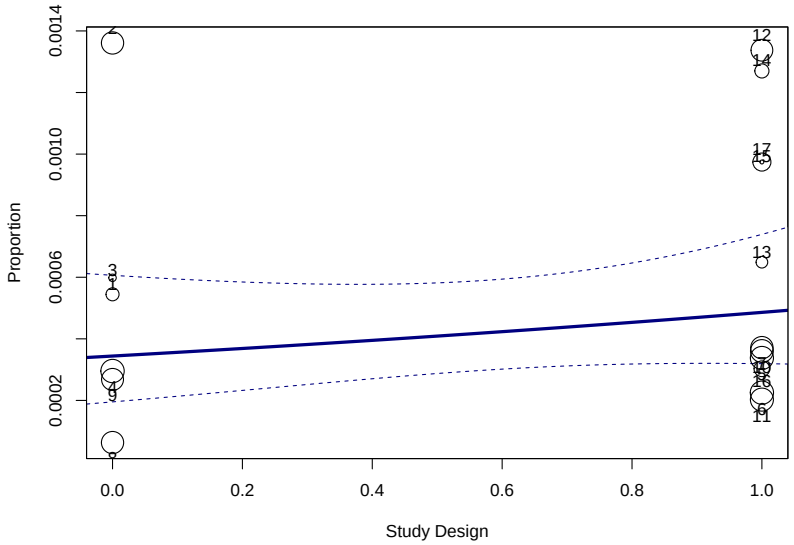

Figure 18. A scatter plot using the study design as the moderator

Upon visual inspection of the scatter plot, it is evident that the slope of the estimated regression line is neither entirely horizontal nor excessively steep, suggesting a weak association between the study design and the observed effects. Furthermore, nearly half of the studies fall outside of the 95% CI bounds, indicating the presence of potentially unidentified moderators.[7]

In the second example, we use the sample size as the moderator (the variable "size" in the provided data set) and evaluate it in a subgroup analysis:

```
subganal.size <- rma(yi, vi, data = ies.logit, mods =
   ~ size, method = "DL")
regplot(subganal.size,
        mod = "size",
        transf=transf.ilogit,
        xlab = "Sample size",
        legend = "topright",
        label = TRUE,
        shade = "white",
        bg = "transparent",
        lcol = "navy",
        digits = 6)
```

[7] If one wants to change the curved slope and 95% CIs lines to straight lines, further steps are needed in R. I've included relevant R code in the supplementary materials.

The generated scatter plot is presented in Figure 19. The code is self-explanatory. Note that the *legend* argument determines if a legend is added to the scatter plot, with its location specified by the user.

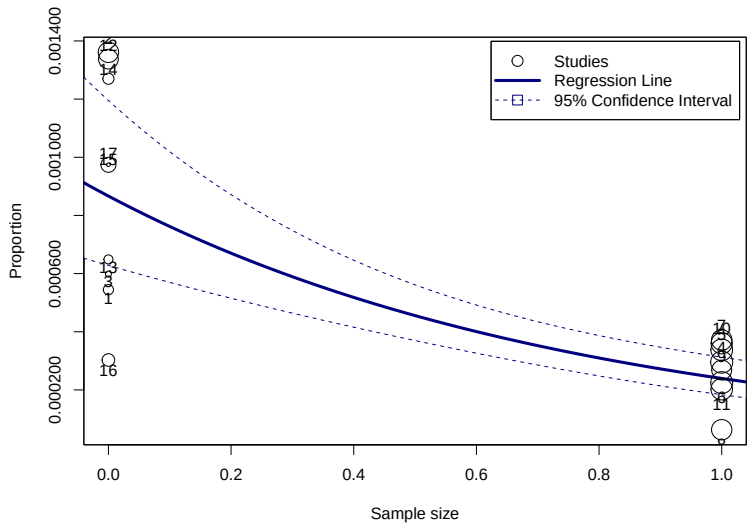

Figure 19. A scatter plot using sample size as the moderator

In this case, the estimated regression line exhibits a noticeably steeper slope. A visual inspection of this scatter plot indicates a negative correlation between the sample size and the observed proportions. When the sample size is less than 100,000, the proportions tend to be higher; when the sample size is larger than 100,000, the proportions tend to be lower. Again, it is important to acknowledge that potential missing moderators may introduce a degree of omitted variable bias here. The outcomes of the subgroup analysis are shown below in Figure 20.

