Technical Appendix: Details on the Power Calculations for Two- and Three-level Models with Missing Data

Kristoffer Magnusson
Department of Clinical Neuroscience
Karolinska Institutet

2018-08-12

This vignette details how the power calculations are implemented in **powerlmm**. We will focus on the fully *nested* three-level model, since the two- and partially nested three-level model are just reduced forms of the three-level model. Thus, in standard multilevel notation the fully nested three-level model is

$$
\begin{aligned}
&\text{Level 1} \\
&\quad Y_{ijk} = \beta_{0jk} + \beta_{1jk} t_{ijk} + R_{ijk} \\
&\text{Level 2} \\
&\quad \beta_{0jk} = \gamma_{00k} + U_{0jk} \\
&\quad \beta_{1jk} = \gamma_{10k} + U_{1jk} \\
&\text{Level 3} \\
&\quad \gamma_{00k} = \delta_{000} + \delta_{001} TX_k + V_{0k} \\
&\quad \gamma_{10k} = \delta_{100} + \delta_{101} TX_k + V_{1k}
\end{aligned}
$$

where we have $i = 1, \dots, n_{1j}$ equally spaced time points for subject $j = 1, \dots, N_2$, where N_2 is the total number of subjects in the treatment arm. Furthermore, the subjects are nested within $k = 1, \dots, n_3$ clusters, where n_3 is the total number of clusters in the treatment arm. To allow for varying cluster sizes we let each cluster have $j = 1, \dots, n_{2[k]}$ subjects, where $n_{2[k]}$ is the total number of subjects in cluster k.

The parameter of interest is δ_{101}, i.e. the mean difference in slopes between the two treatment groups. However, in **powerlmm** the calculations are simplified by calculating the variance of the slope-coefficient separately for each treatment group. Since the slopes in the treatment and control group are independent, the variance of the interaction-term is simply

$$\mathrm{V}(\delta_{101}) = \mathrm{V}(\delta_{100[tx]} - \delta_{100[c]}) = \mathrm{V}(\delta_{100[tx]}) + \mathrm{V}(\delta_{100[c]}),$$

where $\delta_{100[tx]}$ and $\delta_{100[c]}$ are the fixed time effects in the treatment and control group respectively. In order to calculate the variances we begin by formulating the three-level model for the complete data vector $\mathbf{Y}$ from a single treatment arm,

$$\mathbf{Y} = \mathbf{XZW}\boldsymbol{\beta} + \mathbf{Xu} + \mathbf{XZv} + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathbf{Y}$ is the $N_1 \times 1$ outcome vector containing all the observations from all the subjects in the treatment arm, $\mathbf{X}$ is a $N_1 \times 2N_2$ matrix containing co-variate information for all N_2 subjects in the treatment arm, $\mathbf{X}$ is also used as the design matrix for the second-level random effects. $\mathbf{Z}$ is a $2n_3 \times 2N_2$ matrix containing the level-three random effects design matrices for each kth cluster in the treatment arm. $\mathbf{W}$ is a $2n_3 \times 2$ matrix relating the third-level to the overall effects β, and here β is simply a 2×1 vector with the population values for the fixed intercept and slope effects. Lastly, $\mathbf{u}$ is a $2N_2 \times 1$ vector with the level two random effects, $\mathbf{v}$ is a $2n_3 \times 1$ vector with the third-level random effects, and $\boldsymbol{\epsilon}$ a $N_1 \times 1$ vector with the level one residuals.

The random effects and residuals are distributed as follows,

$$\begin{aligned}\mathbf{u} &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_2),\\ \mathbf{v} &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_3),\\ \boldsymbol{\epsilon} &\sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{N_1}).\end{aligned}$$

With the second and third level variance components being

$$\boldsymbol{\Psi}_2 = \mathbf{I}_{N_2} \otimes \begin{pmatrix} u_0^2 & u_{01} \\ u_{01} & u_1^2 \end{pmatrix}, \boldsymbol{\Psi}_3 = \mathbf{I}_{n_3} \otimes \begin{pmatrix} v_0^2 & v_{01} \\ v_{01} & v_1^2 \end{pmatrix},$$

with $\otimes$ denoting the Kronecker product. The co-variate matrix $\mathbf{X}$ is block-diagonal containing a sub-matrix $\mathbf{X}_{jk}$ for each subject (level-two unit), thus

