Online Supplemental Materials

A Simple Monte Carlo Method for Estimating Power in Multilevel Designs

Craig K. Enders, Brian T. Keller, Michael P. Woller,

Contents

Illustration 1 Vignette: Cross-Sectional Power Analysis

The first illustration demonstrates a power analysis for a cross-sectional application of the following multilevel model. To provide a substantive context, consider a prototypical education example where students are nested in schools. Intraclass correlations for achievement-related outcomes often range between .10 and .25 (Hedges & Hedberg, 2007; Hedges & Hedberg, 2013; Sellström & Bremberg, 2006; Spybrook et al., 2011; Stockford, 2009). To accommodate uncertainty about this important parameter, the power simulations investigate intraclass correlation values of .10 and .25.

The multilevel model for the illustration is

$$\begin{gathered}
Y_{ij} = \left(\beta_0 + b_{0j}\right) + \left(\beta_1 + b_{1j}\right)\left(X_{1ij} - \overline{X}_{1j}\right) + \beta_2\left(X_{2ij} - \overline{X}_2\right) \\
+ \beta_3\left(Z_{1j} - \overline{Z}_1\right) + \beta_4\left(Z_{2j} - \overline{Z}_2\right) + \beta_5\left(X_{1ij} - \overline{X}_{1j}\right)\left(Z_{1j} - \overline{Z}_1\right) + \varepsilon_{ij} \\
\boldsymbol{b}_j \sim N(0, \boldsymbol{\Sigma_b}) \qquad \boldsymbol{\Sigma_b} = \begin{pmatrix} \sigma^2_{b_0} & \\ \sigma_{b_1,b_0} & \sigma^2_{b_1} \end{pmatrix} \qquad \varepsilon_{ij} \sim N(0, \sigma^2_\varepsilon)
\end{gathered} \tag{1}$$

where Y_{ij} is the outcome score for observation i in cluster j, X_{1ij} is a focal within-cluster predictor, $\overline{X}_{1j}$ is the variable's level-2 cluster mean, X_{2ij} is a grand mean centered level-1 covariate, Z_{1j} is a level-2 moderator score for cluster j, and Z_{2j} is a level-2 covariate. Turning to the residual terms, b_{0j} and b_{1j} are between-cluster random effects that capture residual variation in the cluster-specific intercepts and slopes, and ε_{ij} is a within-cluster residual. By assumption, the random effects follow a multivariate normal distribution with a between-cluster covariance matrix $\boldsymbol{\Sigma_b}$, and within-cluster residuals are normal with constant variance σ^2_ε.

Importantly, group mean centering X_1 yields a pure within-cluster predictor, whereas grand mean centering X_2 gives a predictor with two sources of variation. Although it need not be true in practice, the within- and between-cluster parts of any level-1 predictor variables with non-zero intraclass correlations share a common slope coefficient. To simplify notation, Equation 1 can be rewritten as

$$Y_{ij} = \left(\beta_0 + b_{0j}\right) + \left(\beta_1 + b_{1j}\right)X_{1ij}^{w} + \beta_2\left(X_{2ij}^{w} + X_{2j}^{b}\right) + \beta_3 Z_{1j}^{b} + \beta_4 Z_{2j}^{b} + \beta_5 X_{1ij}^{w} Z_{1j}^{b} + \varepsilon_{ij} \quad (2)$$

where the w and b superscripts reference within- and between-cluster deviation scores, respectively.

The key inputs to the model object are the unconditional intraclass correlation and effect size values. This illustration investigates power for intraclass correlation values of .10 and .25. The within- and between-cluster fixed effects are set to R_w^2 = .065 and R_b^2 = .065, respectively, the sum of which corresponds to Cohen's (1988) medium effect size benchmark. Note that R_b^2 cannot exceed the outcome's intraclass correlation, as this would imply that between-cluster predictors explain more than 100% of the available variance. Following conventional wisdom that interaction effects tend to produce small effects (Aguinis, Beaty, Boik, & Pierce, 2005; Chaplin, 1991), the cross-level product is assigned R_p^2 = .01. Finally, the random coefficient effect size is set to R_{rc}^2 = .03 based on values from Table 2 in Enders, Keller, and Woller (2023).

By default, `mlmsim` assigns equal weights to all quantities contributing to a particular source of variance. To assign non-uniform weights, assume X_1 and Z_1 (the interacting variables) are the focal predictors and X_2 and Z_2 are covariates. To mimic a situation where the covariates explain a small amount of variation, the fixed effect R^2

values are predominantly allocated to the focal predictors using weights of .80 and .20. A small covariate allocation could be appropriate for a set of background or sociodemographic characteristics that weakly predict the outcome. Researchers often need power estimates for individual partial regression slopes. Although the weights do not exactly carve R^2_w and R^2_b into additive components when predictors are correlated, adopting weak associations among the regressors allows us to infer that each focal predictor roughly accounts for 5% of the explained variation at each level (i.e., $.80 \times R^2_w$ or $R^2_b \approx .05$).

The following code block shows the `mlmsim` model object for the example.

```
example1 <- (
    effect_size(
        icc          = c(.10,.25),
        within       = .065,
        between      = .065,
        product      = .01,
        random_slope = .03
    )
    + outcome('y', mean = 50, sd = 10)
    + within_predictor('x1', icc = 0, weight = .80)
    + within_predictor('x2', weight = .20)
    + between_predictor('z1', weight = .80)
    + between_predictor('z2', weight = .20)
    + product('x1','z1', weight = 1)
    + random_slope('x1', weight = 1)
)
```

The `mlmsim` package groups R^2 values and intraclass correlations into a single object called `effect_size()`, with the five inputs separated by commas. The `within` argument corresponds to R^2_w, the `between` argument aligns with R^2_b, the `product` argument specifies R^2_p, and the `random_slope` argument corresponds to R^2_{rc}. The `icc` argument assigns a global intraclass correlation to all level-1 variables, and separate simulations are performed for each requested level.

Variable attributes are referenced by adding the following objects: `outcome()`, `within_predictor()`, `between_predictor()`, `product()`, and `random_slope()`. All five objects have an argument for the variable name, weight (`weight =`), mean (`mean =`), and standard deviation (`sd =`). Level-1 variables additionally have an intraclass correlation argument (`icc =`) that supersedes the global setting in `effect_size()`.

The previous code block assigns explicit weights to all variables contributing to a given effect. The unit weights in the `product()` and `random_slope()` objects result from a single variable determining those sources of variation. Next, the `within_predictor('x1', icc = 0, weight = .80)` object overrides the global intraclass correlation setting, defining X_1 as a pure within-cluster deviation variable with no between-cluster variation. Finally, with the exception of the outcome variable, which has a mean and standard deviation of 50 and 10, the script accepts the default settings for the means and variances of the predictors (0 and 1, respectively).

The multilevel model parameters also require three sets of correlations: within-cluster correlations among level-1 predictors, between-cluster correlations among level-2 predictors, and random effect correlations. These correlations are optional, but they can specified using the `correlations()` object. The earlier code block omits this object, thereby accepting the default specification. When the `correlations()` object is omitted, the `mlmsim` package iteratively samples all correlations from a uniform distribution between .10 and .30, such that the resulting power estimates average over a distribution of possible associations. This default range spans Cohen's (1988) small to medium effect size benchmarks, and it brackets common correlation values from published research

(Bosco, Aguinis, Singh, Field, & Pierce, 2015; Funder & Ozer, 2019; Gignac & Szodorai, 2016).

The default specifications could be invoked by explicitly appending the `correlations()` object to the earlier script, as follows.

```
example1 <- (
    effect_size(
        icc          = c(.10,.25),
        within       = .065,
        between      = .065,
        product      = .01,
        random_slope = .03
    )
    + outcome('y', mean = 50, sd = 10)
    + within_predictor('x1', icc = 0, weight = .80)
    + within_predictor('x2', weight = .20)
    + between_predictor('z1', weight = .80)
    + between_predictor('z2', weight = .20)
    + product('x1','z1', weight = 1)
    + random_slope('x1', weight = 1)
    + correlations(
        within  = random(0.1, 0.3),
        between = random(0.1, 0.3),
        randeff = random(0.1, 0.3)
    )
)
```

Researchers can modify the upper and lower limits of each correlation range, or they can specify a constant correlation by inputting a scalar value. For example, specifying `randeff = 0` would define $\boldsymbol{\Sigma}_b$ as a diagonal matrix. The illustrative simulations presented in Enders et al. (2023) suggest that predictor and random effect correlations tend not to matter very much.

It may be instructive to inspect the population parameters prior to running the simulation. Executing `summary(example1)` returns tabular summaries of the multilevel model parameters.

```
! Parameters are solved based on average correlation.
$`icc = 0.1`
$r2
                                          Proportion
Variance Within-Cluster Fixed Effects    0.065000000
Variance Cross-Level Interaction Effects 0.010000000
Variance Random Slopes                   0.030000000
Within-Cluster Error Variance            0.795000000
Variance Between-Cluster Fixed Effects   0.071476494
Incremental Variance Level-2 Predictors  0.065000000
Between Variance Level-1 Covariates      0.006476494
Variance Random Intercepts               0.028523506

$phi_b
           x2_b         z1         z2
x2_b 0.10000000 0.06324555 0.06324555
z1   0.06324555 1.00000000 0.20000000
z2   0.06324555 0.20000000 1.00000000

$phi_p
        x1_w*z1
x1_w*z1       1

$phi_w
          x1_w      x2_w
x1_w 1.0000000 0.1897367
x2_w 0.1897367 0.9000000

$mean_Z
z1 z2
 0  0

$mean_X
x1 x2
 0  0

$var_e
```

```
          Value
Res. Var.  79.5

$tau
          Icept      x1_w
Icept 2.8523506 0.5850488
x1_w  0.5850488 3.0000000

$gammas
             Value
Icept   50.0000000
x1_w     2.3646137
x2_w     0.6231304
x1_w*z1  1.0000000
x2_b     0.6231304
z1       2.4308622
z2       0.6077155

$mean_Y
 y
50


$`icc = 0.25`
$r2
                                          Proportion
Variance Within-Cluster Fixed Effects    0.065000000
Variance Cross-Level Interaction Effects 0.010000000
Variance Random Slopes                   0.030000000
Within-Cluster Error Variance            0.645000000
Variance Between-Cluster Fixed Effects   0.074006354
Incremental Variance Level-2 Predictors  0.065000000
Between Variance Level-1 Covariates      0.009006354
Variance Random Intercepts               0.175993646

$phi_b
     x2_b  z1  z2
x2_b 0.25 0.1 0.1
z1   0.10 1.0 0.2
z2   0.10 0.2 1.0

$phi_p
        x1_w*z1
x1_w*z1       1
```

```
$phi_w
          x1_w      x2_w
x1_w 1.0000000 0.1732051
x2_w 0.1732051 0.7500000

$mean_Z
z1 z2
 0  0

$mean_X
x1 x2
 0  0

$var_e
          Value
Res. Var.  64.5

$tau
          Icept     x1_w
Icept 17.599365 1.453246
x1_w   1.453246 3.000000

$gammas
             Value
Icept   50.0000000
x1_w     2.3646137
x2_w     0.6826052
x1_w*z1  1.0000000
x2_b     0.6826052
z1       2.4308622
z2       0.6077155

$mean_Y
 y
50
```

Having specified the target model, you next use the `power_analysis()` function to conduct simulations. The function requires four inputs: the `model` argument specifies the parameter value object (e.g., `example1`), the `replications` input specifies the number of artificial data sets, `n_between` is a vector of level-2 sample sizes, and `n_within` is a vector of level-1 sample sizes (i.e., the number of observations per cluster). The code block below pairs four level-2 sample size values (J = 30, 60, 90, and 120) with three level-1 sample sizes (n_j = 10, 20, or 30 observations per cluster), and it requests 2,000 artificial data sets for each combination.

```
powersim <-
     power_analysis(
        model = example1,
        replications = 2000,
        n_between = c(30,60,90,120),
        n_within = c(10,20,30)
      )
```