```
Mixed-Effects Model (k = 17; tau^2 estimator: DL)

tau^2 (estimated amount of residual heterogeneity):     0.1398 (SE = 0.0911)
tau (square root of estimated tau^2 value):             0.3739
I^2 (residual heterogeneity / unaccounted variability): 93.90%
H^2 (unaccounted variability / sampling variability):   16.40
R^2 (amount of heterogeneity accounted for):            57.07%

Test for Residual Heterogeneity:
QE(df = 15) = 246.0073, p-val < .0001

Test of Moderators (coefficient 2):
QM(df = 1) = 36.4266, p-val < .0001

Model Results:

         estimate      se      zval    pval    ci.lb    ci.ub
intrcpt   -7.0500  0.1643  -42.9109  <.0001  -7.3720  -6.7280  ***
size      -1.2867  0.2132   -6.0354  <.0001  -1.7046  -0.8689  ***

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

Figure 20. A subgroup analysis for sample size

The results of the test of moderators ($QM(1) = 36.43$, $p < 0.0001$) as well as the significant regression coefficient (-1.29; $Z(15) = -6.04$, $p < 0.0001$) are consistent with our visual interpretation. In stark contrast with the previous subgroup analysis, the R^2 indicates that 57.07% of the true heterogeneity in the observed effect size can be explained by the sample size.

In the running example, Wu et al. (2012) did not examine any continuous predictors. To demonstrate how to generate a weighted scatter plot for a meta-regression with a continuous predictor in R, we will plot the observed effect sizes against the year of publication, represented by the "year" variable in the provided dataset. The code is provided below:

```
metareg.year <- rma(yi, vi, data = ies.logit, mods = ~
   year, method = "DL")
regplot(metareg.year,
        mod = "year",
        transf = transf.ilogit,
        xlab = "Year of publication",
        legend = "topleft",
        label = TRUE,
        shade = "white",
        bg = "white",
        lcol = "navy",
        digits = 6)
```

The generated scatter plot is presented in Figure 21.

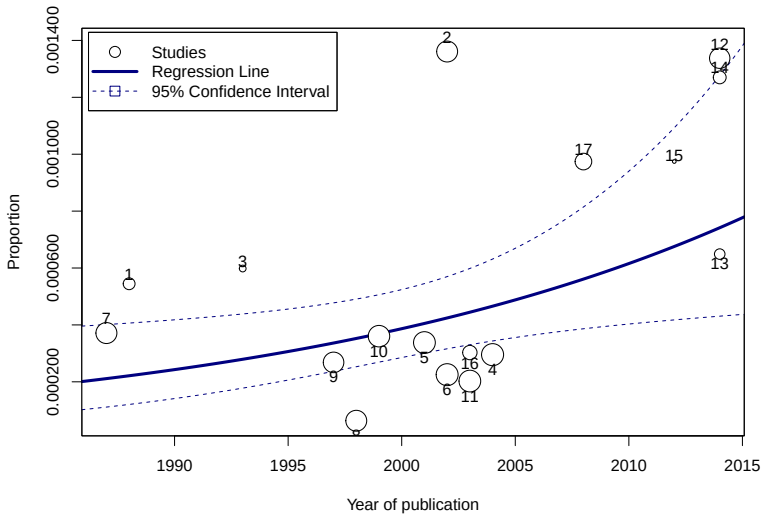

Figure 21. A scatter plot using the publication year as the moderator

8 Common procedures addressing publication bias do not apply to meta-analyses of proportions

One of the major threats to the validity of meta-analysis is publication bias. This is a phenomenon where journals tend to accept and publish a study depending on the direction or strength of its results (MarksAnglin & Chen, 2020). Compared with studies with statistically significant results, small studies reporting insignificant results or small effects are less likely to be published and subsequently included in a meta analysis (Dickersin, 1990; Littell et al., 2008). Omitting unpublished studies in a systematic review could lead to a biased meta-analytic estimate of the summary effect (Song, Eastwood, Gilbody, Duley, & Sutton, 2000). As smaller studies require larger effect sizes to achieve statistical significance (Sterne, Gavaghan, & Egger, 2000), only those small studies with large effects get published and included in a relevant meta-analysis. Thus, a meta-analysis that only includes studies with large effects and fails to include studies with small effects at the same time could overestimate the true effect (Cuijpers, 2016).