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{X}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{X}_{N_2} \end{pmatrix}.$$

Since each subject can have a different number of observations due to dropout, each $\mathbf{X}_{jk}$ will have dimension $n_{1[j]} \times 2$, where $n_{1[j]}$ is the total number of observations for subject j in cluster k,

$$\mathbf{X}_{jk} = \begin{pmatrix} 1 & T_0 \\ 1 & T_1 \\ \vdots & \vdots \\ 1 & T_{n1[j]} \end{pmatrix}. \tag{2}$$

$\mathbf{Z}$ is a block-diagonal matrix containing the level-three design matrices for each cluster k,

$$\mathbf{Z} = \begin{pmatrix} \mathbf{Z}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{Z}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{Z}_{n3} \end{pmatrix}.$$

With the sub-matrices $\mathbf{Z}_k$ being stacks of 2×2 matrices for each subject in cluster k,

$$\mathbf{Z}_k = \mathbf{1}_{n2[k]} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

thus the dimension of $\mathbf{Z}_k$ will be $n_{2[k]} \times 2$, where $n_{2[k]}$ is the number of subjects in cluster k. This enables power calculations for designs with varying number of subjects per cluster.

The matrix $\mathbf{W}$, relates the cluster-level effects to the overall effects $\boldsymbol{\beta}$,

$$\mathbf{W} = \mathbf{1}_{n3} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and thus $\mathbf{XZW}$ simply stacks all the N_2 sub-matrices, $\mathbf{X}_{jk}$, into a $N_1 \times 2$ matrix.

Then we can calculate the marginal variance-covariance matrix for $\mathbf{Y}$ as

$$\mathrm{V}(\mathbf{Y}) = \mathbf{X}\boldsymbol{\Psi}_2\mathbf{X}^\top + \mathbf{XZ}\boldsymbol{\Psi}_3\mathbf{Z}^\top\mathbf{X}^\top + \boldsymbol{\epsilon}^2\mathbf{I}_{N1},$$

and the variance of the population parameters in $\boldsymbol{\beta}$ as

$$\mathrm{V}(\boldsymbol{\beta}) = [(\mathbf{XZW})^\top \mathrm{V}(\mathbf{Y})^{-1} \mathbf{XZW}]^{-1}. \tag{3}$$

The lower right corner of $\mathrm{V}(\boldsymbol{\beta})$ corresponds to the variance of the time-coefficient. As we noted earlier we can use the slope variances to calculate the variance of the time×treatment-interaction.

Accounting for dropout

Dropout is accounted for by defining a dropout vector $\mathbf{D} = (p_1, \ldots, p_{n_1})^\top$, where p_i is the proportion of participants that have dropped out at time point i, for the $i, \ldots, n_1$ scheduled time points, and $p_0 = 0$ and $p_i \leq p_{i+1}$. The default in **powerlmm** is to treat the values in $\mathbf{D}$ as known, i.e. exactly p_i subjects will have dropped out at time i. This is done by randomly sampling which $p_i N_2$ participants should drop out a time i, then adjusting their design matrices $\mathbf{X}_{ij}$ to be of size $(i-1) \times 2$, thus their last time point will be $i-1$. Since, it is random which subjects will dropout, the power calculations will differ slightly each time. It is also possible to treat $\mathbf{D}$ as random (using the option `deterministic_dropout = FALSE`), then dropout will be sampled from a multinomial distribution, by converting the elements of $\mathbf{D}$ to the probability p_i that a subject will have exactly i measurements. This approach is similar to Galbraith, Stat, and Marschner (2002), and Verbeke and Lesaffre (1999) who presents power calculations for two-level models with missing data.

