Appendix A: Reporting results of multilevel models

Learning objectives

After reading and studying this section, students should be able to:

- Report on the different elements of a multilevel analysis.
- Output regression tables for different types of models.
- Customize regression tables to include/exclude certain components.
- Export tables to Microsoft Word (or some other format).

.1 What to consider when reporting results

When writing up multilevel model (MLM) results, it is important to be transparent with regard to how the data were analyzed and what approaches were used in order for readers (and reviewers) to have faith in the study findings. Studies have examined the reporting practices associated with multilevel models over the years and have found that, although reporting has gotten better over time, practices can still be greatly improved (Dedrick et al., 2009). For example, in a review (Luo et al., 2021) of approximately 301 studies that used MLM from 19 journals published from 2009 to 2018, around a third did not report what software was used to estimate the models, less than 14% reported checking for model assumptions, and 90% did not screen for outliers.

As with many statistical procedures, modeling choices can impact the tenability and interpretation of results. Throughout the model building process, several decisions must be made. Researchers should know the implications of making (or not making) certain choices. Statistics has been called the "science of defaults" (Gelman, 2014, p. 291)– established over time through convenience or convention– though some decisions are too important to leave to software defaults. For academics, the choices made (if deemed acceptable/unacceptable by reviewers) can spell the difference between the rejection or acceptance of a manuscript (or proposal).

There are plenty of modeling choices to be made. For example:

- When reporting pseudo-R^2 measures, there are different kinds approaches that can be used (LaHuis, Hartman, Hakoyama, & Clark, 2014). Depending as well on the field, some types of pseudo-R^2 measures may be more well understood than others.
- For generalized linear mixed models (GLMMs; see Chapter 8), there are several types of intraclass correlation coefficients (ICCs) that can be computed (Chakraborty & Hossain, 2018) and estimation methods too may also differ (McCoach et al., 2018).
- For studies with a few clusters, the choice of using restricted maximum likelihood (REML) vs. maximum likelihood (ML) can make a difference (Huang, 2016) (see Chapter 4, p. 46).
- When testing for the statistical significance of the random effects, was a likelihood ratio test [LRT] or a *modified* LRT (LaHuis & Ferguson, 2009) used (see Chapter 4, pp. 53-54)?

Generally, you want readers to not have to ask, "I wonder what they did to get those results?" I present below a list of questions that should be considered when reporting MLM results.[1]

Sample characteristics:

- What was the nested data structure (e.g., how many levels; what were the units at each level?)
- How many units were in each level, on average?
- What was the range of the number of lower-level units in each group/cluster?

Analytic strategy:

- What equation can best represent your model?
- What estimation method was used (e.g., ML, REML)?
- If there were convergence issues, how was this addressed?
- What software (and version) was used (when using R, what packages as well)?
- If degrees of freedom were used, what kind?
- If robust standard errors were used, what kind?
- What type of models were estimated (i.e., unconditional, random intercept, random slope)?
- If a GLMM is fit, what distribution family (e.g., binomial, Poisson) and link function (e.g., logit, log) were used?
- What variables were centered and what kind of centering was used?

[1] For MLMs fit using a Bayesian framework, which I do not discuss, additional information should be reported (Depaoli & van de Schoot, 2017).

- What model assumptions were checked and what were the results?
- Were outliers present and how were they treated?
- Were multicollinearity and homoskedasticity examined (and addressed if ever)?
- What percent of data were missing (if any) and how were missing data addressed?
- Was a power analysis performed?
- If the data or syntax are publicly available, how can these be accessed?

Results:

- What was the ICC of the outcome variable (if a binary outcome, what kind of ICC method was used)?
- Are fixed effects and variance components reported?
- What inferential statistics were used (e.g., t-statistics, LRTs)?
- How precise were the results (report the standard errors and/or confidence intervals)?
- Were model comparisons performed (e.g., AIC, BIC, if using an LRT, report the χ^2, degrees of freedom, and p value)?
- Were effect sizes reported (e.g., Cohen's d, pseudo-R^2)?
- What kind of pseudo-R^2 were reported?