The package uses `lme4` (Bates et al., 2021) for model fitting, and it defaults to an alpha level of .05 for all significance tests. Significance tests of random slope variation use a likelihood ratio test with a mixture chi-square reference distribution (i.e., a chi-bar distribution; Snijders & Bosker, 2012, p. 99), as implemented in the `varTestnlme` package (Baey & Kuhn, 2022). Executing `summary(powersim)` returns the tabular summaries of the simulation results shown below.

```
── Power Analysis Specification ──

Replications = 2000
N_Within = 10, 20, and 30
N_Between = 30, 60, 90, and 120

── lme4 Model Syntax

y ~ 1 + cwc(x1) + cgm(x2) + cwc(x1):cgm(z1) + cgm(z1) + cgm(z2) + (1 + cwc(x1)
| `_id`)

ℹ cwc() = centering within cluster
ℹ cgm() = centering with grand mean
```

── Effect Sizes

• WITHIN = 0.065
• BETWEEN = 0.065
• RANDOM SLOPE = 0.03
• PRODUCT = 0.01

── Power Analysis Results ──

```
── global icc = 0.1, n_within = 10, n_between = 30
                              Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         0.92 ± 0.01
Fixed:  cgm(x2)         0.22 ± 0.02
Fixed:  cgm(z1)         0.91 ± 0.01
Fixed:  cgm(z2)         0.14 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.32 ± 0.02
Random: Slopes Test     0.16 ± 0.02

── global icc = 0.25, n_within = 10, n_between = 30
                              Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         0.96 ± 0.01
Fixed:  cgm(x2)         0.25 ± 0.02
Fixed:  cgm(z1)         0.66 ± 0.02
Fixed:  cgm(z2)         0.10 ± 0.01
Fixed:  cwc(x1):cgm(z1) 0.33 ± 0.02
Random: Slopes Test     0.25 ± 0.02

── global icc = 0.1, n_within = 20, n_between = 30
                              Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         0.99 ± 0.00
Fixed:  cgm(x2)         0.37 ± 0.02
Fixed:  cgm(z1)         0.98 ± 0.01
Fixed:  cgm(z2)         0.23 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.48 ± 0.02
Random: Slopes Test     0.45 ± 0.02

── global icc = 0.25, n_within = 20, n_between = 30
                              Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
```

```
Fixed:  cgm(x2)         0.42 ± 0.02
Fixed:  cgm(z1)         0.71 ± 0.02
Fixed:  cgm(z2)         0.11 ± 0.01
Fixed:  cwc(x1):cgm(z1) 0.51 ± 0.02
Random: Slopes Test     0.62 ± 0.02

── global icc = 0.1, n_within = 30, n_between = 30
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.52 ± 0.02
Fixed:  cgm(z1)         0.99 ± 0.00
Fixed:  cgm(z2)         0.28 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.57 ± 0.02
Random: Slopes Test     0.73 ± 0.02

── global icc = 0.25, n_within = 30, n_between = 30
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.59 ± 0.02
Fixed:  cgm(z1)         0.74 ± 0.02
Fixed:  cgm(z2)         0.12 ± 0.01
Fixed:  cwc(x1):cgm(z1) 0.61 ± 0.02
Random: Slopes Test     0.84 ± 0.02

── global icc = 0.1, n_within = 10, n_between = 60
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.40 ± 0.02
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.26 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.56 ± 0.02
Random: Slopes Test     0.32 ± 0.02

── global icc = 0.25, n_within = 10, n_between = 60
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.44 ± 0.02
Fixed:  cgm(z1)         0.93 ± 0.01
Fixed:  cgm(z2)         0.16 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.61 ± 0.02
```

```
Random: Slopes Test     0.44 ± 0.02

── global icc = 0.1, n_within = 20, n_between = 60
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.63 ± 0.02
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.41 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.77 ± 0.02
Random: Slopes Test     0.78 ± 0.02

── global icc = 0.25, n_within = 20, n_between = 60
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.70 ± 0.02
Fixed:  cgm(z1)         0.96 ± 0.01
Fixed:  cgm(z2)         0.17 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.82 ± 0.02
Random: Slopes Test     0.89 ± 0.01

── global icc = 0.1, n_within = 30, n_between = 60
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.81 ± 0.02
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.48 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.86 ± 0.02
Random: Slopes Test     0.95 ± 0.01

── global icc = 0.25, n_within = 30, n_between = 60
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.85 ± 0.02
Fixed:  cgm(z1)         0.97 ± 0.01
Fixed:  cgm(z2)         0.17 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.89 ± 0.01
Random: Slopes Test     0.99 ± 0.00

── global icc = 0.1, n_within = 10, n_between = 90
                        Power
```

```
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.55 ± 0.02
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.37 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.73 ± 0.02
Random: Slopes Test     0.46 ± 0.02

── global icc = 0.25, n_within = 10, n_between = 90
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.58 ± 0.02
Fixed:  cgm(z1)         0.98 ± 0.01
Fixed:  cgm(z2)         0.21 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.80 ± 0.02
Random: Slopes Test     0.64 ± 0.02

── global icc = 0.1, n_within = 20, n_between = 90
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.81 ± 0.02
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.55 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.93 ± 0.01
Random: Slopes Test     0.91 ± 0.01

── global icc = 0.25, n_within = 20, n_between = 90
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.87 ± 0.01
Fixed:  cgm(z1)         0.99 ± 0.00
Fixed:  cgm(z2)         0.24 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.95 ± 0.01
Random: Slopes Test     0.98 ± 0.01

── global icc = 0.1, n_within = 30, n_between = 90
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.94 ± 0.01
Fixed:  cgm(z1)         1.00 ± 0.00
```

```
Fixed:  cgm(z2)         0.62 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.97 ± 0.01
Random: Slopes Test     0.99 ± 0.00

── global icc = 0.25, n_within = 30, n_between = 90
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.96 ± 0.01
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.23 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.97 ± 0.01
Random: Slopes Test     1.00 ± 0.00

── global icc = 0.1, n_within = 10, n_between = 120
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.66 ± 0.02
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.47 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.86 ± 0.02
Random: Slopes Test     0.58 ± 0.02

── global icc = 0.25, n_within = 10, n_between = 120
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.73 ± 0.02
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.24 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.91 ± 0.01
Random: Slopes Test     0.75 ± 0.02

── global icc = 0.1, n_within = 20, n_between = 120
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.90 ± 0.01
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.66 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.98 ± 0.01
Random: Slopes Test     0.97 ± 0.01
```

```
── global icc = 0.25, n_within = 20, n_between = 120
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.94 ± 0.01
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.29 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.99 ± 0.01
Random: Slopes Test     1.00 ± 0.00

── global icc = 0.1, n_within = 30, n_between = 120
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.98 ± 0.01
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.75 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.99 ± 0.00
Random: Slopes Test     1.00 ± 0.00

── global icc = 0.25, n_within = 30, n_between = 120
                        Power
Fixed:  (Intercept)     1.00 ± 0.00
Fixed:  cwc(x1)         1.00 ± 0.00
Fixed:  cgm(x2)         0.99 ± 0.00
Fixed:  cgm(z1)         1.00 ± 0.00
Fixed:  cgm(z2)         0.31 ± 0.02
Fixed:  cwc(x1):cgm(z1) 0.99 ± 0.00
Random: Slopes Test     1.00 ± 0.00

ℹ Margin of error (MOE) computed as ± 1.96 * standard error
```

Illustration 2 Vignette: Growth Curve Power Analysis

The second illustration demonstrates a power analysis for a longitudinal growth curve model with a pair of cross-level interactions involving a binary level-2 moderator. Intraclass correlations for longitudinal and intensive repeated measures data often reach values of .40 or higher (Arend & Schäfer, 2019; Bolger & Laurenceau, 2013; Singer & Willett, 2003). To accommodate uncertainty about this important parameter, the simulation investigates intraclass correlation values of .40 and .60.

The multilevel model for the illustration is

$$\begin{gathered} Y_{ij} = \left(\beta_0 + b_{0j}\right) + \left(\beta_1 + b_{1j}\right) X_{1ij}^{w} + \beta_2\left(X_{2ij}^{w} + X_{2j}^{b}\right) + \beta_3\left(X_{3ij}^{w} + X_{3j}^{b}\right) \\ + \beta_4 Z_{1j}^{b} + \beta_5 Z_{2j}^{b} + \beta_6 Z_{3j}^{b} + \beta_7 X_{1ij}^{w} Z_{1j}^{b} + \varepsilon_{ij} \\ \boldsymbol{b}_j \sim N(0, \boldsymbol{\Sigma_b}) \quad \boldsymbol{\Sigma_b} = \begin{pmatrix} \sigma_{b_0}^2 & \\ \sigma_{b_1,b_0} & \sigma_{b_1}^2 \end{pmatrix} \quad \varepsilon_{ij} \sim N(0, \sigma_{\varepsilon}^2) \end{gathered} \tag{3}$$

Following established notation (see Equations 1 and 2), the w and b superscripts reference within- and between-cluster deviation scores, respectively. The explanatory variables include a time score predictor with a random coefficient, a pair of time-varying covariates, a binary level-2 moderator, a pair of level-2 covariates, and a cross-level (group-by-time) interaction. For the purposes of weighting, we designated X_1 and Z_1 (the interacting variables) as focal predictors, X_2 and X_3 as level-1 (time-varying) covariates, and Z_2 and Z_3 as level-2 covariates. To mimic the scaling of a typical temporal index, we assume the major time increments are coded X_1^{w} = (0, 1, 2, 3, 4).

A brief discussion of Z_1 (the binary moderator) is warranted before continuing. First, like other level-2 variables, this variable's population mean must be 0 (i.e., Z_1 is centered in the population) in order to maintain the orthogonality of the cross-level interaction terms. Grand mean centering a level-2 binary predictor creates an ANOVA-like contrast code, such that β_0 is the grand mean and β_4 is the predicted group mean difference when X_1^w (the time score predictor) equals 0. In this case, a code of 0 corresponds to the baseline assessment. Because β_4 is a conditional effect that represents the group mean difference at the first occasion, Z_1's contribution to the between-cluster effect size depends on whether we view this regressor as a naturally occurring classification or an intervention assignment indicator. In the former case, we might expect a group mean difference at baseline, and the presence of such a difference would require a non-zero weight. In contrast, random assignment to conditions would eliminate a group mean difference at baseline, and Z_1's weight would equal 0. Note that this conclusion changes if the time scores are coded differently. For illustration purposes, we assume Z_1 is an intervention assignment indicator.

The key inputs to the model object are the unconditional intraclass correlation and effect size values. This illustration investigates power for intraclass correlation values of .40 and .60. The within- and between-cluster fixed effects are set to $R_w^2 = .13$ and $R_b^2 = .065$, respectively; the former corresponds to Cohen's (1988) medium effect size benchmark. Note that R_b^2 cannot exceed the outcome's intraclass correlation, as this would imply that between-cluster predictors explain more than 100% of the available variance. Following conventional wisdom that interaction effects tend to produce small effects (Aguinis et al., 2005; Chaplin, 1991), $R_p^2 = .05$ is assigned to the pair of cross-level product terms. Finally, the random coefficient effect size is set to $R_{rc}^2 = .03$ based on values from Table 2 in Enders et al. (2023).

To refresh, we designated X_1 and Z_1 (the interacting variables) as focal predictors, X_2 and X_3 as level-1 (time-varying) covariates, and Z_2 and Z_3 as level-2 covariates. To mimic a situation where the linear change predominates the level-1 model, we used weights of .50, .25, .25 to allocate the within-cluster R^2 to X_1, X_2, and X_3. At level-2, we used weights equal to 0, .50, .50 to allocate the between-cluster R^2 to Z_1, Z_2, and Z_3. As noted previously, Z_1's slope represents the group mean difference at baseline, and we are assuming that random assignment nullifies this effect. Finally, R^2_p and R^2_{rc} do not require weights because a single variable determines each source of variation.

To illustrate how to modify default correlation settings, we sampled within-cluster predictor correlations between the range of .20 and .40 under the assumption that the time scores could have stronger associations with other time-varying predictors. We similarly sampled the random effect correlations between the range of .30 and .50 to mimic a scenario where higher baseline scores (i.e., random intercepts) are associated with higher (more positive) growth rates. Finally, we adopted the default correlation range for the between-cluster predictors. The simulations from the previous cross-sectional example suggest that the correlations are somewhat arbitrary and would not have a material impact on power estimates.