Speeding up the computation of $V(Y)^{-1}$

Doing the matrix inversion of $V(\mathbf{Y})$, which is of dimension $N_1 \times N_1$, can be extremely slow for some designs. De Leeuw and Kreft (1986) (where they credit Swamy (1971)) noted a more computationally efficient formulation, adopting it to the three-level formulation in Equation 1, lets us write

$$V(\mathbf{Y})^{-1} = \sigma^{-2}[\mathbf{I}_{N_1} - \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top] + \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{A}^{-1}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X},$$

where,

$$\mathbf{A}^{-1} = [\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1} + \boldsymbol{\Psi}_2 + \mathbf{Z}\boldsymbol{\Psi}_3\mathbf{Z}^\top].$$

Here $\mathbf{A}$ of size $2N_2 \times 2N_2$. However, since $\mathbf{A}$ is block-diagonal, with each block for cluster k being of size $n_{2[k]}$, the computation done in **powerlmm**, takes advantage of the sparse matrix functions from the `Matrix`-package. By using sparse matrix algebra the speed of computing $\mathbf{A}^{-1}$ will depend greatly on the number of subjects per cluster. In most cases this solution is dramatically faster then directly solving $V(\mathbf{Y})^{-1}$. For instance, calculating $\mathrm{V}(\beta)$ for a study with $n_1 = 10$, $n_2 = 30$, $n_3 = 20$ is approximately 50 times faster using this method.

Changes in powerlmm 0.2

As of version 0.2, $\mathrm{V}(\boldsymbol{\beta})$ is now computed using the sparse Cholesky factorization used in **lme4**, and the implementation specifically borrows from **lme4pureR**. Thus,

$$\mathrm{V}(\boldsymbol{\beta}) = \sigma^2 \mathbf{R}_X^{-1}(\mathbf{R}_X^\top)^{-1}$$

where $\mathbf{R}_X$ is the Cholesky factor of the fixed effects, see Eq. 54 in Bates et al. (2015).

Power

To make the power calculations accurate for small samples sizes, power is calculated using the t distribution. Thus, we can define the power function as,

$$1 - \beta = P(t_{\nu,\lambda} > t_{\nu,1-\alpha/2}) + P(t_{\nu,\lambda} < t_{\nu,\alpha/2}),$$

where λ is the non-centrality parameter,

$$\lambda = \delta_{101}/\sqrt{\mathrm{V}(\delta_{101})},$$

and ν is the appropriate degrees of freedom of the t distribution. For the balanced fully nested three-level model, the degrees of freedom are $N3 - 2$, where $N3$ is the total number of clusters in both treatment arms.

Satterthwaite's degrees of freedom approximation

For small samples, the choice of degrees of freedom will potentially influence the accuracy of the power analysis a lot. In **powerlmm** it is therefore possible to use Satterthwaite's DF approximation in the power analysis. The degrees of freedom of the t distribution is approximated as,

$$\nu = \frac{2(\mathbf{L}^\top \mathrm{V}(\boldsymbol{\beta})\mathbf{L})^2}{V(\mathbf{L}^\top \mathrm{V}(\boldsymbol{\beta})\mathbf{L})},$$

and $\mathbf{L}$ specifies the linear contrast we are testing. Moreover, $V(\mathbf{L}^\top \mathrm{V}(\boldsymbol{\beta})\mathbf{L})$ is approximated using the delta method

$$V(\mathbf{L}^\top \mathrm{V}(\boldsymbol{\beta})\mathbf{L}) \cong [\Delta_{f(\boldsymbol{\theta})}(\boldsymbol{\theta})]^\top \mathrm{V}(\boldsymbol{\theta})[\Delta_{f(\boldsymbol{\theta})}(\boldsymbol{\theta})],$$

$\Delta_{f(\boldsymbol{\theta})}(\boldsymbol{\theta})$ is the gradient of $\mathbf{L}V(\boldsymbol{\beta})\mathbf{L}$ with respect to $\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is the vector of variance components, and thus $\mathrm{V}(\boldsymbol{\theta})$ is the asymptotic covariance matrix of the random effects (including σ^2), which is approximated as $\mathrm{V}(\boldsymbol{\theta}) = 2\boldsymbol{\mathcal{I}}_E^{-1}$, where $\boldsymbol{\mathcal{I}}_E$ is the expected information matrix for the variance parameters. The calculation of $\boldsymbol{\mathcal{I}}_E$ is described in Equation 25 in Halekoh and Højsgaard (2014). However, this implementation involves manipulating $V(\mathbf{Y})$, i.e. the full variance-covariance matrix including all N observations. For large sample sizes this will be very computationally intensive, and the computation time will depend mostly on n_1 and n_2. For instance, for a fully nested model with $n_1 = 10$, $n_2 = 100$, $n_3 = 4$, computations will likely take 30-60 seconds, and be very RAM intensive.