Current methods of detecting publication bias and assessing its impact are developed for meta-analyses of randomized control trials. These methods rely on certain assumptions (Borenstein et al., 2009). Firstly, regardless of the significance of their effects, large studies are most likely to be published. Secondly, only small studies demonstrating significant and substantial effects tend to be published. Lastly, most moderate-scale studies that yield significant results also

tend to be published. Consequently, as the sample size of a study decreases, the likelihood of it being affected by publication bias increases. Traditional methods such as trim-and-fill, the rank correlation test, Egger's regression model, as well as the more sophisticated weighted selection approaches (e.g., Vevea & Hedges, 1995; Vevea & Woods, 2005) have all operated under the assumption that the publication likelihood depends on sample size, statistical significance, or the direction of results (Coburn & Vevea, 2015).

While empirical research has confirmed the dominant role of statistical significance in study publication (Preston, Ashby, & Smyth, 2004), the actual publication selection process across different fields is much more intricate. Cooper, DeNeve, and Charlton (1997) have demonstrated that decisions regarding study publication are influenced by various criteria or "filters" set by journal editors and reviewers, independent of methodological quality and significance. These filters can include factors such as research funding sources, societal preferences related to race and gender during the study's conduction, and even findings that challenge pre-existing beliefs. Consequently, the traditional methods may fail to capture the full complexity of the publication selection process.

In practice, authors of meta-analyses of proportions have employed these methods in their attempts to detect publication bias. However, studies included in meta-analyses of proportions are observational and non-comparative. In other words, they only report a proportion or prevalence of an event, which inherently precludes the testing of statistical significance for their findings (Borenstein, 2019). Consequently, the interpretation of the outcomes from such studies is not contingent on the null hypothesis significance test and thus cannot be categorized as either "positive/negative" or "desirable/undesirable." The significance levels are, therefore, unlikely to influence publication decisions regarding these studies (Maulik, Mascarenhas, Mathers, Dua, & Saxena, 2011). Authors who report low proportions (e.g., rare event rates) are equally likely to have their work published as those reporting very high proportions (e.g., high cure rates), given that the study quality meets rigorous publication standards. Consequently, the traditional publication bias assessment procedures may struggle to identify publication bias in meta-analyses of proportions, as bias in non-comparative studies can be introduced for reasons unrelated to statistical significance.

Borenstein (2019) warns meta-analysts that it is a mistake to apply publication bias procedures to studies of prevalence. Our suggestion aligns with his. When conducting meta-analyses of proportions, we believe that the traditional publication bias tests and modeling tools developed for randomized controlled trials have limited utility and, therefore, should not be used. Any conclusions drawn regarding the presence of publication bias based on these methods should be approached with caution.

References

Agresti, A., & Coull, B. A. (1998). Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*,

52(2), 119–126. doi: https://doi.org/10.2307/2685469

Anzures-Cabrera, J., & Higgins, J. P. (2010). Graphical displays for meta-analysis: An overview with suggestions for practice. *Research Synthesis Methods*, *1*(1), 66–80. doi: https://doi.org/10.1002/jrsm.6

Barendregt, J. J., Doi, S. A., Lee, Y. Y., Norman, R. E., & Vos, T. (2013). Meta-analysis of prevalence. *Journal of Epidemiology and Community Health*, *67*(11), 974–978. doi: https://doi.org/10.1136/jech-2013-203104

Borenstein, M. (2019). *Common mistakes in meta-analysis and how to avoid them*. Biostat.

Borenstein, M., Hedges, L., Higgins, J., & Rothstein, H. (2005). *Comprehensive meta-analysis version 2*. Biostat.