This may seem like a lot but these do not need to have long-winded explanations and many of these can be reported in a few sentences. When publishing, some physical/print journals will have page limits as space is at a premium.[2] For example, we may write (if appropriate):

- "Models were fit using restricted maximum likelihood using the *lme4* (Bates, Mächler, Bolker, & Walker, 2015) package in R 4.0 (R Core Team, 2020)."
- "Inferential tests for fixed effects were conducted using t-tests with Satterthwaite degrees of freedom (dof) approximations computed using the *lmerTest* package (Kuznetsova, Brockhoff, & Christensen, 2017). Satterthwaite dof have been shown to be effective at controlling for Type I error rates when used with multilevel models (Luke, 2017)."

Some of these may also be put (e.g., power analyses details) in an online appendix. Of course, include the relevant citations too (also good to support your decision choices if necessary). For another checklist (with a focus on the medical literature) of what to report when conducting multilevel analyses, interested readers can also consult the open access (freely downloadable) article of Monsalves et al. (2020).

[2] As an example of this, *School Psychology Review* has a 35 (double-spaced) page limit, inclusive of all tables, figures, and references. For *Prevention Science*, the (inclusive) limit is 30 pages.

Also, it is up to the researcher to determine what is most relevant to show for their intended audience. For example, at times, I do not fixate on pseudo-R^2 measures. When presenting results from cluster randomized controlled trials, reviewers may easily misinterpret what these measures mean (more so with nonnormal outcomes). An illustrative case I often discuss is that of one of the most known experiments in education on classroom size, the Tennessee Project STAR (discussed in detail in Chapter 9). For that experiment, the intervention marginal R^2, for both reading and math scores, was less than 1% (for kindergarteners). Reviewers may dismiss that as being of negligible size and the Cohen's d was approximately 0.20 (which may also be deemed as small). However, if a student is subject to the intervention over multiple years (e.g., is in a small class), this effect may accumulate over time making it more sizeable.

.2 Reporting regression results

Most of the modeling choices made will be boiled down into a table of results. Manually making this table (e.g., copying and pasting results from R) is possible. However, that can be time consuming and also error prone. Using various packages, a compact table of results can be exported into a format (e.g., Microsoft Word) that is easier to work with and edit outside of R.

Over the years, various packages have been developed that can help automate the formatting of regression model results. Some popular packages include: `stargazer` (Hlavac, 2022), `jtools` (Long, 2020) (through the use of the `export_summs` function), `texreg` (Leifeld & Zucca, 2022), and `gtsummary` (Sjoberg, Whiting, Curry, Lavery, & Larmarange, 2021). The different packages all offer some level of customization that is important when dealing with different types of models. The different packages also have some peculiarities which users should be aware of as well (so make sure you go through the help files that are provided with the packages).

Although I have used several packages for creating regression model output, a package that I more frequently use nowadays is the `modelsummary` (Arel-Bundock et al., 2022) package. The package is constantly being enhanced, creates output in a format that I like (but is also customizable), works with a variety of other regression modeling functions, and is relatively straightforward to use. A nice help site (at https://vincentarelbundock.github.io/modelsummary/reference/modelsummary.html) is also available which can walk users through a variety of customizations. For the function to work, the `broom.mixed` package will have to be installed (so make sure to install that). Although the packages mentioned earlier all work well, they can (on occasion) need some workarounds to display results in the format that I often use.[3]

[3] For example, `stargazer` for a long time did not recognize model results created by `lmerTest` and had to be "fooled" that this was an `lme4` object (i.e., `lmerMod`). When using logistic regression models, `texreg` would only display a single `*` to show statistical significance

.2.1 Basic usage of the `modelsummary` package

The main function for creating the regression table results is the `modelsummary` function (which can be also called using the shorter `msummary` function) in the `modelsummary` package. Basic usage is straightforward and instead of using the `summary` function on the model object, we can use the `modelsummary` function.

```
library(modelsummary)
library(lmerTest)
library(MLMusingR)
data(engage) #in the MLMusingR package
fullm <- lmer(eng ~ mot + gpa + grade + frpm + rural + (1|school),
  data = engage) #sample model
modelsummary(fullm, stars = TRUE,
 title = 'Multilevel Regression Model Results for
 Student Engagement.')
```

All that is needed is to use the model object (`fullm`) with the function. Two options are also included. The `stars = TRUE` is included to add the asterisks to signify the different p values (as conventionally done). The `title =` is added to provide a title to the table.[4] In the fixed effects portion, the regression coefficients and the model-based standard errors are shown (in parenthesis).

At the bottom of the table (see Table 1), several other additional measures are included by default– which can be customized or omitted as necessary. Note that the *SD*s shown represent the standard deviation of the random effects, not the variance. The random effect *SD*s should be squared to get the variance estimates of τ_{00} and σ^2.