The following code block shows the `mlmpower` model object for this example.

```
example2 <- (
  effect_size(
    icc          = c(.40,.60),
    within       = .13,
    between      = .065,
    product      = .03,
    random_slope = .10
  )
  + outcome('y', mean = 50, sd = 10)
  + within_time_predictor('x1', weight = .50, values = 0:4)
  + within_predictor('x2', weight = .25)
```

```
  + within_predictor('x3', weight = .25)
  + between_binary_predictor('z1', proportion = .50, weight = 0)
  + between_predictor('z2', weight = .50)
  + between_predictor('z3', weight = .50)
  + product('x1','z1', weight = 1)
  + random_slope('x1', weight = 1)
  + correlations(
    within  = random(.20, .40),
    between = random(.10, .30),
    randeff = random(.30, .50)
  )
)
```

The `mlmsim` package groups R^2 values and intraclass correlations into a single object called `effect_size()`, with the five inputs separated by commas. The `within` argument corresponds to R^2_w, the `between` argument aligns with R^2_b, the `product` argument specifies R^2_p, and the `random_slope` argument corresponds to R^2_{rc}. The `icc` argument assigns a global intraclass correlation to all level-1 variables, and separate simulations are performed for each requested level.

Variable attributes are referenced by adding the following base objects: `outcome()`, `within_predictor()`, `between_predictor()`, `product()`, and `random_slope()`. All five objects have an argument for the variable name, weight (`weight =`), mean (`mean =`), and standard deviation (`sd =`). Level-1 variables additionally have an intraclass correlation argument (`icc =`) that supersedes the global setting in `effect_size()`.

This illustration additionally uses the `within_time_predictor()` object to specify a set of fixed time scores for X_1, and it uses `between_binary_predictor()` object to define the level-2 moderator Z_1 as a binary predictor. In addition to a name and weight, the `within_time_predictor()` object requires a vector of time scores as an argument (`values =`). The `between_binary_predictor()` object requires a name, weight, and the highest category proportion (`proportion =`).

First, the `within_time_predictor()` object specifies X_1 as a temporal predictor with the fixed set of time scores described earlier. The `values = 0:4` argument specifies an integer sequence, but unequally spaced increments can also be specified using a vector as input (e.g., `values = c(0,1,3,6)`). This object does not require a mean or standard deviation argument, as these quantities are determined from the time scores. Additionally, the variable's intraclass correlation is automatically fixed to 0 because the time scores are constant across level-2 units. Next the `between_binary_predictor('z1', proportion = .50, weight = 0)` object specifies Z_1 (the moderator variable) as a binary predictor with a 50/50 split. This object does not require a mean or standard deviation argument, as the category proportions determine these quantities. Finally, within the exception of the outcome, the code block uses default values for all means and standard deviations (0 and 1, respectively). The means and standard deviations of the time scores and binary predictor are automatically determined by the user inputs.

The multilevel model parameters also require three sets of correlations: within-cluster correlations among level-1 predictors, between-cluster correlations among level-2 predictors, and random effect correlations. These correlations are optional, but they can specified using the `correlations()` object. This example samples within-cluster predictor correlation values between .20 and .40 (e.g., to mimic a situation where the time scores have salient correlations with other repeated measures predictors). Random effect correlation values are similarly sampled between the range of .30 and .50 to mimic a scenario where higher baseline scores (i.e., random intercepts) are associated with higher (more positive) growth rates. Finally, the script specifies the default setting for between-cluster correlations, which is to iteratively sample correlation values between .10 and .30.

It may be instructive to inspect the population parameters prior to running the simulation. Executing `summary(example2)` returns tabular summaries of the multilevel model parameters.

```
! Parameters are solved based on average correlation.
$`icc = 0.4`
$r2
                                          Proportion
Variance Within-Cluster Fixed Effects    0.13000000
Variance Cross-Level Interaction Effects 0.03000000
Variance Random Slopes                   0.10000000
Within-Cluster Error Variance            0.34000000
Variance Between-Cluster Fixed Effects   0.12474120
Incremental Variance Level-2 Predictors  0.06500000
Between Variance Level-1 Covariates      0.05974120
Variance Random Intercepts               0.03964212

$phi_b
                x2_b       x3_b  z1_binary         z2         z3
x2_b      0.40000000 0.12000000 0.06324555 0.12649111 0.12649111
x3_b      0.12000000 0.40000000 0.06324555 0.12649111 0.12649111
z1_binary 0.06324555 0.06324555 0.25000000 0.07978846 0.07978846
z2        0.12649111 0.12649111 0.07978846 1.00000000 0.20000000
z3        0.12649111 0.12649111 0.07978846 0.20000000 1.00000000

$phi_p
               x1_w*z1_binary
x1_w*z1_binary            0.5

$phi_w
          x1_w      x2_w      x3_w
x1_w 2.0000000 0.3286335 0.3286335
x2_w 0.3286335 0.6000000 0.1800000
x3_w 0.3286335 0.1800000 0.6000000

$mean_Z
 z1  z2  z3
```

```
0.5 0.0 0.0

$mean_X
x1 x2 x3
 2  0  0

$var_e
          Value
Res. Var.    34

$tau
         Icept     x1_w
Icept 3.964212 1.780834
x1_w  1.780834 5.000000

$gammas
                   Value
Icept          46.600654
x1_w            1.699673
x2_w            1.551582
x3_w            1.551582
x1_w*z1_binary  2.449490
x2_b            1.551582
x3_b            1.551582
z2              1.737198
z3              1.737198

$mean_Y
 y
50


$`icc = 0.6`
$r2
                                         Proportion
Variance Within-Cluster Fixed Effects     0.1300000
Variance Cross-Level Interaction Effects  0.0300000
Variance Random Slopes                    0.1000000
Within-Cluster Error Variance             0.1400000
Variance Between-Cluster Fixed Effects    0.1696753
Incremental Variance Level-2 Predictors   0.0650000
Between Variance Level-1 Covariates       0.1046753
Variance Random Intercepts                0.1589955
```

```
$phi_b
                x2_b       x3_b  z1_binary         z2         z3
x2_b      0.60000000 0.18000000 0.07745967 0.15491933 0.15491933
x3_b      0.18000000 0.60000000 0.07745967 0.15491933 0.15491933
z1_binary 0.07745967 0.07745967 0.25000000 0.07978846 0.07978846
z2        0.15491933 0.15491933 0.07978846 1.00000000 0.20000000
z3        0.15491933 0.15491933 0.07978846 0.20000000 1.00000000

$phi_p
               x1_w*z1_binary
x1_w*z1_binary            0.5

$phi_w
          x1_w      x2_w      x3_w
x1_w 2.0000000 0.2683282 0.2683282
x2_w 0.2683282 0.4000000 0.1200000
x3_w 0.2683282 0.1200000 0.4000000

$mean_Z
 z1  z2  z3
0.5 0.0 0.0

$mean_X
x1 x2 x3
 2  0  0

$var_e
          Value
Res. Var.    14

$tau
         Icept    x1_w
Icept 15.89955 3.56646
x1_w   3.56646 5.00000

$gammas
                   Value
Icept          46.600654
x1_w            1.699673
x2_w            1.900292
x3_w            1.900292
x1_w*z1_binary  2.449490
x2_b            1.900292
x3_b            1.900292
```

```
z2              1.737198
z3              1.737198

$mean_Y
 y
50
```

Having specified the target model, you next use the `power_analysis()` function to conduct simulations. The function requires four inputs: the `model` argument specifies the parameter value object (e.g., `example2`), the `replications` input specifies the number of artificial data sets, `n_between` is a vector of level-2 sample sizes, and `n_within` is a vector of level-1 sample sizes (i.e., the number of observations per cluster). The code block below specifies six level-2 sample size conditions (J = 50, 60, 70, 80, 90, and 100), each with five repeated measurements and 2,000 replications.

```
powersim <-
     power_analysis(
        model = example2,
        replications = 2000,
        n_between = c(50,60,70,80,90,100),
        n_within = 5
      )
```

The package uses `lme4` (Bates et al., 2021) for model fitting, and it defaults to an alpha level of .05 for all significance tests. Significance tests of random slope variation use a likelihood ratio test with a mixture chi-square reference distribution (i.e., a chi-bar distribution; Snijders & Bosker, 2012, p. 99), as implemented in the `varTestnlme` package (Baey & Kuhn, 2022). Executing `summary(powersim)` returns the tabular summaries of the simulation results shown below.

```
── Power Analysis Specification ──

Replications = 2000
N_Within = 5
N_Between = 50, 60, 70, 80, 90, and 100

── lme4 Model Syntax

y ~ 1 + x1 + cgm(x2) + cgm(x3) + x1:cgm(z1) + cgm(z1) + cgm(z2) + cgm(z3) + (1
+ x1 | `_id`)

ℹ cwc() = centering within cluster
ℹ cgm() = centering with grand mean

── Effect Sizes

• WITHIN = 0.13
• BETWEEN = 0.065
• RANDOM SLOPE = 0.1
• PRODUCT = 0.03

── Power Analysis Results ──

── global icc = 0.4, n_within = 5, n_between = 50
                        Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          0.97 ± 0.01
Fixed:  cgm(x2)     0.84 ± 0.02
Fixed:  cgm(x3)     0.85 ± 0.02
Fixed:  cgm(z1)     0.05 ± 0.01
Fixed:  cgm(z2)     0.68 ± 0.02
Fixed:  cgm(z3)     0.66 ± 0.02
Fixed:  x1:cgm(z1)  0.81 ± 0.02
Random: Slopes Test 1.00 ± 0.00

── global icc = 0.6, n_within = 5, n_between = 50
                        Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          0.99 ± 0.00
Fixed:  cgm(x2)     0.99 ± 0.00
Fixed:  cgm(x3)     0.99 ± 0.00
Fixed:  cgm(z1)     0.05 ± 0.01
Fixed:  cgm(z2)     0.64 ± 0.02
```

```
Fixed:  cgm(z3)     0.62 ± 0.02
Fixed:  x1:cgm(z1)  0.91 ± 0.01
Random: Slopes Test 1.00 ± 0.00

── global icc = 0.4, n_within = 5, n_between = 60
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          0.99 ± 0.01
Fixed:  cgm(x2)     0.91 ± 0.01
Fixed:  cgm(x3)     0.90 ± 0.01
Fixed:  cgm(z1)     0.05 ± 0.01
Fixed:  cgm(z2)     0.75 ± 0.02
Fixed:  cgm(z3)     0.77 ± 0.02
Fixed:  x1:cgm(z1)  0.89 ± 0.01
Random: Slopes Test 1.00 ± 0.00

── global icc = 0.6, n_within = 5, n_between = 60
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          1.00 ± 0.00
Fixed:  cgm(x2)     1.00 ± 0.00
Fixed:  cgm(x3)     0.99 ± 0.00
Fixed:  cgm(z1)     0.05 ± 0.01
Fixed:  cgm(z2)     0.71 ± 0.02
Fixed:  cgm(z3)     0.72 ± 0.02
Fixed:  x1:cgm(z1)  0.96 ± 0.01
Random: Slopes Test 1.00 ± 0.00

── global icc = 0.4, n_within = 5, n_between = 70
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          1.00 ± 0.00
Fixed:  cgm(x2)     0.94 ± 0.01
Fixed:  cgm(x3)     0.93 ± 0.01
Fixed:  cgm(z1)     0.05 ± 0.01
Fixed:  cgm(z2)     0.82 ± 0.02
Fixed:  cgm(z3)     0.83 ± 0.02
Fixed:  x1:cgm(z1)  0.94 ± 0.01
Random: Slopes Test 1.00 ± 0.00

── global icc = 0.6, n_within = 5, n_between = 70
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          1.00 ± 0.00
```

```
Fixed:  cgm(x2)     1.00 ± 0.00
Fixed:  cgm(x3)     1.00 ± 0.00
Fixed:  cgm(z1)     0.06 ± 0.01
Fixed:  cgm(z2)     0.77 ± 0.02
Fixed:  cgm(z3)     0.79 ± 0.02
Fixed:  x1:cgm(z1)  0.98 ± 0.01
Random: Slopes Test 1.00 ± 0.00

── global icc = 0.4, n_within = 5, n_between = 80
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          1.00 ± 0.00
Fixed:  cgm(x2)     0.97 ± 0.01
Fixed:  cgm(x3)     0.97 ± 0.01
Fixed:  cgm(z1)     0.05 ± 0.01
Fixed:  cgm(z2)     0.88 ± 0.01
Fixed:  cgm(z3)     0.88 ± 0.01
Fixed:  x1:cgm(z1)  0.96 ± 0.01
Random: Slopes Test 1.00 ± 0.00

── global icc = 0.6, n_within = 5, n_between = 80
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          1.00 ± 0.00
Fixed:  cgm(x2)     1.00 ± 0.00
Fixed:  cgm(x3)     1.00 ± 0.00
Fixed:  cgm(z1)     0.05 ± 0.01
Fixed:  cgm(z2)     0.84 ± 0.02
Fixed:  cgm(z3)     0.83 ± 0.02
Fixed:  x1:cgm(z1)  0.99 ± 0.00
Random: Slopes Test 1.00 ± 0.00

── global icc = 0.4, n_within = 5, n_between = 90
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          1.00 ± 0.00
Fixed:  cgm(x2)     0.98 ± 0.01
Fixed:  cgm(x3)     0.98 ± 0.01
Fixed:  cgm(z1)     0.05 ± 0.01
Fixed:  cgm(z2)     0.90 ± 0.01
Fixed:  cgm(z3)     0.91 ± 0.01
Fixed:  x1:cgm(z1)  0.98 ± 0.01
Random: Slopes Test 1.00 ± 0.00
```

```
── global icc = 0.6, n_within = 5, n_between = 90
                     Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          1.00 ± 0.00
Fixed:  cgm(x2)     1.00 ± 0.00
Fixed:  cgm(x3)     1.00 ± 0.00
Fixed:  cgm(z1)     0.05 ± 0.01
Fixed:  cgm(z2)     0.87 ± 0.01
Fixed:  cgm(z3)     0.88 ± 0.01
Fixed:  x1:cgm(z1)  0.99 ± 0.00
Random: Slopes Test 1.00 ± 0.00

── global icc = 0.4, n_within = 5, n_between = 100
                     Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          1.00 ± 0.00
Fixed:  cgm(x2)     0.99 ± 0.01
Fixed:  cgm(x3)     0.98 ± 0.01
Fixed:  cgm(z1)     0.06 ± 0.01
Fixed:  cgm(z2)     0.94 ± 0.01
Fixed:  cgm(z3)     0.94 ± 0.01
Fixed:  x1:cgm(z1)  0.99 ± 0.00
Random: Slopes Test 1.00 ± 0.00

── global icc = 0.6, n_within = 5, n_between = 100
                     Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  x1          1.00 ± 0.00
Fixed:  cgm(x2)     1.00 ± 0.00
Fixed:  cgm(x3)     1.00 ± 0.00
Fixed:  cgm(z1)     0.05 ± 0.01
Fixed:  cgm(z2)     0.92 ± 0.01
Fixed:  cgm(z3)     0.90 ± 0.01
Fixed:  x1:cgm(z1)  1.00 ± 0.00
Random: Slopes Test 1.00 ± 0.00

ℹ Margin of error (MOE) computed as ± 1.96 * standard error
```