Partially nested designs

For the partially nested designs $\mathrm{V}(\delta_{100[tx]})$ is calculated as above, and $\mathrm{V}(\delta_{100[c]})$ by setting the cluster-level random effects to zero. Degrees of freedom for this model is trickier, and I *recommend always* using Satterthwaite DFs whenever possible. If balanced DFs are requested, then currently $n_3 - 1$ i used, where n_3 is the number of clusters in the treatment group only.

Two-level designs

For the two-level designs, $\mathrm{V}(\delta_{101})$ can be calculated using the three-level formulas with the cluster-level random effects set to zero. Deleting these terms reduces the model to the classical two-level formulation. Degrees of freedom for the balanced model is $N_2 - 2$, where N_2 is the total number of subjects in both treatment arms.

Standardized formulation

If there's no missing data and the clusters sizes are balanced, the variance of the slope can be calculated more simply as

$$\mathrm{V}(\delta_{100}) = \frac{\sigma^2 + n_1\sigma^2_{u_1}V(\mathbf{t}) + n_1n_2\sigma^2_{v_1}V(\mathbf{t})}{n_1n_2n_3V(\mathbf{t})},$$

with,

$$\mathrm{V}(\mathbf{t}) = \Sigma_{i=1}^{n_1}(t_i - \bar{t})^2.$$

By defining the amount of slope variance at the cluster-level as $\rho_s = \sigma^2_{v_1}/(\sigma^2_{v_1} + \sigma^2_{u_1})$, and `ICC_pre_subjects` $= \rho_1 = (\sigma^2_{u_0} + \sigma^2_{v_0})/(\sigma^2_{v_0} + \sigma^2_{u_0} + \sigma^2_e)$, and the variance ratio as $r_\tau = (\sigma^2_{v_1} + \sigma^2_{u_1})/\sigma^2_e$ we can then rewrite the formula using the relative parameters ρ_1, ρ_s and r_τ,

$$\mathrm{V}(\delta^*_1) = \frac{(1-\rho_1) + n_1\mathrm{Var}(\mathbf{t})(1-\rho_1)[n_2\rho_s r + (1-\rho_s)r]}{n_1n_2n_3\mathrm{Var}(\mathbf{t})},$$

which will yield the same non-centrality parameters as long as the interaction-coefficient corresponds to the same standardized value, e.g. Cohen's d. Thus, we see that power depends on $n1$, $n2$, $n3$, the duration of the study, the proportion of intercept variance at baseline, the amount of slope variance at the third level, and the variance ratio.

References

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software* 67 (1): 1–48. doi:10.18637/jss.v067.i01.

De Leeuw, J, and I Kreft. 1986. "Random coefficient models for multilevel analysis." *Journal of Educational Statistics* 11 (1): 57.

Galbraith, S, M Stat, and I C Marschner. 2002. "Guidelines for the design of clinical trials with longitudinal outcomes." *Controlled Clinical Trials* 23 (3): 257–73.

Halekoh, Ulrich, and Søren Højsgaard. 2014. "A Kenward-Roger Approximation and Parametric Bootstrap Methods for Tests in Linear Mixed Models–the R Package Pbkrtest." *Journal of Statistical Software* 59 (9): 1–32.

Swamy, Paravastu Ananta Venkata Bhattanadha. 1971. *Statistical Inference in Random Coefficient Regression Models*. New York: Springer Verlag.

Verbeke, Geert, and Emmanuel Lesaffre. 1999. "The Effect of Drop-Out on the Efficiency of Longitudinal Experiments." *Journal of the Royal Statistical Society* 48 (3): 363–75.