In addition, several other goodness of fit measures are shown. Also note that the R^2 measures shown are based on the formulas of Nakagawa and Schielzeth (2013) which were discussed in detail in Chapter 8. `RMSE` refers to the root mean square error which can be computed manually using: `sqrt(mean(resid(fullm)^2))`.

To completely omit the goodness of fit measures, the option `gof_map = NA` can be added. Alternatively, multiple measures can be omitted too (separated by `|`) using (just as an example) `gof_omit = 'RMSE|BIC|Obs'`. The whole name of the statistic does not have to be spelled out but just match a part of it (e.g., `Obs` can be used instead of specifying `Num.Obs`). If `IC` were specified, this would omit `BIC`, `AIC`, and `ICC` as they all contain the letters `IC`.

(regardless of whether $p < .05$ or $p < .001$) when results were shown using odds ratios and confidence intervals. The `gtsummary` package would, by default, not show standard errors and needed some other option to "unhide" the column with the standard errors.

[4]If using the American Psychological Association (APA) format, the table number and the table title should be on separate lines.

Table 1: Multilevel Regression Model Results for Student Engagement.

	Model 1
(Intercept)	4.816**
	(1.463)
mot	0.545***
	(0.126)
gpa	0.536***
	(0.154)
gradegr7	−1.212***
	(0.287)
gradegr8	−1.562***
	(0.367)
frpm	−0.035***
	(0.009)
rural	0.519+
	(0.309)
SD (Intercept school)	0.518
SD (Observations)	2.930
Num.Obs.	528
R2 Marg.	0.133
R2 Cond.	0.159
AIC	2673.5
BIC	2711.9
ICC	0.03
RMSE	2.88

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

To display the 95% confidence intervals instead of the standard errors, you can indicate: `modelsummary(fullm, statistic = "({conf.low}, {conf.high})")`. The keywords of `{conf.low}` and `{conf.high}` refer to the lower and upper bounds of the 95% confidence intervals. The confidence interval level can also be set manually using the `conf_level =` option (which is set by default to `conf_level = .95`).

If you want the results to be horizontal instead of stacked vertically, you can use: `modelsummary(fullm, estimate = "{estimate} {stars} ({conf.low}, {conf.high})", statistic = NULL)`. The `{estimate}` keyword refers to the regression coefficient and the `{stars}` keyword refers to the `*` that are used to show different p values. If you want standard errors displayed on one line with the regression coefficient: `modelsummary(fullm, estimate = "{estimate} {stars} ({std.error})", statistic = NULL)`. The `statistic` option basically shows what is reported below the estimate (the regression coefficient). When it is set to `NULL`, it is not shown. Also, by default, output is shown to 3 decimal places for the regression coefficients and the standard errors. To change this to only two decimal places, specify the option `fmt = 2`.

.2.2 Comparing multiple models

The story of one's analysis is often told through multiple models instead of just one model. Comparing models– building from simpler to more complex models– allows researchers to piece apart the contributions of the additional variables, over and above the variables in a previous model. To show multiple models, we can use a `list` function to string together several models.

This is shown below (note how the multiple models are specified in a list). Note, we can also (if desired), omit the random effects that begin with `SD` by including `coef_omit = "SD"` (make sure no other variables have the letters `SD` in them or they too will be omitted). Model names can also be specified instead of showing the generic "Model 1", "Model 2", and "Model 3" (see syntax below and output in Table 2):

```
nullm <- lmer(eng ~ (1|school), data = engage) #the unconditional model
lev1only <- lmer(eng ~ mot + gpa + grade + (1|school),
 data = engage)
#only level-1 predictors
modelsummary(list("Unconditional" = nullm,
 "Level-1 variables only" = lev1only,
 "Full model" = fullm), stars = TRUE, gof_omit = 'RMSE|IC|Obs',
 coef_omit = "SD",  title = 'Multilevel Regression Model Results
 for Student Engagement.')
```

Table 2: Multilevel Regression Model Results for Student Engagement.