Illustration 3 Vignette: Cluster-Randomized Design

The third vignette demonstrates a power analysis for a cluster-randomized design (Raudenbush, 1997) where level-2 units are randomly assigned to one of two experimental conditions. To provide a substantive context, consider a prototypical education example where students are nested in schools, and schools are the unit of randomization. Intraclass correlations for achievement-related outcomes often range between .10 and .25 (Hedges & Hedberg, 2007; Hedges & Hedberg, 2013; Sellström & Bremberg, 2006; Spybrook et al., 2011; Stockford, 2009). To accommodate uncertainty about this important parameter, the simulation investigates intraclass correlation values of .10 and .25.

The multilevel model for the illustration is

$$\begin{aligned} Y_{ij} &= \beta_0 + \beta_1\left(X_{1ij}^{w} + X_{1j}^{b}\right) + \beta_2\left(X_{2ij}^{w} + X_{2j}^{b}\right) \\ &+ \beta_3\left(X_{3ij}^{w} + X_{3j}^{b}\right) + \beta_4\left(X_{4ij}^{w} + X_{4j}^{b}\right) + \beta_5 Z_{1j}^{b} + b_{0j} + \varepsilon_{ij} \end{aligned} \tag{4}$$

where X_1 to X_4 are grand mean centered level-1 covariates, and Z_1 is a binary intervention assignment indicator. Following established notation, the w and b superscripts reference within- and between-cluster deviation scores, respectively. The notation conveys that all level-1 predictors contain within- and between-cluster variation. By default, all predictors are centered, including the binary dummy code. Grand mean centering a level-2 binary predictor creates an ANOVA-like contrast code, such that β_0 is the grand mean and β_5 is the predicted mean difference, adjusted for the covariates. Turning to the residual terms, b_{0j} is a between-cluster random effect that captures residual variation in the cluster-specific intercepts, and ε_{ij} is a within-cluster residual. By assumption, both residuals are normal with constant variance.

The key inputs to the model object are the unconditional intraclass correlation and effect size values. This illustration investigates power for intraclass correlation values of .10 and .25. To mimic a scenario with a strong covariate set (e.g., one that includes a pretest measure of the outcome), the within-cluster effect size is set to R_w^2 = .18 (a value roughly midway between Cohen's small and medium benchmarks). Most of this variation was allocated to X_1 (the pretest) by assigning it a weight of .70, and the remaining three predictors had their within-cluster weights set to .10. We previously argued that the allocation of the weights among covariates is arbitrary because these predictors are not the focus. To demonstrate that point, we conducted a second simulation that weighted the four level-1 covariates equally.

Turning to the level-2 predictor, researchers often prefer the Cohen's (1988) d effect size when working with binary explanatory variables. To illustrate, consider power for d = .40, the midway point between Cohen's small and medium benchmarks. Substituting this value into the conversion equation below returns R_b^2 = .038.

$$R^2 = \frac{d^2}{d^2 + 4} \tag{5}$$

The following code block shows the `mlmsim` model object for the example.

```
example3 <- (
    effect_size(
        icc          = c(.10,.25),
        within       = .18,
        between      = .038,
```

```
    )
    + outcome('y')
    + within_predictor('x1', weight = .70)
    + within_predictor('x2', weight = .10)
    + within_predictor('x3', weight = .10)
    + within_predictor('x4', weight = .10)
    + between_binary_predictor('z1', proportion = .50, weight = 1)
)
```

The `mlmsim` package groups R^2 values and intraclass correlations into a single object called `effect_size()`, with three inputs separated by commas. The `within` argument corresponds to R^2_w, the `between` argument aligns with R^2_b, and the `icc` argument assigns a global intraclass correlation to all level-1 variables. Separate simulations are performed for each requested level.

Variable attributes for this example require three objects: `outcome()`, `within_predictor()`, and `between_binary_predictor()`. The first two objects have an argument for the variable name, weight (`weight =`), mean (`mean =`), standard deviation (`sd =`), and an intraclass correlation argument (`icc =`) that supersedes the global setting in `effect_size()`. The `between_binary_predictor()` object requires a name, weight, and the highest category proportion (`proportion =`).

The previous code block assigns explicit weights to all variables contributing to a given effect, as described previously. The unit weight in the `between_binary_predictor()` object reflects the fact that this Z_1 solely determines R^2_b. Finally, the `proportion` argument assigns a 50/50 split to the level-2 groups.

The multilevel model parameters also require two sets of correlations: within-cluster correlations among level-1 predictors, and between-cluster correlations among level-2 predictors (in this case, the cluster means and the intervention assignment indicator). These correlations are optional, but they can specified using the `correlations()` object. The earlier code block omits this object, thereby accepting the default specification. When the `correlations()` object is omitted, the `mlmsim` package iteratively samples all correlations from a uniform distribution between .10 and .30, such that the resulting power estimates average over a distribution of possible associations. This default range spans Cohen's (1988) small to medium effect size benchmarks, and it also brackets common correlations from published research (Bosco et al., 2015; Funder & Ozer, 2019; Gignac & Szodorai, 2016). The default specifications could be invoked by explicitly appending the `correlations()` object to the earlier script, as follows.

```
example3 <- (
    effect_size(
        icc          = c(.10,.25),
        within       = .18,
        between      = .038,
    )
    + outcome('y')
    + within_predictor('x1', weight = .70)
    + within_predictor('x2', weight = .10)
    + within_predictor('x3', weight = .10)
    + within_predictor('x4', weight = .10)
    + between_binary_predictor('z1', proportion = .50, weight = 1)
    + correlations(
        within  = random(0.1, 0.3),
        between = random(0.1, 0.3)
    )
)
```

Researchers can modify the upper and lower limits of each correlation range, or they can specify a constant correlation by inputting a scalar value. The illustrative simulations presented in Enders et al. (2023) suggest that predictor correlations tend not to matter very much.

It may be instructive to inspect the population parameters prior to running the simulation. Executing `summary(example3)` returns tabular summaries of the multilevel model parameters.

```
! Parameters are solved based on average correlation.
$`icc = 0.1`
$r2
                                          Proportion
Variance Within-Cluster Fixed Effects    0.18000000
Variance Cross-Level Interaction Effects 0.00000000
Variance Random Slopes                   0.00000000
Within-Cluster Error Variance            0.72000000
Variance Between-Cluster Fixed Effects   0.07703223
Incremental Variance Level-2 Predictors  0.03800000
Between Variance Level-1 Covariates      0.03903223
Variance Random Intercepts               0.02296777

$phi_b
                x1_b       x2_b       x3_b       x4_b  z1_binary
x1_b      0.10000000 0.02000000 0.02000000 0.02000000 0.03162278
x2_b      0.02000000 0.10000000 0.02000000 0.02000000 0.03162278
x3_b      0.02000000 0.02000000 0.10000000 0.02000000 0.03162278
x4_b      0.02000000 0.02000000 0.02000000 0.10000000 0.03162278
z1_binary 0.03162278 0.03162278 0.03162278 0.03162278 0.25000000

$phi_p
<0 x 0 matrix>

$phi_w
     x1_w x2_w x3_w x4_w
x1_w 0.90 0.18 0.18 0.18
x2_w 0.18 0.90 0.18 0.18
x3_w 0.18 0.18 0.90 0.18
x4_w 0.18 0.18 0.18 0.90

$mean_Z
 z1
0.5

$mean_X
x1 x2 x3 x4
```

```
 0  0  0  0

$var_e
          Value
Res. Var.    72

$tau
         Icept
Icept 2.296777

$gammas
               Value
Icept     50.0000000
x1_w       3.9886202
x2_w       0.5698029
x3_w       0.5698029
x4_w       0.5698029
x1_b       3.9886202
x2_b       0.5698029
x3_b       0.5698029
x4_b       0.5698029
z1_binary  4.1096093

$mean_Y
 y
50


$`icc = 0.25`
$r2
                                         Proportion
Variance Within-Cluster Fixed Effects     0.1800000
Variance Cross-Level Interaction Effects  0.0000000
Variance Random Slopes                    0.0000000
Within-Cluster Error Variance             0.5700000
Variance Between-Cluster Fixed Effects    0.1278739
Incremental Variance Level-2 Predictors   0.0380000
Between Variance Level-1 Covariates       0.0898739
Variance Random Intercepts                0.1221261

$phi_b
          x1_b x2_b x3_b x4_b z1_binary
x1_b      0.25 0.05 0.05 0.05      0.05
x2_b      0.05 0.25 0.05 0.05      0.05
```

```
x3_b      0.05 0.05 0.25 0.05      0.05
x4_b      0.05 0.05 0.05 0.25      0.05
z1_binary 0.05 0.05 0.05 0.05      0.25

$phi_p
<0 x 0 matrix>

$phi_w
     x1_w x2_w x3_w x4_w
x1_w 0.75 0.15 0.15 0.15
x2_w 0.15 0.75 0.15 0.15
x3_w 0.15 0.15 0.75 0.15
x4_w 0.15 0.15 0.15 0.75

$mean_Z
 z1
0.5

$mean_X
x1 x2 x3 x4
 0  0  0  0

$var_e
          Value
Res. Var.    57

$tau
         Icept
Icept 12.21261

$gammas
               Value
Icept     50.0000000
x1_w       4.3693145
x2_w       0.6241878
x3_w       0.6241878
x4_w       0.6241878
x1_b       4.3693145
x2_b       0.6241878
x3_b       0.6241878
x4_b       0.6241878
z1_binary  4.1096093

$mean_Y
```

```
 y
50
```