Borenstein, M., Hedges, L. V., Higgins, J., & Rothstein, H. R. (2009). *Introduction to meta-analysis*. John Wiley & Sons. doi: https://doi.org/10.1002/9781119558378

Borenstein, M., Hedges, L. V., Higgins, J. P., & Rothstein, H. R. (2010). A basic introduction to fixed-effect and random-effects models for meta-analysis. *Research synthesis methods*, *1*(2), 97–111. doi: https://doi.org/10.1002/jrsm.12

Borenstein, M., Higgins, J. P., Hedges, L. V., & Rothstein, H. R. (2017). Basics of meta-analysis: I^2 is not an absolute measure of heterogeneity. *Research synthesis methods*, *8*(1), 5–18.

Box, G. E., Hunter, J. S., & Hunter, W. G. (2005). *Statistics for experimenters: design, innovation, and discovery* (Vol. 2). Wiley-Interscience.

Card, N. A. (2015). *Applied meta-analysis for social science research*. Guilford Publications.

Chung, Y., Rabe-Hesketh, S., & Choi, I. H. (2013). Avoiding zero between-study variance estimates in random-effects meta-analysis. *Statistics in Medicine*, *32*(23), 4071–4089. doi: https://doi.org/10.1002/sim.5821

Coburn, K. M., & Vevea, J. L. (2015). Publication bias as a function of study characteristics. *Psychological Methods*, *20*(3), 310–330. doi: https://doi.org/10.1037/met0000046

Cochran, W. G. (1954). The combination of estimates from different experiments. *Biometrics*, *10*(1), 101–129. doi: https://doi.org/10.2307/3001666

Cooper, H., DeNeve, K., & Charlton, K. (1997). Finding the missing science: The fate of studies submitted for review by a human subjects committee. *Psychological Methods*, *2*(4), 447–452. doi: https://doi.org/10.1037/1082-989x.2.4.447

Cornell, J. E., Mulrow, C. D., Localio, R., Stack, C. B., Meibohm, A. R., Guallar, E., & Goodman, S. N. (2014). Random-effects meta-analysis of inconsistent effects: a time for change. *Annals of Internal Medicine*, *160*(4), 267–270. doi: https://doi.org/10.7326/m13-2886

Cuijpers, P. (2016). *Meta-analyses in mental health research: A practical guide*. Pim Cuijpers Uitgeverij.

Davey, J., Turner, R. M., Clarke, M. J., & Higgins, J. P. (2011). Characteristics of meta-analyses and their component studies in the cochrane database of

systematic reviews: a cross-sectional, descriptive analysis. *BMC Medical Research Methodology*, *11*(1), 1–11. doi: https://doi.org/10.1186/1471-2288-11-160

Del Re, A. C. (2015). A practical tutorial on conducting meta-analysis in R. *The Quantitative Methods for Psychology*, *11*(1), 37–50. doi: https://doi.org/10.20982/tqmp.11.1.p037

DerSimonian, R., & Laird, N. (1986). Meta-analysis in clinical trials. *Controlled Clinical Trials*, *7*(3), 177–188. doi: https://doi.org/10.1016/0197-2456(86)90046-2

Dickersin, K. (1990). The existence of publication bias and risk factors for its occurrence. *JAMA*, *263*(10), 1385–1389. doi: https://doi.org/10.1001/jama.1990.03440100097014

Egger, M., Schneider, M., & Smith, G. D. (1998). Spurious precision? meta-analysis of observational studies. *British Medical Journal*, *316*(7125), 140–144.

Evangelou, E., & Veroniki, A. A. (2022). *Meta-research: Methods and protocols*. Springer.

Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y., & Tu, X. M. (2014). Log-transformation and its implications for data analysis. *Shanghai Archives of Psychiatry*, *26*(2), 105–109.