	Unconditional	Level-1 variables only	Full model
(Intercept)	8.777***	2.735*	4.816**
	(0.196)	(1.341)	(1.463)
mot		0.545***	0.545***
		(0.126)	(0.126)
gpa		0.505**	0.536***
		(0.155)	(0.154)
gradegr7		−1.194***	−1.212***
		(0.290)	(0.287)
gradegr8		−1.525***	−1.562***
		(0.371)	(0.367)
frpm			−0.035***
			(0.009)
rural			0.519+
			(0.309)
R2 Marg.	0.000	0.089	0.133
R2 Cond.	0.078	0.157	0.159

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

.2.3 Exporting tables

The different variable names can also be changed using syntax. However, it may be easier to do all the renaming and customization by exporting the table into another format. For example, the R^2 measures and ICC should not have a leading zero (e.g., `.133` vs `0.133`) and it might just be easier to edit this in Microsoft Word or Excel.[5]

To export the table to Microsoft Word, we can specify:

```
modelsummary(list("Unconditional" = nullm,
 "Level-1 variables only" = lev1only,
 "Full model" = fullm), stars = TRUE, gof_omit = 'RMSE|BIC|Obs',
 notes = "Standard errors within parentheses", output = 'results.docx')
```

This will create a Word file called `results.docx`.[6] I also added a note at the bottom of the table to indicate that the numbers in parentheses are standard errors (as that may not always be obvious). A specific path where to save the

[5] The American Psychological Association style guide indicates that numbers that cannot be greater than one (e.g., correlation coefficients, p values) should not have leading zeroes.

[6] There are other output formats as well that include .html, .jpg, or .txt. The .html output can be opened and edited using Microsoft Word or Excel. See the help file for more supported formats.

file can be specified (e.g., `output = wordfiles/results.docx`) as well. If the path is not specified, to find where the file is located, you can just run the `getwd()` function in the console.

The Word file may not look 100% like what you want, but now it is now in a format that is easily editable (and you don't have to copy and paste all the regression coefficients, standard errors, etc.). You may have to resize the columns, change the font, rename the predictors, spell out abbreviations, and add borders/lines as well to conform with whatever publication standards/guidelines you are using. Also, remember that when using dummy codes, the reference category should be indicated as well. Some of these may actually be done also using syntax (with some workarounds and additional packages) but as a practical guide, it's probably faster to just edit this in Word. Don't forget that tables convey a large amount of information and should be able to stand on their own (i.e., readers do not have to consult the text to understand it) so edit accordingly. At the very least, I would remove the the `RMSE` and the `Num.Obs.` output (it is redundant as it is the same for all the models and I would indicate the N in the title [or footnote] of the table instead).

.2.4 Using robust standard errors

One of the reasons that I like the `modelsummary` function is that it is easy to incorporate robust standard errors (as we have discussed in Chapters 4 and 5). In order to do this, we need to save the results (using the `robust_mixed` function in the `MLMusingR` package) in another object and put those in a list (see output in Table 3):

```
rob0 <- robust_mixed(nullm)
rob1 <- robust_mixed(lev1only)
rob2 <- robust_mixed(fullm)
modelsummary(list(rob0, rob1, rob2), stars = TRUE,
 notes = "Notes. Robust standard errors within parentheses",
 gof_omit = 'RMSE|IC|Obs', coef_omit = "SD",
 title = 'Multilevel Regression Model Results for Student
 Engagement (N = 528).')
```

Note that this will work only if using the `MLMusingR` package that is version `0.3.0` or higher. To check the version of the package installed, enter `packageVersion("MLMusingR")` in the console. If using an older version (e.g., `0.2.0`), just simply reinstall the package.

If using `clubSandwich` to obtain robust standard errors, we need to save the robust variance/covariance matrix (which have the standard errors on the square root of the diagonal of the matrix) and use it in place of the model-based variance/covariance matrix. We can save the robust matrices to a list called `robustse`.

Huang, F. (2023). Practical multilevel modeling using R. Sage.

Table 3: Multilevel Regression Model Results for Student Engagement (N = 528).