Having specified the target model, you next use the `power_analysis()` function to conduct the simulations. The function requires four inputs: the `model` argument specifies the parameter value object (e.g., `example3`), the `replications` input specifies the number of artificial data sets, `n_between` is a vector of level-2 sample sizes, and `n_within` is a vector of level-1 sample sizes (i.e., the number of observations per cluster). The code block below pairs four level-2 sample size values (J = 30, 60, 90, and 120) with two level-1 sample sizes (n_j = 15 or 30 observations per cluster), and it requests 2,000 artificial data sets for each combination.

```
powersim <-
     power_analysis(
        model = example3,
        replications = 2000,
        n_between = c(30,60,90,120),
        n_within = c(15,30)
      )
```

The package uses `lme4` (Bates et al., 2021) for model fitting, and it defaults to an alpha level of .05 for all significance tests. Executing `summary(powersim)` returns the tabular summaries of the simulation results shown below.

```
── Power Analysis Specification ──

Replications = 2000
N_Within = 15 and 30
N_Between = 30, 60, 90, and 120

── lme4 Model Syntax
```

```
y ~ 1 + cgm(x1) + cgm(x2) + cgm(x3) + cgm(x4) + cgm(z1) + (1 | `_id`)

ℹ cwc() = centering within cluster
ℹ cgm() = centering with grand mean

── Effect Sizes

• WITHIN = 0.18
• BETWEEN = 0.038
• RANDOM SLOPE = 0
• PRODUCT = 0

── Power Analysis Results ──

── global icc = 0.1, n_within = 15, n_between = 30
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.27 ± 0.02
Fixed:  cgm(x3)     0.29 ± 0.02
Fixed:  cgm(x4)     0.24 ± 0.02
Fixed:  cgm(z1)     0.98 ± 0.01

── global icc = 0.25, n_within = 15, n_between = 30
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.31 ± 0.02
Fixed:  cgm(x3)     0.32 ± 0.02
Fixed:  cgm(x4)     0.30 ± 0.02
Fixed:  cgm(z1)     0.76 ± 0.02

── global icc = 0.1, n_within = 30, n_between = 30
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.47 ± 0.02
Fixed:  cgm(x3)     0.46 ± 0.02
Fixed:  cgm(x4)     0.45 ± 0.02
Fixed:  cgm(z1)     1.00 ± 0.00

── global icc = 0.25, n_within = 30, n_between = 30
                    Power
```

```
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.53 ± 0.02
Fixed:  cgm(x3)     0.53 ± 0.02
Fixed:  cgm(x4)     0.56 ± 0.02
Fixed:  cgm(z1)     0.81 ± 0.02

── global icc = 0.1, n_within = 15, n_between = 60
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.44 ± 0.02
Fixed:  cgm(x3)     0.44 ± 0.02
Fixed:  cgm(x4)     0.46 ± 0.02
Fixed:  cgm(z1)     1.00 ± 0.00

── global icc = 0.25, n_within = 15, n_between = 60
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.53 ± 0.02
Fixed:  cgm(x3)     0.56 ± 0.02
Fixed:  cgm(x4)     0.54 ± 0.02
Fixed:  cgm(z1)     0.96 ± 0.01

── global icc = 0.1, n_within = 30, n_between = 60
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.75 ± 0.02
Fixed:  cgm(x3)     0.75 ± 0.02
Fixed:  cgm(x4)     0.74 ± 0.02
Fixed:  cgm(z1)     1.00 ± 0.00

── global icc = 0.25, n_within = 30, n_between = 60
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.83 ± 0.02
Fixed:  cgm(x3)     0.82 ± 0.02
Fixed:  cgm(x4)     0.83 ± 0.02
Fixed:  cgm(z1)     0.99 ± 0.00

── global icc = 0.1, n_within = 15, n_between = 90
```

```
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.61 ± 0.02
Fixed:  cgm(x3)     0.61 ± 0.02
Fixed:  cgm(x4)     0.60 ± 0.02
Fixed:  cgm(z1)     1.00 ± 0.00

── global icc = 0.25, n_within = 15, n_between = 90
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.71 ± 0.02
Fixed:  cgm(x3)     0.73 ± 0.02
Fixed:  cgm(x4)     0.71 ± 0.02
Fixed:  cgm(z1)     1.00 ± 0.00

── global icc = 0.1, n_within = 30, n_between = 90
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.88 ± 0.01
Fixed:  cgm(x3)     0.90 ± 0.01
Fixed:  cgm(x4)     0.89 ± 0.01
Fixed:  cgm(z1)     1.00 ± 0.00

── global icc = 0.25, n_within = 30, n_between = 90
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.95 ± 0.01
Fixed:  cgm(x3)     0.95 ± 0.01
Fixed:  cgm(x4)     0.93 ± 0.01
Fixed:  cgm(z1)     1.00 ± 0.00

── global icc = 0.1, n_within = 15, n_between = 120
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.75 ± 0.02
Fixed:  cgm(x3)     0.75 ± 0.02
Fixed:  cgm(x4)     0.76 ± 0.02
Fixed:  cgm(z1)     1.00 ± 0.00
```

```
── global icc = 0.25, n_within = 15, n_between = 120
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.83 ± 0.02
Fixed:  cgm(x3)     0.84 ± 0.02
Fixed:  cgm(x4)     0.84 ± 0.02
Fixed:  cgm(z1)     1.00 ± 0.00

── global icc = 0.1, n_within = 30, n_between = 120
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.95 ± 0.01
Fixed:  cgm(x3)     0.96 ± 0.01
Fixed:  cgm(x4)     0.95 ± 0.01
Fixed:  cgm(z1)     1.00 ± 0.00

── global icc = 0.25, n_within = 30, n_between = 120
                    Power
Fixed:  (Intercept) 1.00 ± 0.00
Fixed:  cgm(x1)     1.00 ± 0.00
Fixed:  cgm(x2)     0.99 ± 0.00
Fixed:  cgm(x3)     0.98 ± 0.01
Fixed:  cgm(x4)     0.99 ± 0.01
Fixed:  cgm(z1)     1.00 ± 0.00

ℹ Margin of error (MOE) computed as ± 1.96 * standard error
```

Cross-Sectional Power Estimates With Non-Zero and Zero Correlations

Table A1

Power Estimates From the Cross-Sectional Power Simulations (Paper Table 3)

Sample Sizes	$J = 30$ $n_j = 10$		$J = 30$ $n_j = 20$		$J = 30$ $n_j = 30$		$J = 60$ $n_j = 10$		$J = 60$ $n_j = 20$		$J = 60$ $n_j = 30$	
ICC	.10	.25	.10	.25	.10	.25	.10	.25	.10	.25	.10	.25
X_1^w Slope	.92	.96	.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_2^{w+b} Slope	.22	.25	.37	.42	.52	.59	.40	.44	.63	.70	.81	.85
Z_1^b Slope	.91	.66	.98	.71	.99	.74	1.00	.93	1.00	.96	1.00	.97
Z_2^b Slope	.14	.10	.23	.11	.28	.12	.26	.16	.41	.17	.48	.17
$X_1^w Z_1^b$ Slope	.32	.33	.48	.51	.57	.61	.56	.61	.77	.82	.86	.89
$\boldsymbol{\Sigma}_{\boldsymbol{b}(rc)}$	.16	.25	.45	.62	.73	.84	.32	.44	.78	.89	.95	.99
Sample Sizes	$J = 90$ $n_j = 10$		$J = 90$ $n_j = 20$		$J = 90$ $n_j = 30$		$J = 120$ $n_j = 10$		$J = 120$ $n_j = 20$		$J = 120$ $n_j = 30$	
ICC	.10	.25	.10	.25	.10	.25	.10	.25	.10	.25	.10	.25
X_1^w Slope	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_2^{w+b} Slope	.55	.58	.81	.87	.94	.96	.66	.73	.90	.94	.98	.99
Z_1^b Slope	1.00	.98	1.00	.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Z_2^b Slope	.37	.21	.55	.24	.62	.23	.47	.24	.66	.29	.75	.31
$X_1^w Z_1^b$ Slope	.73	.80	.93	.95	.97	.97	.86	.91	.98	.99	.99	.99
$\boldsymbol{\Sigma}_{\boldsymbol{b}(rc)}$	.46	.64	.91	.98	.99	1.00	.58	.75	.97	1.00	1.00	1.00

Note. ICC = intraclass correlation, J = the number of level-2 units, n_j = the number of level-1 observations within each level-2 cluster, and $\boldsymbol{\Sigma}_{\boldsymbol{b}(rc)}$ references the random slope variances and their corresponding covariances. Power estimates are the proportion of 2,000 data sets that produced a significant test statistic at $p < .05$.

Table A2

Power Estimates From the Cross-Sectional Power Illustration With Zero Correlations

Sample Sizes	$J = 30$ $n_j = 10$		$J = 30$ $n_j = 20$		$J = 30$ $n_j = 30$		$J = 60$ $n_j = 10$		$J = 60$ $n_j = 20$		$J = 60$ $n_j = 30$	
ICC	.10	.25	.10	.25	.10	.25	.10	.25	.10	.25	.10	.25
X_1^w Slope	.95	.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_2^{w+b} Slope	.23	.28	.41	.47	.55	.63	.40	.46	.67	.77	.84	.91
Z_1^b Slope	.92	.68	.98	.74	.99	.77	1.00	.94	1.00	.97	1.00	.97
Z_2^b Slope	.14	.10	.25	.11	.25	.11	.26	.15	.38	.18	.44	.19
$X_1^w Z_1^b$ Slope	.31	.35	.47	.53	.55	.60	.54	.65	.79	.82	.86	.90
$\boldsymbol{\Sigma}_{\boldsymbol{b}(rc)}$	.15	.22	.43	.56	.71	.84	.32	.41	.76	.88	.95	.98
Sample Sizes	$J = 90$ $n_j = 10$		$J = 90$ $n_j = 20$		$J = 90$ $n_j = 30$		$J = 120$ $n_j = 10$		$J = 120$ $n_j = 20$		$J = 120$ $n_j = 30$	
ICC	.10	.25	.10	.25	.10	.25	.10	.25	.10	.25	.10	.25
X_1^w Slope	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_2^{w+b} Slope	.54	.62	.84	.90	.95	.98	.67	.74	.92	.96	.98	.99
Z_1^b Slope	1.00	.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Z_2^b Slope	.36	.22	.53	.24	.61	.24	.49	.27	.64	.30	.74	.31
$X_1^w Z_1^b$ Slope	.75	.81	.93	.94	.97	.97	.87	.90	.98	.99	.99	1.00
$\boldsymbol{\Sigma}_{\boldsymbol{b}(rc)}$	.43	.57	.91	.96	.99	1.00	.56	.71	.97	.99	1.00	1.00

Note. ICC = intraclass correlation, J = the number of level-2 units, n_j = the number of level-1 observations within each level-2 cluster, and $\boldsymbol{\Sigma}_{\boldsymbol{b}(rc)}$ references the random slope variances and their corresponding covariances. Power estimates are the proportion of 2,000 data sets that produced a significant test statistic at $p < .05$.