Freeman, M. F., & Tukey, J. W. (1950). Transformations related to the angular and the square root. *Annals of Mathematical Statistics*, *21*(4), 607–611. doi: https://doi.org/10.1214/aoms/1177729756

Fusar-Poli, P., Schultze-Lutter, F., Cappucciati, M., Rutigliano, G., Bonoldi, I., Stahl, D., & Woods, S. W. (2015). The dark side of the moon: meta-analytical impact of recruitment strategies on risk enrichment in the clinical high risk state for psychosis. *Schizophrenia Bulletin*, *42*(3), 732–743. doi: https://doi.org/10.1093/schbul/sbv162

Gillen, S., Schuster, T., Meyer Zum Bschenfelde, C., Friess, H., & Kleeff, J. (2010). Preoperative/neoadjuvant therapy in pancreatic cancer: a systematic review and meta-analysis of response and resection percentages. *Plos Medicine*, *7*(4), e1000267–e1000267. doi: https://doi.org/10.1371/journal.pmed.1000267

Hamza, T. H., van Houwelingen, H. C., & Stijnen, T. (2008). The binomial distribution of meta-analysis was preferred to model within-study variability. *Journal of Clinical Epidemiology*, *61*(1), 41–51. doi: https://doi.org/10.1016/j.jclinepi.2007.03.016

Hardy, R. J., & Thompson, S. G. (1998). Detecting and describing heterogeneity in meta-analysis. *Statistics in Medicine*, *17*(8), 841–856. doi: https://doi.org/10.1002/(sici)1097-0258(19980430)17:8¡841::aid-sim781¿3.0.co;2-d

Harrer, M., Cuijpers, P., A, F. T., & Ebert, D. D. (2021). *Doing meta-analysis with R: A hands-on guide* (1st ed.). Boca Raton, FL and London: Chapman & Hall/CRC Press. doi: https://doi.org/10.1201/9781003107347

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. Aca-

demic Press. doi: https://doi.org/10.2307/2289186

Hedges, L. V., & Vevea, J. L. (1998). Fixed-and random-effects models in meta-analysis. *Psychological methods*, *3*(4), 486. doi: https://doi.org/10.1037/1082-989x.3.4.486

Higgins, J. P., & Green, S. (2006). Cochrane handbook for systematic reviews of interventions 4.2. 6 [updated september 2006]. *The cochrane library*, *4*, 2006.

Higgins, J. P., & Thompson, S. G. (2002). Quantifying heterogeneity in a meta-analysis. *Statistics in Medicine*, *21*(11), 1539–1558. doi: https://doi.org/10.1002/sim.1186

Higgins, J. P., Thompson, S. G., Deeks, J. J., & Altman, D. G. (2003). Measuring inconsistency in meta-analyses. *British Medical Journal*, *327*(7414), 557–560. doi: https://doi.org/10.1136/bmj.327.7414.557

Huedo-Medina, T. B., Snchez-Meca, J., Marn-Martnez, F., & Botella, J. (2006). Assessing heterogeneity in meta-analysis: Q statistic or I^2 index? *Psychological methods*, *11*(2), 193–206. doi: https://doi.org/10.1037/1082-989x.11.2.193

Hunter, J., Saratzis, A., Sutton, A. J., Boucher, R. H., Sayers, R. D., & Bown, M. J. (2014). In meta-analyses of proportion studies, funnel plots were found to be an inaccurate method of assessing publication bias. *Journal of clinical epidemiology*, *67*(8), 897–903. doi: https://doi.org/10.1016/j.jclinepi.2014.03.003

Hunter, J., & Schmidt, F. (2000). Fixed effects vs. random effects meta-analysis models: Implications for cumulative research knowledge. *International Journal of selection and assessment*, *8*(4), 275–292. doi: https://doi.org/10.1111/1468-2389.00156

Ioannidis, J. P., Patsopoulos, N. A., & Evangelou, E. (2007). Uncertainty in heterogeneity estimates in meta-analyses. *British Medical Journal*, *335*(7626), 914–916. doi: https://doi.org/10.1136/bmj.39343.408449.80

Keithlin, J., Sargeant, J., Thomas, M. K., & Fazil, A. (2014). Systematic review and meta-analysis of the proportion of campylobacter cases that develop chronic sequelae. *BMC Public Health*, *14*(1), 1–19. doi: https://doi.org/10.1186/1471-2458-14-1203

Knapp, G., Biggerstaff, B. J., & Hartung, J. (2006). Assessing the amount of heterogeneity in random-effects meta-analysis. *Biometrical journal*, *48*(2), 271–285. doi: https://doi.org/10.1002/bimj.200510175

Lewis, S., & Clarke, M. (2001). Forest plots: trying to see the wood and the trees. *British Medical Journal*, *322*(7300), 1479–1480. doi: https://doi.org/10.1136/bmj.322.7300.1479

Lijmer, J. G., Bossuyt, P. M., & Heisterkamp, S. H. (2002). Exploring sources of heterogeneity in systematic reviews of diagnostic tests. *Statistics in medicine*, *21*(11), 1525–1537. doi: https://doi.org/10.1002/sim.1185

Lin, L., & Xu, C. (2020). Arcsine-based transformations for meta-analysis of proportions: Pros, cons, and alternatives. *Health Science Reports*, *3*(3), e178. doi: https://doi.org/10.1002/hsr2.178

Lipsey, M. W., & Wilson, D. B. (2001). *Practical meta-analysis*. Sage Publications.

