

Understanding meta-analysis through data simulation with application to power analysis

Supplementary materials

Filippo Gambarota

Gianmarco Altoè

Contents

1	Introduction	2
2	Simulations	2
3	Three-level model	4
4	Multivariate model	7
5	Extra	14
6	Useful resources	16
7	References	16

1 Introduction

In the main paper, we presented the most straightforward situation for a meta-analysis when each included study contributes with a single effect size. However, in real-world meta-analyses, it is common to find complex data structures such as multilevel and multivariate. There are numerous sources of dependency for a meta-analysis (Cheung, 2014; 2015, pp. 121–122).

The first type of dependency (*multilevel*) consists of multiple effect sizes from independent participants. Despite being collected with different participants, effect sizes from the same study could be correlated. We obtain a multilevel data structure with independent effect sizes in the same study. Figure S1 depict the general idea of a *multivariate* model. There are multiple ways to model this data structure (see Cheung, 2014 for an overview). Still, the most common way is to use a *multilevel meta-analysis model*, considering the heterogeneity between and within studies. The standard *random-effects* (but also the *equal-effects model*) model is also called a *two-level* model because it combines data after aggregating the participants-level (*first-level*) data. The *three-level* model is commonly used to manage the multilevel data structure (Cheung, 2014, 2019). Compared to the *two-level* model, the *three-level* model estimates two heterogeneity parameters (τ^2 and ω^2), one for the heterogeneity between studies and the other for the heterogeneity within studies.

A second type of dependency (*multivariate*) consists of multiple effect sizes collected from the same pool of participants. Figure S2 depict the general idea of a *multivariate* model. For example, when using two cognitive tasks on the same pool of participants. This is commonly known as *multivariate meta-analysis* because the effect sizes sampling errors are correlated given that participants are assessed multiple times. A *multivariate meta-analysis model* could be used to consider the correlation between sampling errors.

A third type of dependency arises when true effect sizes are correlated at the population level (Cheung, 2015, pp. 121–122). For example, two different memory tasks could be correlated because the latent psychological constructs are intrinsically correlated. This correlation is present even when other groups of participants perform the two memory tasks (i.e., no *multivariate* dependency) or are administered by different researchers (i.e., no *multilevel* dependency). This type of dependency could be managed by estimating the correlation between outcomes using a *multivariate random-effects model* (Cheung, 2015, pp. 127–130). Figures S1 and S2 depict the conceptual idea of *multilevel* and *multivariate* models. The following sections will show two examples to simulate these complex data structures.

2 Simulations

```
library(dplyr)
library(tidyr)
seed <- 2023 # random seed for simulations
```

In the paper, we used the `mapply()` function, a particularly useful tool in R, to apply the same function (in parallel) with multiple parameters using multiple vectors. In our case, the `sim_study()` function has several parameters (`es`, `nt`, among others), and we can apply the function multiple times with a combination of parameters. It can be considered essentially a `for` loop but more compact and less verbose.

The following code illustrates the idea of `mapply()` compared to a standard `for` loop. The `sim` data set contains our pool of k studies, and we want to apply the `sim_study()` function to each dataset row. We can use a `for` loop that is verbose but more explicit or the `mapply()` function that is more compact but less explicit. The computation time and results `for` the two approaches are equivalent, given the appropriate setup.

```
# using a for loop

# preallocate for speed (not necessary with mapply)
```