Table A3

Power Estimates From the Cluster-Randomized Trial With Unequally Weighted Level-1 Covariates

Sample Sizes	$J = 30$, $n_j = 15$		$J = 30$, $n_j = 30$		$J = 60$, $n_j = 15$		$J = 60$, $n_j = 30$	
ICC	.10	.25	.10	.25	.10	.25	.10	.25
X_1^{w+b} Slope	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_2^{w+b} Slope	.27	.31	.47	.53	.44	.53	.75	.83
X_3^{w+b} Slope	.29	.32	.46	.53	.44	.56	.75	.82
X_4^{w+b} Slope	.24	.30	.45	.56	.46	.54	.74	.83
Z_1^{b} Slope	.98	.76	1.00	.81	1.00	.96	1.00	.99
Sample Sizes	$J = 90$, $n_j = 15$		$J = 90$, $n_j = 30$		$J = 120$, $n_j = 15$		$J = 120$, $n_j = 30$	
ICC	.10	.25	.10	.25	.10	.25	.10	.25
X_1^{w+b} Slope	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_2^{w+b} Slope	.61	.71	.88	.95	.75	.83	.95	.99
X_3^{w+b} Slope	.61	.73	.90	.95	.75	.84	.96	.98
X_4^{w+b} Slope	.60	.71	.89	.93	.76	.84	.95	.99
Z_1^{b} Slope	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Note. ICC = intraclass correlation, J = the number of level-2 units, and n_j = the number of level-1 observations within each level-2 cluster. Power estimates are the proportion of 2,000 data sets that produced a significant test statistic at $p < .05$.

Table A4

Power Estimates From the Cluster-Randomized Trial With Equally Weighted Level-1 Covariates

Sample Sizes	$J = 30$ $n_j = 15$		$J = 30$ $n_j = 30$		$J = 60$ $n_j = 15$		$J = 60$ $n_j = 30$	
ICC	.10	.25	.10	.25	.10	.25	.10	.25
X_1^{w+b} Slope	.98	.99	1.00	1.00	1.00	1.00	1.00	1.00
X_2^{w+b} Slope	.98	.99	1.00	1.00	1.00	1.00	1.00	1.00
X_3^{w+b} Slope	.98	.99	1.00	1.00	1.00	1.00	1.00	1.00
X_4^{w+b} Slope	.98	.99	1.00	1.00	1.00	1.00	1.00	1.00
Z_1^b Slope	.98	.78	1.00	.83	1.00	.97	1.00	1.00
Sample Sizes	$J = 90$ $n_j = 15$		$J = 90$ $n_j = 30$		$J = 120$ $n_j = 15$		$J = 120$ $n_j = 30$	
ICC	.10	.25	.10	.25	.10	.25	.10	.25
X_1^{w+b} Slope	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_2^{w+b} Slope	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_3^{w+b} Slope	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_4^{w+b} Slope	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Z_1^b Slope	.99	1.00	.99	1.00	1.00	1.00	1.00	1.00

Note. ICC = intraclass correlation, J = the number of level-2 units, and n_j = the number of level-1 observations within each level-2 cluster. Power estimates are the proportion of 2,000 data sets that produced a significant test statistic at $p < .05$.

Covariance Matrix of Cross-Level Interaction Terms

The coefficients solutions require the variance–covariance matrix of cross-level product terms. The covariance between a pair of cross-level product terms or a cross-level product term and another variable can be derived from Equation 13 in Bohrnstedt and Goldberger (1969), shown below.

$$\begin{aligned} C(XY, UV) \;=\; & E(X)E(U)C(Y,V) \;+\; E(X)E(V)C(Y,U) \;+\; E(Y)E(U)C(X,V) \\ & + E(Y)E(V)C(X,U) \;+\; C(X,U)C(Y,V) \;+\; C(X,V)C(Y,U) \end{aligned} \tag{A1}$$

For example, Equation 10 of the paper gives the expression needed to obtain the covariances between the level-1 predictor X_{ij}^{w} and the cross-level product term $X_{ij}^{w}W_{j}^{b}$. This quantity is obtained by setting $X = X_{ij}^{w}$, $Y = 1$, $U = X_{ij}^{w}$, and $V = W_{j}^{b}$, as follows.

$$\begin{aligned} C(X_{ij}^{w}, X_{ij}^{w}W_{j}^{b}) \;=\; & E(X_{ij}^{w})E(X_{ij}^{w})C(1, W_{j}^{b}) \;+\; E(X_{ij}^{w})E(W_{j}^{b})C(1, X_{ij}^{w}) \;+\; E(1)E(X_{ij}^{w})C(X_{ij}^{w}, W_{j}^{b}) \\ & + E(1)E(W_{j}^{b})C(X_{ij}^{w}, X_{ij}^{w}) \;+\; C(X_{ij}^{w}, X_{ij}^{w})C(1, W_{j}^{b}) \;+\; C(X_{ij}^{w}, W_{j}^{b})C(1, X_{ij}^{w}) \\ =\; & E\left(X_{ij}^{w}\right)C\left(X_{ij}^{w}, W_{j}^{b}\right) + E\left(W_{j}^{b}\right)C\left(X_{ij}^{w}, X_{ij}^{w}\right) \end{aligned} \tag{A2}$$

The equation simplifies to 0 because centering sets $E(X_{ij}^{w})$ and $E(W_{j}^{b})$ equal to 0 and because the covariance between a group mean centered level-1 predictor and a level-2 predictor equals 0 (e.g., $C(X_{ij}^{w}, W_{j}^{b}) = 0$).

To further illustrate, consider a multilevel model with four cross-level product terms: $X_{1ij}^{w}W_{1j}^{b}$, $X_{1ij}^{w}W_{2j}^{b}$, $X_{2ij}^{w}W_{1j}^{b}$, and $X_{2ij}^{w}W_{2j}^{b}$. The variance–covariance matrix of the product random variables is as follows.

$$\mathbf{\Phi}_p = \begin{pmatrix} C\left(X^w_{1ij}W^b_{1j}, X^w_{1ij}W^b_{1j}\right) & C\left(X^w_{1ij}W^b_{1j}, X^w_{1ij}W^b_{2j}\right) & C\left(X^w_{1ij}W^b_{1j}, X^w_{2ij}W^b_{1j}\right) & \left(X^w_{1ij}W^b_{1j}, X^w_{2ij}W^b_{2j}\right) \\ C\left(X^w_{1ij}W^b_{2j}, X^w_{1ij}W^b_{1j}\right) & C\left(X^w_{1ij}W^b_{2j}, X^w_{1ij}W^b_{2j}\right) & C\left(X^w_{1ij}W^b_{2j}, X^w_{2ij}W^b_{1j}\right) & C\left(X^w_{1ij}W^b_{2j}, X^w_{2ij}W^b_{2j}\right) \\ C\left(X^w_{2ij}W^b_{1j}, X^w_{1ij}W^b_{1j}\right) & C\left(X^w_{2ij}W^b_{1j}, X^w_{1ij}W^b_{2j}\right) & C\left(X^w_{2ij}W^b_{1j}, X^w_{2ij}W^b_{1j}\right) & C\left(X^w_{2ij}W^b_{2j}, X^w_{2ij}W^b_{1j}\right) \\ C\left(X^w_{2ij}W^b_{2j}, X^w_{1ij}W^b_{1j}\right) & C\left(X^w_{2ij}W^b_{2j}, X^w_{1ij}W^b_{2j}\right) & C\left(X^w_{2ij}W^b_{1j}, X^w_{2ij}W^b_{2j}\right) & C\left(X^w_{2ij}W^b_{2j}, X^w_{2ij}W^b_{2j}\right) \end{pmatrix} \quad \text{(A3)}$$

Applying Equation A1 gives the following solutions for the elements in $\mathbf{\Phi}_p$.

$$\mathbf{\Phi}_p = \begin{pmatrix} V\left(X^w_{1ij}\right)V\left(W^b_{1j}\right) & V\left(X^w_{1ij}\right)C\left(W^b_{1j}, W^b_{2j}\right) & C\left(X^w_{1ij}, X^w_{2ij}\right)V\left(W^b_{1j}\right) & C\left(X^w_{1ij}, X^w_{2ij}\right)C\left(W^b_{1j}, W^b_{2j}\right) \\ V\left(X^w_{1ij}\right)C\left(W^b_{2j}, W^b_{1j}\right) & V\left(X^w_{1ij}\right)V\left(W^b_{2j}\right) & C\left(X^w_{1ij}, X^w_{2ij}\right)C\left(W^b_{2j}, W^b_{1j}\right) & C\left(X^w_{1ij}, X^w_{2ij}\right)V\left(W^b_{2j}\right) \\ C\left(X^w_{2ij}, X^w_{1ij}\right)V\left(W^b_{1j}\right) & C\left(X^w_{1ij}, X^w_{2ij}\right)C\left(W^b_{1j}, W^b_{2j}\right) & V\left(X^w_{2ij}\right)V\left(W^b_{1j}\right) & V\left(X^w_{2ij}\right)C\left(W^b_{1j}, W^b_{2j}\right) \\ C\left(X^w_{2ij}, X^w_{1ij}\right)C\left(W^b_{2j}, W^b_{1j}\right) & C\left(X^w_{2ij}, X^w_{1ij}\right)V\left(W^b_{2j}\right) & V\left(X^w_{2ij}\right)C\left(W^b_{2j}, W^b_{1j}\right) & V\left(X^w_{2ij}\right)V\left(W^b_{2j}\right) \end{pmatrix} \quad \text{(A4)}$$

From this expression, it is evident that the covariance matrix can be written more succinctly as the Kronecker product of $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_b$.

$$\mathbf{\Phi}_p = \mathbf{\Phi}_w \otimes \mathbf{\Phi}_b = \begin{pmatrix} V\left(X^w_{1ij}\right) & C\left(X^w_{1ij}, X^w_{2ij}\right) \\ C\left(X^w_{2ij}, X^w_{1ij}\right) & V\left(X^w_{2ij}\right) \end{pmatrix} \otimes \begin{pmatrix} V\left(W^b_{1j}\right) & C\left(W^b_{1j}, W^b_{2j}\right) \\ C\left(W^b_{2j}, W^b_{1j}\right) & V\left(W^b_{2j}\right) \end{pmatrix} \quad \text{(A5)}$$

Solutions for Multilevel Model Parameters

Solving for the multilevel model parameters requires the following inputs beyond the R^2 values: the intraclass correlations of all level-1 variables, means and variances of all variables, within-cluster correlations among the level-1 predictors, between-cluster correlations among the level-2 predictors, and correlations among the random effects. Table 1 summarizes the inputs, catalogs our software's default settings, and provides recommendations for specifying values.

We focus on a two-level regression with a continuous outcome because this model enjoys widespread use in published applications. The following model is from Equation 1 in the paper.

$$\begin{gathered}
Y_{ij} = \left(\beta_0 + b_{0j}\right) + \left(\beta_1 + b_{1j}\right)\left(X_{1ij} - \overline{X}_{1j}\right) + \beta_2\left(X_{2ij} - \overline{X}_2\right) \\
+ \beta_3\left(X_{3j} - \overline{X}_3\right) + \beta_4\left(X_{4j} - \overline{X}_4\right) + \beta_5\left(X_{1ij} - \overline{X}_{1j}\right)\left(X_{3j} - \overline{X}_3\right) + \varepsilon_{ij} \\
\boldsymbol{b}_j \sim N(0, \boldsymbol{\Sigma_b}) \quad \boldsymbol{\Sigma_b} = \begin{pmatrix} \sigma^2_{b_0} & \\ \sigma^2_{b_1,b_0} & \sigma^2_{b_1} \end{pmatrix} \quad \varepsilon_{ij} \sim N(0, \sigma^2_\varepsilon)
\end{gathered} \tag{B6}$$

where Y_{ij} is the outcome score for observation i in cluster j, X_{1ij} is a focal within-cluster predictor, $\overline{X}_{1j}$ is the variable's level-2 cluster mean, X_{2ij} is a grand mean centered level-1 covariate, X_{3j} is a level-2 moderator score for cluster j, and X_{4j} is a level-2 covariate. Turning to the residual terms, b_{0j} and b_{1j} are between-cluster random effects that capture residual variation in the cluster-specific intercepts and slopes, and ε_{ij} is a within-cluster residual. By assumption, the random effects follow a multivariate normal distribution with a between-cluster covariance matrix $\boldsymbol{\Sigma}_b$, and within-cluster residuals are normal with constant variance σ^2_ε.