Littell, J. H., Corcoran, J., & Pillai, V. (2008). *Systematic reviews and meta-analysis*. Oxford University Press. doi: https://doi.org/10.1093/acprof:oso/9780195326543.001.0001

Ma, Y., Chu, H., & Mazumdar, M. (2016). Meta-analysis of proportions of rare events a comparison of exact likelihood methods with robust variance estimation. *Communications in statistics: Simulation and computation*, *45*(8), 3036–3052.

MarksAnglin, A., & Chen, Y. (2020). A historical review of publication bias. *Research synthesis methods*, *11*(6), 725–742. doi: https://doi.org/10.31222/osf.io/zmdpk

Maulik, P. K., Mascarenhas, M. N., Mathers, C. D., Dua, T., & Saxena, S. (2011). Prevalence of intellectual disability: a meta-analysis of population-based studies. *Research in Developmental Disabilities*, *32*(2), 419–436. doi: https://doi.org/10.1016/j.ridd.2010.12.018

Miller, J. J. (1978). The inverse of the freeman-tukey double arcsine transformation. *American Statistician*, *32*(4), 138. doi: https://doi.org/10.2307/2682942

Nyaga, V. N., Arbyn, M., & Aerts, M. (2014). Metaprop: a stata command to perform meta-analysis of binomial data. *Archives of Public Health*, *72*(1), 39. doi: https://doi.org/10.1186/2049-3258-72-39

Petrie, A., Bulman, J. S., & Osborn, J. F. (2003). Further statistics in dentistry part 8: Systematic reviews and meta-analyses. *British Dental Journal*, *194*(2), 73–78. doi: https://doi.org/10.1038/sj.bdj.4809877

Preston, C., Ashby, D., & Smyth, R. (2004). Adjusting for publication bias: modelling the selection process. *Journal of Evaluation in Clinical Practice*, *10*(2), 313–322. doi: https://doi.org/10.1111/j.1365-2753.2003.00457.x

R Core Team. (2022). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `http://www.R-project.org`

Raudenbush, S. W., & Bryk, A. S. (1985). Empirical bayes meta-analysis. *Journal of Educational Statistics*, *10*(2), 75–98. doi: https://doi.org/10.2307/1164836

Ried, K. (2006). Interpreting and understanding meta-analysis graphs: a practical guide. *Australian Family Physician*, *35*(8), 635–638.

RStudio Team. (2022). Rstudio: Integrated development for R [Computer software manual]. Boston, MA. Retrieved from `http://www.rstudio.com/`

Rücker, G., Schwarzer, G., Carpenter, J. R., & Schumacher, M. (2008). Undue reliance on I^2 in assessing heterogeneity may mislead. *BMC medical research methodology*, *8*, 1–9. doi: https://doi.org/10.1186/1471-2288-8-79

Sahai, H., & Ageel, M. I. (2012). *The analysis of variance: Fixed, random and mixed models*. Springer Science Business Media.