	Model 1	Model 2	Model 3
(Intercept)	8.777***	2.735	4.816*
	(0.196)	(1.675)	(1.782)
mot		0.545***	0.545***
		(0.146)	(0.143)
gpa		0.505**	0.536**
		(0.163)	(0.160)
gradegr7		−1.194***	−1.212***
		(0.225)	(0.222)
gradegr8		−1.525***	−1.562***
		(0.343)	(0.341)
frpm			−0.035**
			(0.010)
rural			0.519+
			(0.307)
R2 Marg.	0.000	0.089	0.133
R2 Cond.	0.078	0.157	0.159
Log.Lik.	−1354.949	−1332.191	−1327.746

Notes. Robust standard errors within parentheses
+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

```
library(clubSandwich)
vc1 <- as.matrix(vcovCR(lev1only, type = 'CR2', cluster = engage$school))
vc2 <- as.matrix(vcovCR(fullm, type = 'CR2', cluster = engage$school))
robustse <- list(vc1, vc2)
```

We then pass this to the `modelsummary` function using the `vcov` option. I am also just showing two models as the unconditional model merely has the intercept.

```
modelsummary(list("Level-1 variables only" = lev1only,
 "Full model" = fullm), stars = TRUE, gof_map = NA,
 notes = "Notes. Robust standard errors within parentheses",
 vcov = robustse, coef_omit = "SD")
```

Note when using the `vcov` option, that even though the robust standard errors may be shown, the p values (signified by the stars) may be slightly off (with small samples). So, make sure you inspect the results from the original output and the `modelsummary` output and make sure they match (readers can compare the results on their own and see that there are differences). A reason may be that when specifying the variance/covariance matrix in this case, the robust standard errors are used but then the degrees of freedom (which are used when computing the p values) may be off.

.2.5 Working with logistic regression models

Next to linear regression models, logistic regression models may be the most common type of model fit. However, reporting the regression coefficients of a logistic regression model (i.e., in log odds) may not be all that interpretable compared to the commonly used odds ratio (OR). The OR is computed by exponentiating the regression coefficient. However, if ORs are used, the confidence intervals (not the standard errors) should be reported and, at the same time, the intercept can be excluded as this is not an OR (and can be misinterpreted as such). As an example, we fit two models:

```
data(suspend) #from MLMusingR
s1 <- glmer(sus ~ male + frpl + fight + (1|school), family = binomial,
 data = suspend)
s2 <- update(s1, . ~ . + frpm.c + pminor.c)
```

Note: I also use the `update` function (described in Chapter 7) to keep things short and simple. To use the `modelsummary` function to get the ORs and the confidence intervals, the `exponentiate = TRUE` option is included as well as the `statistic = "({conf.low}, {conf.high})"` option which shows the 95% confidence intervals (CIs). We see that the coefficients with the CIs that do not contain 1 are statistically significant (see Table 4).

Table 4: Multilevel Logistic Regression Model Results Predicting Suspensions Using Odds Ratios (ORs)

	Model 1	Model 2
male	1.448***	1.455***
	(1.191, 1.760)	(1.196, 1.769)
frpl	1.971***	1.881***
	(1.617, 2.403)	(1.536, 2.303)
fight	9.362***	9.294***
	(7.719, 11.354)	(7.662, 11.273)
frpm.c		1.008
		(0.997, 1.019)
pminor.c		1.003
		(0.996, 1.010)
Num.Obs.	8465	8465
R2 Marg.	0.228	0.241
R2 Cond.	0.296	0.308
AIC	3396.5	3395.9
BIC	3431.8	3445.2
ICC	0.09	0.09
RMSE	0.23	0.23

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

```
modelsummary(list(s1, s2),
 statistic = "({conf.low}, {conf.high})",
 exponentiate = TRUE, stars = TRUE,
 coef_omit = "Intercept",
 title = 'Multilevel Logistic Regression Model Results
 Predicting Suspensions Using Odds Ratios (ORs)'
)
```

If you want to omit the goodness-of-fit measures, you can also add: `gof_map = NA` as was done in earlier examples. You should mention (in the table title or in a note) that the results are shown using *OR*s and the numbers in the parentheses are the 95% CIs.

We have just touched the surface of the customizations that are possible with the `modelsummary` package. Regression results can also be visualized using the `modelplot` function (try it out: `modelplot(fullm)`) though its use is not as common as the regression tables shown.[7]

[7]See https://vincentarelbundock.github.io/modelsummary/articles/modelplot.html for examples of using the function.

.3 Summary

This section began by discussing the different aspects of what can be reported when conducting multilevel analyses. One of the goals when reporting results is to guide readers with regard to all the choices made to arrive at the model results. Being transparent in what was done allows readers to have more faith in your study findings. Exporting regression model table results (including some customizations to the output) to different formats (e.g., on-screen, Microsoft Word) was also shown using the `modelsummary` package (which avoids having to manually cut-and-paste model results into a table). After exporting the results, make sure to attend to some other formatting requirements (e.g., renaming variables, adding footnotes, adding table borders) based on the publication style guide you are following.