Table A4

mlmpower Package Inputs and Recommendations (Paper Table 1)

Input	Default	Recommendation
Within-cluster predictor R^2 (i.e., R^2_w)	None	Pilot data, previous studies, off-the-shelf values with sensitivity checks.
Cross-level product R^2 (i.e., R^2_p)	None	Pilot data, previous studies, off-the-shelf values with sensitivity checks.
Random coefficient R^2 (i.e., R^2_{rc})	None	Pilot data, Table 2 values with sensitivity checks
Between-cluster predictor R^2 (i.e., R^2_b)	None	Pilot data, previous studies, off-the-shelf values with sensitivity checks.
Outcome variable ICC	None	Pilot data, previous studies, off-the-shelf ICCs with sensitivity checks.
Level-1 predictor ICC (group mean centered)	0	Fixed quantity. Must be 0.
Level-1 predictor ICC (grand mean centered)	None	Pilot data, previous studies, off-the-shelf ICCs with sensitivity checks.
Outcome mean	50	Arbitrary.
Level-1 predictor means	0	Arbitrary. Time score predictors may require a non-zero mean.
Level-2 predictor means	0	Fixed quantity. Must be 0.
Level-2 binary predictor group proportions	.50/.50	Pilot data, previous studies, expert judgment.
Outcome variance	100	Arbitrary.
Level-1 predictor variances	1	Arbitrary. Time score predictors may require a different variance.
Level-2 predictor variances	1	Arbitrary.
Within-cluster predictor correlations	.10–.30	Somewhat arbitrary. Tends not to be very impactful.
Between-cluster predictor correlations	.10–.30	Somewhat arbitrary. Tends not to be very impactful.
Random effect correlations	.30	Somewhat arbitrary. Tends not to be very impactful.

Step 1: Variance–Covariance Matrices of Predictors

Along with the predictor correlations, the intraclass correlations and arbitrary variance values fully determine the variance–covariance matrices of the predictor variables, which then combine with the R^2 effect sizes from the previous section to give solutions for the multilevel model's parameters. To illustrate, the illustrative model for this example requires the following variance–covariance matrices

$$\mathbf{\Phi}_w = \begin{pmatrix} \left(1 - ICC_{X_1}\right)\text{var}\left(X_{1ij}\right) & \\ R(X_{2ij}^w, X_{1ij}^w)\sqrt{\left(1 - ICC_{X_2}\right)\text{var}\left(X_{2ij}\right)\left(1 - ICC_{X_1}\right)\text{var}\left(X_{1ij}\right)} & \left(1 - ICC_{X_2}\right)\text{var}\left(X_{2ij}\right) \end{pmatrix}$$

$$\mathbf{\Phi}_b = \begin{pmatrix} \text{var}\left(X_{3j}\right) & \\ R(X_{3j}, X_{4j})\sqrt{\text{var}\left(W_{1j}\right)\text{var}\left(W_{2j}\right)} & \text{var}\left(X_{4j}\right) \end{pmatrix} \tag{B7}$$

$$\mathbf{\Phi}_p = \mathbf{\Phi}_{w(p)} \otimes \mathbf{\Phi}_{b(p)}$$

where $R(X_{2ij}^w, X_{1ij}^w)$ is the within-cluster correlation between X_1 and X_2, $\mathbf{\Phi}_{w(p)}$ and $\mathbf{\Phi}_{b(p)}$ are submatrices of $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_b$ that contain the variances and covariances of the interacting variables, and $\otimes$ denotes a Kronecker product.

Step 2: Predictor Weighting

We obtain unique solutions for the model parameters by invoking weights that define the relative contribution of each predictor to a given source of explained variance. Assigning proportional weights to the pair of predictors imposes the constraints necessary to obtain unique solutions for β_1 and β_2. For example, specifying $\boldsymbol{\omega}_w = (.50, .50)$ as a weight vector implies that both predictors contribute equally to explained variation, whereas setting $\boldsymbol{\omega}_w = (.75, .25)$ means that X_1^w's unique contribution is about three times larger than X_2^w's. Analogous weights are required for the other effect sizes: $\boldsymbol{\omega}_p$ is the weight vector for cross-level product terms that contribute to R_p^2, $\boldsymbol{\omega}_b$ weights the contribution of the pure level-2 predictors to R_b^2, and $\boldsymbol{\omega}_{rc}$ weights the relative contributions of the random coefficient predictors to R_{rc}^2.

To account for variational differences among the predictors, we first divide each raw weight by the variable's standard deviation. To illustrate, consider $\boldsymbol{\omega}_w$, the weight vector for the level-1 predictors that contribute to R_w^2. The scaled weight vector is

$$\widetilde{\boldsymbol{\omega}}_w = \boldsymbol{\omega}_w \circ \mathrm{diag}(\boldsymbol{\Phi}_w)^{\circ -.5} \tag{B8}$$

where $\circ$ denotes an elementwise operation (i.e., a Hadamard product and Hadamard power). In this illustrative model from Equation 1, the rightmost term of the expression is a vector with $1/\sqrt{\boldsymbol{\Phi}_{w(1,1)}}$ and $1/\sqrt{\boldsymbol{\Phi}_{w(2,2)}}$ as its elements. The remaining weights are as follows.

$$\begin{aligned} \widetilde{\boldsymbol{\omega}}_b &= \boldsymbol{\omega}_b \circ \mathrm{diag}(\boldsymbol{\Phi}_b)^{\circ -.5} \\ \widetilde{\boldsymbol{\omega}}_p &= \boldsymbol{\omega}_p \circ \mathrm{diag}\left(\boldsymbol{\Phi}_p\right)^{\circ -.5} \\ \widetilde{\boldsymbol{\omega}}_{rc} &= \boldsymbol{\omega}_{rc} \circ \mathrm{diag}\left(\boldsymbol{\Phi}_{w(rc)}\right)^{\circ -1} \end{aligned} \tag{B9}$$

Step 3: Coefficient Solutions

Broadly speaking, the coefficient solutions are obtained as follows. First, when multiple predictors contribute to a given effect size expression, we reduce the multivariate information to a single dimension by forming a weighted linear composite of the predictors. This operation reduces the covariance matrix in each expression to a scalar variance term, and it simultaneously redefines each R^2 formula as a function of a single composite predictor variable. This data reduction step provides a unique solution to an intermediate parameter that links the composite predictor to a given source of variation. For example, when applying this procedure to $\boldsymbol{\beta}^w$, we obtain a single intermediate slope coefficient that represents the regression of the outcome on the weighted composite. Finally, multiplying the composite parameter by the weights distributes the global effect to the individual predictors.

This process yields the following solutions for the coefficients.

$$\boldsymbol{\beta}_w = \widetilde{\boldsymbol{\omega}}_w \sqrt{\frac{\text{var}\left(Y_{ij}\right) R_w^2}{\widetilde{\boldsymbol{\omega}}_w' \boldsymbol{\Phi}_w \widetilde{\boldsymbol{\omega}}_w}} \tag{B10}$$

$$\boldsymbol{\beta}_p = \widetilde{\boldsymbol{\omega}}_p \sqrt{\frac{\text{var}\left(Y_{ij}\right) R_p^2}{\widetilde{\boldsymbol{\omega}}_p' \boldsymbol{\Phi}_p \widetilde{\boldsymbol{\omega}}_p}} \tag{B11}$$

$$\boldsymbol{\beta}_b = \widetilde{\boldsymbol{\omega}}_b \sqrt{\frac{\text{var}\left(Y_{ij}\right) R_b^2}{\widetilde{\boldsymbol{\omega}}_b' \boldsymbol{\Phi}_b \widetilde{\boldsymbol{\omega}}_b}} \tag{B12}$$

The term involving the radical can be viewed as the slope for a linear composite of the predictors, and multiplying that quantity by the scaled weights apportions the overall effect to the individual regressors. Equation 5 applies to situations where are predictors are centered at their group means, such that within- and between- predictors account for unique variation. When level-1 predictors are grand mean centered, the solution requires the following expression.

$$\boldsymbol{\beta}_b = \widetilde{\boldsymbol{\omega}}_b \sqrt{\frac{\text{var}\left(Y_{ij}\right) R_b^2}{\widetilde{\boldsymbol{\omega}}_b' \left(\boldsymbol{\Phi}_{b(l2)} - \left(\left(\boldsymbol{\Phi}_{b(l1)}\right)^{-1} \boldsymbol{\Phi}_{b(l1,l2)}\right)' \boldsymbol{\Phi}_{b(l1,l2)}\right) \widetilde{\boldsymbol{\omega}}_b}} \tag{B13}$$

The expression partitions the between-cluster covariance matrix of the predictors as follows

$$\boldsymbol{\Phi}_b = \begin{pmatrix} \boldsymbol{\Phi}_{b(l1)} & \boldsymbol{\Phi}_{b(l1,l2)} \\ \boldsymbol{\Phi}_{b(l2,l1)} & \boldsymbol{\Phi}_{b(l2)} \end{pmatrix}. \tag{B14}$$

where $\boldsymbol{\Phi}_{b(l1)}$ contains variances and covariances of the cluster means, $\boldsymbol{\Phi}_{b(l2)}$ is the submatrix corresponding to the pure level-2 variables, and $\boldsymbol{\Phi}_{b(l1,l2)}$ contains covariances between the cluster means and pure level-2 variables.

$$\beta_0 = \mu_Y - \boldsymbol{\mu}_w' \boldsymbol{\beta}_w - \boldsymbol{\mu}_p' \boldsymbol{\beta}_p - \boldsymbol{\mu}_b' \boldsymbol{\beta}_b \tag{B15}$$

Note that $\boldsymbol{\mu}_p$ is a zero vector because the covariance between the interacting variables equals 0 and the means of level-2 variables equal 0. This follows from Equation 3 in Bohrnstedt and Goldberger (1969).

Step 4: Random Effect Variances and Covariances

Our solution for the variance–covariance matrix of the random slopes depends on the number of random coefficients in the model. With a single random slope predictor, the variation due to random slopes simplifies from $\text{tr}\left(\boldsymbol{\Sigma}_{b(rc)} \boldsymbol{\Phi}_{w(rc)}\right)$ to the product of the random slope variance and the level-1 predictor's variance.

$$\mathrm{tr}\left(\mathbf{\Sigma}_{\boldsymbol{b}(rc)}\mathbf{\Phi}_{w(rc)}\right) = \sigma^2_{b_1}\sigma^2_{X_1} \tag{B16}$$

This leads to the following solution for the random slope variance.

$$\sigma^2_{b_1} = \frac{\mathrm{var}\left(Y_{ij}\right)R^2_{rc}}{\mathrm{var}\left(X^w_{1ij}\right)} \tag{B17}$$

With more than one random coefficient, we assume that the random slopes are uncorrelated. That is, $\mathbf{R}_{rc} = \mathbf{I}_{rc}$, where $\mathbf{R}_{rc}$ is the correlation matrix of the random coefficients and $\mathbf{I}_{rc}$ is an identity matrix with the same dimensions. Some authors have argued that setting random effect correlations to 0 leads to conservative power estimates (Arend & Schäfer, 2019; Martin, Nussey, Wilson, & Réale, 2011), and others suggest that fixed effect coefficients are not very sensitive to these correlations (Snijders & Bosker, 1993). Finally, let $\widetilde{\boldsymbol{\omega}}_{rc}$ denote the scaled weight vector defined in Equation B4.