Schmidt, F. L., & Hunter, J. E. (2014). *Methods of meta-analysis: Correcting error and bias in research findings*. Sage Publications. doi:

https://doi.org/10.4135/9781483398105

Schoonjans, F. (2017). Medcalc manual: Easy-to-use statistical software [Computer software manual]. Retrieved from https://www.medcalc.org/download/medcalcmanual.pdf

Schwarzer, G., Carpenter, J. R., & Rücker, G. (2015). *Meta-analysis with R*. Springer. doi: https://doi.org/10.1007/978-3-319-21416-0

Schwarzer, G., Chemaitelly, H., Abu-Raddad, L. J., & Rücker, G. (2019). Seriously misleading results using inverse of freeman-tukey double arcsine transformation in meta-analysis of single proportions. *Research synthesis methods*, *10*(3), 476–483. doi: https://doi.org/10.1002/jrsm.1348

Song, F., Eastwood, A., Gilbody, S., Duley, L., & Sutton, A. (2000). Publication and related biases: a review. *Health Technology Assessment*, *4*(10), 1–115.

Sterne, J. A., Gavaghan, D., & Egger, M. (2000). Publication and related bias in meta-analysis: power of statistical tests and prevalence in the literature. *Journal of Clinical Epidemiology*, *53*(11), 1119–1129.

Tabachnick, B. G., Fidell, L. S., & Osterlind, S. J. (2013). *Using multivariate statistics*. Pearson.

Thompson, S. G. (1994). Why sources of heterogeneity in meta-analysis should be investigated. *British Medical Journal*, *309*(6965), 1351–1355.

Thompson, S. G., & Higgins, J. P. (2002). How should meta-regression analyses be undertaken and interpreted? *Statistics in Medicine*, *21*(11), 1539–1558. doi: https://doi.org/10.1002/sim.1187

Thompson, S. G., & Sharp, S. J. (1999). Explaining heterogeneity in meta-analysis: a comparison of methods. *Statistics in Medicine*, *18*(20), 2693–2708. doi: https://doi.org/10.1002/(sici)1097-0258(19991030)18:20¡2693::aid-sim235¿3.0.co;2-v

Thorlund, K., Wetterslev, J., Awad, T., Thabane, L., & Gluud, C. (2011). Comparison of statistical inferences from the dersimonian-laird and alternative random-effects model meta-analyses-an empirical assessment of 920 cochrane primary outcome meta-analyses. *Research Synthesis Methods*, *2*(4), 238–253.

Veroniki, A. A., Jackson, D., Viechtbauer, W., Bender, R., Bowden, J., Knapp, G., & Salanti, G. (2016). Methods to estimate the between-study variance and its uncertainty in meta-analysis. *Research Synthesis Methods*, *7*(1), 55–79.

Vevea, J. L., & Hedges, L. V. (1995). A general linear model for estimating effect size in the presence of publication bias. *Psychometrika*, *60*(3), 419–435. doi: https://doi.org/10.1007/bf02294384

Vevea, J. L., & Woods, C. M. (2005). Publication bias in research synthesis: sensitivity analysis using a priori weight functions. *Psychological Methods*, *10*(4), 428–443. doi: https://doi.org/10.1037/1082-989x.10.4.428

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, *36*(3), 1–48.

Viechtbauer, W., & Cheung, M. W. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, *1*(2), 112–125. doi:

https://doi.org/10.1002/jrsm.11

Wang, K. S., & Liu, X. (2016). Statistical methods in the meta-analysis of prevalence of human diseases. *Journal of Biostatistics and Epidemiology*, *2*(1), 20–24.

Wu, X., Long, E., Lin, H., & Liu, Y. (2016). Prevalence and epidemiological characteristics of congenital cataract: a systematic review and meta-analysis. *Scientific Reports*, *6*(1), 1–10. doi: https://doi.org/10.1038/srep28564

Xu, C., et al. (2021). The Freeman–Tukey double arcsine transformation for the meta-analysis of proportions: Recent criticisms were seriously misleading. *Journal of evidence-based medicine*, *14*(4), 259–261. doi: https://doi.org/10.1111/jebm.12445