References

Arel-Bundock, V., Gassen, J., Eastwood, N., Huntington-Klein, N., Schwarz, M., Elbers (0000-0001-5392-3448), B., … Wallrich, L. (2022). *modelsummary: Summary tables and plots for statistical models and data: Beautiful, customizable, and publication-ready*. Retrieved from https://CRAN.R-project.org/package=modelsummary

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

Chakraborty, H., & Hossain, A. (2018). R package to estimate intracluster correlation coefficient with confidence interval for binary data. *Computer Methods and Programs in Biomedicine*, *155*, 85–92. https://doi.org/10.1016/j.cmpb.2017.10.023

Dedrick, R. F., Ferron, J. M., Hess, M. R., Hogarty, K. Y., Kromrey, J. D., Lang, T. R., … Lee, R. S. (2009). Multilevel modeling: A review of methodological issues and applications. *Review of Educational Research*, *79*(1), 69–102. Retrieved from http://rer.sagepub.com/content/79/1/69.short

Depaoli, S., & van de Schoot, R. (2017). Improving transparency and replication in Bayesian statistics: The WAMBS-Checklist. *Psychological Methods*, *22*, 240–261. https://doi.org/10.1037/met0000065

Gelman, A. (2014). How do we choose our default methods? In X. Lin, C. Genest, D. Banks, G. Molenberghs, D. Scott, & J. Wang (Eds.), *Past, Present, and Future of Statistical Science* (pp. 317–326). Chapman; Hall/CRC. https://doi.org/10.1201/b16720-33

Hlavac, M. (2022). *stargazer: Well-formatted regression and summary statistics tables*. Retrieved from https://CRAN.R-project.org/package=stargazer

Huang, F. L. (2016). Alternatives to multilevel modeling for the analysis of clustered data. *Journal of Experimental Education*, *84*, 175–196. https://doi.org/10.1080/00220973.2014.952397

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, *82*(13), 1–26. https://doi.org/10.18637/jss.v082.i13

LaHuis, D. M., & Ferguson, M. W. (2009). The accuracy of significance tests for slope variance components in multilevel random coefficient models. *Organizational Research Methods*, *12*(3), 418–435.

LaHuis, D. M., Hartman, M. J., Hakoyama, S., & Clark, P. C. (2014). Explained variance measures for multilevel models. *Organizational Research Methods*, *17*(4), 433–451. https://doi.org/10.1177/1094428114541701

Leifeld, P., & Zucca, C. (2022). *texreg: Conversion of R regression output to LaTeX or HTML tables*. Retrieved from https://CRAN.R-project.org/package=texreg

Long, J. A. (2020). *jtools: Analysis and presentation of social scientific data*. Retrieved from https://cran.r-project.org/package=jtools

Luke, S. G. (2017). Evaluating significance in linear mixed-effects models in R. *Behavior Research Methods*, *49*(4), 1494–1502. https://doi.org/10.3758/s13428-016-0809-y

Luo, W., Li, H., Baek, E., Chen, S., Lam, K. H., & Semma, B. (2021). Reporting practice in multilevel modeling: A revisit after 10 Years. *Review of Educational Research*, *91*(3), 311–355. https://doi.org/10.3102/0034654321991229

McCoach, D. B., Rifenbark, G. G., Newton, S. D., Li, X., Kooken, J., Yomtov, D., … Bellara, A. (2018). Does the package matter? A comparison of five common multilevel modeling software packages. *Journal of Educational and Behavioral Statistics*, *43*(5), 594–627.

Monsalves, M. J., Bangdiwala, A. S., Thabane, A., & Bangdiwala, S. I. (2020). LEVEL (Logical Explanations & Visualizations of Estimates in Linear mixed models): Recommendations for reporting multilevel data and analyses. *BMC Medical Research Methodology*, *20*(1), 1–9.

Nakagawa, S., & Schielzeth, H. (2013). A general and simple method for obtaining R^2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, *4*(2), 133–142. https://doi.org/10.1111/j.2041-210x.2012.00261.x

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Sjoberg, D. D., Whiting, K., Curry, M., Lavery, J. A., & Larmarange, J. (2021). Reproducible summary tables with the gtsummary package. *The R Journal*, *13*, 570–580. https://doi.org/10.32614/RJ-2021-053