The first step expresses the covariance matrix $\mathbf{\Sigma}_{\boldsymbol{b}(rc)}$ in the leftmost term as the product of a correlation matrix and diagonal matrices containing standard deviations of the random slopes. Applying this substitution and rearranging terms gives the following.

$$\mathrm{tr}\left(\mathbf{\Sigma}_{\boldsymbol{b}(rc)}\mathbf{\Phi}_{w(rc)}\right) = \mathrm{tr}\left(\mathbf{S}_{rc}\mathbf{R}_{rc}\mathbf{S}_{rc}\mathbf{\Phi}_{w(rc)}\right) = \mathrm{tr}\left(\mathbf{S}_{rc}\mathbf{I}_{rc}\mathbf{S}_{rc}\mathbf{\Phi}_{w(rc)}\right) = \mathrm{tr}\left(\mathbf{S}_{rc}\mathbf{S}_{rc}\mathbf{\Phi}_{w(rc)}\right) \tag{B18}$$

Because $\mathbf{S}_{rc}$ is a diagonal matrix, the $\mathbf{S}_{rc}\mathbf{S}_{rc}$ product returns another diagonal matrix $\mathbf{V}_{rc}$ with random slope variances on the diagonal.

$$\mathrm{tr}\left(\mathbf{S}_{rc}\mathbf{S}_{rc}\mathbf{\Phi}_{w(rc)}\right) = \mathrm{tr}\left(\mathbf{V}_{rc}\mathbf{\Phi}_{w(rc)}\right) \tag{B19}$$

Next, note that $\mathbf{V}_{rc}$ can be expressed as the sum of the random slope variances (i.e., the trace of $\mathbf{\Sigma}_{\boldsymbol{b}(rc)}$) multiplied by a diagonal matrix with the random slope weights on the diagonal, $\mathbf{W} = \mathrm{diag}(\widetilde{\boldsymbol{\omega}}_{rc})$.

$$\mathbf{V}_{rc} = \mathrm{tr}(\mathbf{V}_{rc})\mathbf{W} = \mathrm{C}\mathbf{W} \tag{B20}$$

Finally, replacing $\mathbf{V}_{rc}$ in Equation B14 with the right side of Equation B15 then factoring out the constant gives the following.

$$\mathrm{tr}\left(\mathrm{C}\mathbf{W}\mathbf{\Phi}_{w(rc)}\right) = \mathrm{C} \times \mathrm{tr}\left(\mathbf{W}\mathbf{\Phi}_{w(rc)}\right) = \mathrm{tr}\left(\mathbf{\Sigma}_{\boldsymbol{b}(rc)}\right)\mathrm{tr}\left(\mathbf{W}\mathbf{\Phi}_{w(rc)}\right) \tag{B21}$$

The above re-expression gives the solution for the *sum* of the random slope variances, and multiplying this variation by the weights gives the diagonal elements of $\mathbf{\Sigma}_{\boldsymbol{b}(rc)}$.

$$\mathrm{tr}\left(\mathbf{\Sigma}_{\boldsymbol{b}(rc)}\right) = \frac{\mathrm{var}\left(Y_{ij}\right)R^2_{rc}}{\mathrm{tr}\left(\mathrm{diag}(\widetilde{\boldsymbol{\omega}}_{rc})\mathbf{\Phi}_{w(rc)}\right)} \tag{B22}$$

$$\mathrm{diag}\left(\mathbf{\Sigma}_{\boldsymbol{b}(rc)}\right) = \widetilde{\boldsymbol{\omega}}_{rc}\mathrm{tr}\left(\mathbf{\Sigma}_{\boldsymbol{b}(rc)}\right) \tag{B23}$$

Turning to the random intercept variance, the between-cluster variance of the random intercepts is obtained by subtracting out variation due to fixed effects from the total between-cluster variance, as follows.

$$\sigma^2_{b_0} = (ICC_Y)\mathrm{var}\left(Y_{ij}\right) - \boldsymbol{\beta}'_b \boldsymbol{\Phi}_b \boldsymbol{\beta}_b \tag{B24}$$

The solution for the random intercept variance changes if one of the level-1 predictors has a non-zero mean (e.g., a time score predictor in a growth model). In this case, the total between-cluster variation is

$$(ICC_Y)\mathrm{var}\left(Y_{ij}\right) = \boldsymbol{\beta}'_b \boldsymbol{\Phi}_b \boldsymbol{\beta}_b + \sigma^2_{b_0} + \boldsymbol{\mu}'_{rc} \boldsymbol{\Sigma}_{\boldsymbol{b}(rc)} \boldsymbol{\mu}_{rc} + 2\boldsymbol{\mu}'_{rc} \boldsymbol{\Sigma}_{\boldsymbol{b}(i,rc)} \tag{B25}$$

where $\boldsymbol{\mu}_{rc}$ is the mean vector of the random coefficient predictors, and $\boldsymbol{\Sigma}_{\boldsymbol{b}(i,rc)}$ is the vector of covariances between the random intercepts and slopes. More details about this expression are found in Snijders and Bosker (2012; Equations 7.9 and 7.10). To solve for $\sigma^2_{b_0}$, we define the following constants

$$v = (ICC_Y)\mathrm{var}\left(Y_{ij}\right) \tag{B26}$$

$$b = \boldsymbol{\beta}'_b \boldsymbol{\Phi}_b \boldsymbol{\beta}_b \tag{B27}$$

$$s = \boldsymbol{\mu}'_{rc} \boldsymbol{\Sigma}_{\boldsymbol{b}(rc)} \boldsymbol{\mu}_{rc} \tag{B28}$$

$$c\sqrt{\sigma^2_{b_0}} = 2\boldsymbol{\mu}'_{rc} \boldsymbol{\Sigma}_{\boldsymbol{b}(i,rc)} = 2\boldsymbol{\mu}'_{rc} \mathbf{R}_{i,rc} \circ \mathrm{diag}\left(\boldsymbol{\Sigma}_{\boldsymbol{b}(rc)}\right)^{\circ.5} \sqrt{\sigma^2_{b_0}} \tag{B29}$$

where $\mathbf{R}_{i,rc}$ is vector of correlations between the random intercepts and slopes, and $\circ$ denotes an elementwise operation (i.e., a Hadamard product and Hadamard power). The elementwise terms in Equation B24 are multiplying the correlation vector by the standard deviations of the random slopes. Finally, the solution for the random intercept variance is as follows.

$$v = b + \sigma^2_{b_0} + s + c\sqrt{\sigma^2_{b_0}} \tag{B30}$$

$$\sigma^2_{b_0} = .5\left(-2b + c^2 - 2s + 2v\right) - .5\sqrt{-4bc^2 + c^4 - 4c^2 s + 4c^2 v} \tag{B31}$$

The off-diagonal elements of $\boldsymbol{\Sigma_b}$ are a deterministic function of the intercept and slope variances and the correlations in $\mathbf{R}_{rc}$. For example, the covariance between the random intercepts and slopes from the illustrative model in Equation 1 is as follows.

$$\sigma^2_{b_1,b_0} = \sigma^2_{b_0,b_1} = R\left(b_{0j}, b_{1j}\right)\sqrt{\sigma^2_{b_0}\sigma^2_{b_1}} \tag{B32}$$

With more than one random slope, all correlations equal 0 because $\mathbf{R}_{rc}$ is an identity matrix.

Step 5: Within-Cluster Residual Variance

Having solved all other parameters, the within-cluster residual variance is obtained by subtraction, as follows.

$$\sigma^2_{\varepsilon} = \left(1 - ICC_Y - R^2_w - R^2_{rc}\right)\mathrm{var}\left(Y_{ij}\right) \tag{B33}$$

Covariances for Binary Level-2 Predictors

The R package accommodate binary level-2 predictors (e.g., a treatment assignment variable). To illustrate, consider a cluster-randomized design where schools are randomly assigned to an intervention and comparison condition. Further, assume that D_j^b is a grand mean centered binary variable (i.e., contrast code) with equal group proportions. Binary variables require a slight modification that attenuates between-cluster correlations among the predictors to account for categorization. Applying ideas from Allen (1997), our functions convert the input correlations to point-biserial correlations as follows

$$R_{\text{pb}}(D^b, W^b) = R(D^b, W^b) \Big/ \left(\sqrt{\text{var}(D^b)} \times f(E(D^b)) \right) \tag{34}$$

where $R(D^b, W^b)$ is the unadjusted correlation (i.e., the input correlation) between the binary variable and another level-2 predictor W^b, $R_{\text{pb}}(D^b, W^b)$ is the corresponding point-biserial correlation, and $f(E(D^b))$ is the height of the standard normal distribution at the z-score cutoff, above which the area under the curve equals $E(D^b) = \left(\Pr(D^b) = 1\right)$. After computing the point-biserial correlations, the R function uses the variances of each variable pair to compute covariances. As noted previously, the functions convert binary variables to contrast codes (i.e., dummy codes are centered) to induce a zero mean, as this feature is a vital to inducing the block diagonal predictor covariance matrices.

References

Aguinis, H., Beaty, J. C., Boik, R. J., & Pierce, C. A. (2005). Effect Size and Power in Assessing Moderating Effects of Categorical Variables Using Multiple Regression: A 30-Year Review. *Journal of Applied Psychology*, *90*(1), 94.

Allen, M. P. (1997). Covariance and Linear Independence. In *Understanding Regression Analysis* (pp. 31-35). Boston, MA: Springer US.

Arend, M. G., & Schäfer, T. (2019). Statistical Power in Two-Level Models: A Tutorial Based on Monte Carlo Simulation. *Psychological Methods*, *24*(1).

Baey, C., & Kuhn, E. (2022). Package 'Vartestnlme'. Retrieved from https://cran.r-project.org/web/packages/varTestnlme/varTestnlme.pdf

Bates, D., Maechler, M., Bolker, B., Walker, S., Christensen, R. H. B., Singmann, H., . . . Krivitsky, P. N. (2021). Package 'Lme4'. Retrieved from https://cran.r-project.org/web/packages/lme4/lme4.pdf

Bohrnstedt, G. W., & Goldberger, A. S. (1969). On the Exact Covariance of Products of Random Variables. *Journal of the American Statistical Association*, *64*, 1439–1442.

Bolger, N., & Laurenceau, J.-P. (2013). *Intensive Longitudinal Methods: An Introduction to Diary and Experience Sampling Research*. New York: Guilford Press.

Bosco, F. A., Aguinis, H., Singh, K., Field, J. G., & Pierce, C. A. (2015). Correlational Effect Size Benchmarks. *Journal of Applied Psychology*, *100*(2), 431.

Chaplin, W. F. (1991). The Next Generation in Moderation Research in Personality Psychology. *Journal of personality*, *59*, 143–178.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Hillsdale, NJ: Erlbaum.

Enders, C. K., Keller, B. T., & Woller, M. P. (2023). A Simple Monte Carlo Method for Estimating Power in Multilevel Designs. *Psychological Methods*.

Funder, D. C., & Ozer, D. J. (2019). Evaluating Effect Size in Psychological Research: Sense and Nonsense. *Advances in Methods and Practices in Psychological Science, 2*(2), 156-168.

Gignac, G. E., & Szodorai, E. T. (2016). Effect Size Guidelines for Individual Differences Researchers. *Personality and individual differences*, *102*, 74-78.

Hedges, L. V., & Hedberg, E. C. (2007). Intraclass Correlation Values for Planning Group-Randomized Trials in Education. *Educational Evaluation and Policy Analysis*, *29*(1), 60–87. doi:10.3102/0162373707299706

Hedges, L. V., & Hedberg, E. C. (2013). Intraclass Correlations and Covariate Outcome Correlations for Planning Two-and Three-Level Cluster-Randomized Experiments in Education. *Evaluation Review*, *37*(6), 445-489.

Martin, J. G., Nussey, D. H., Wilson, A. J., & Réale, D. (2011). Measuring Individual Differences in Reaction Norms in Field and Experimental Studies: A Power Analysis of Random Regression Models. *Methods in Ecology and Evolution*, *2*(4), 362-374.

Raudenbush, S. W. (1997). Statistical Analysis and Optimal Design for Cluster Randomized Trials. *Psychological Methods*, *2*(2), 173–185. doi:Doi 10.1037//1082-989x.2.2.173

Sellström, E., & Bremberg, S. (2006). Is There a "School Effect" on Pupil Outcomes? A Review of Multilevel Studies. *Journal of Epidemiology & Community Health*, *60*(2), 149-155.

Singer, J. D., & Willett, J. B. (2003). *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. New York: Oxford University Press.

Snijders, T. A., & Bosker, R. J. (1993). Standard Errors and Sample Sizes for Two-Level Research. *Journal of Educational Statistics, 18*(3), 237-259.

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel Analysis: An Introduction to Basic and Advanced Multilevel Modeling* (2nd ed.). Thousand Oaks, CA: Sage.

Spybrook, J., Bloom, H., Congdon, R., Hill, C., Martinez, A., & Raudenbush, S. W. (2011). Optimal Design Plus Empirical Evidence: Documentation for the "Optimal Design" Software Version 3.0.

Stockford, S. M. (2009). *Meta-Analysis of Intraclass Correlation Coefficients from Multilevel Models of Educational Achievement*: Arizona State University.