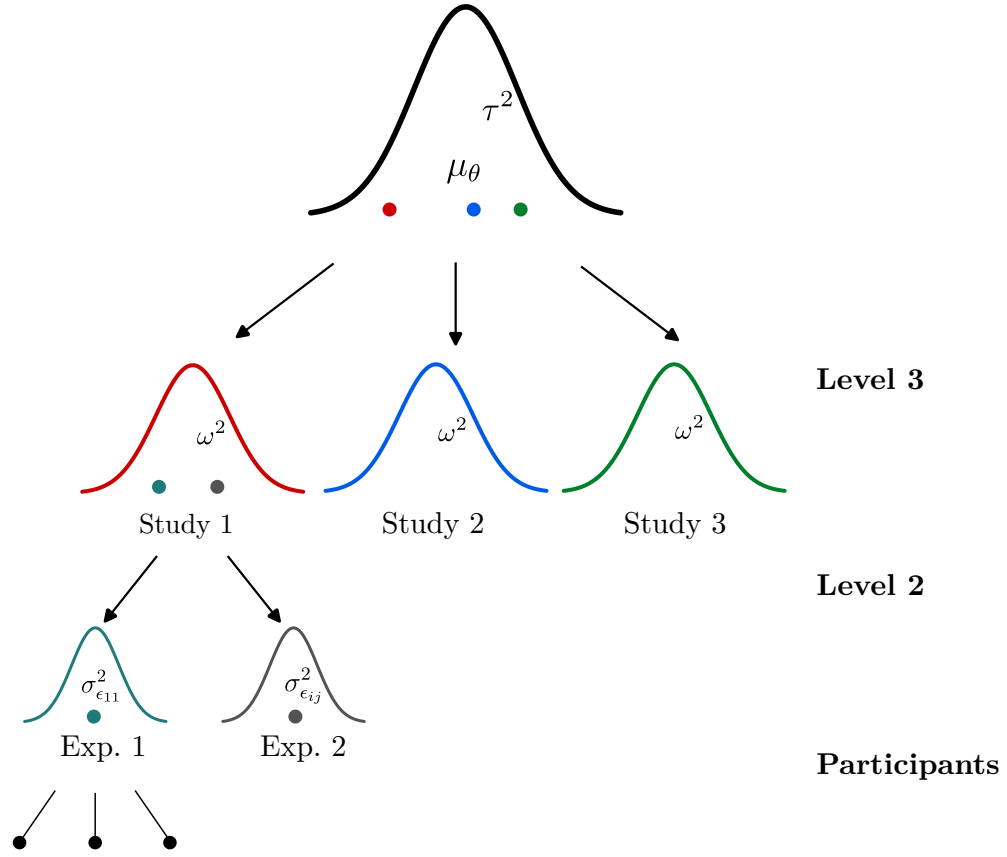


Figure S1: Graphical representation of a *multilevel* (three-level) model. Each observed effect size is defined as the overall effect average effect μ_θ , a study-specific random-effect $\delta_i \sim N(0, \tau^2)$ and a experiment-specific random-effects $\zeta_{ij} \sim N(0, \omega^2)$.

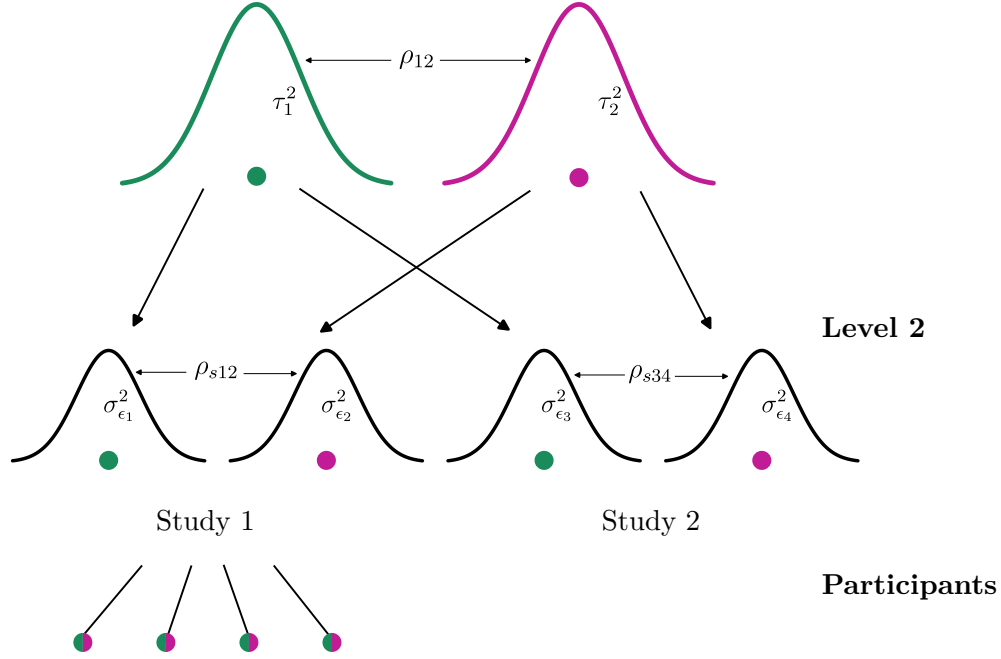


Figure S2: Graphical representation of a *multivariate* model. In this example, we have two outcomes (p) (magenta and green) with an average effect (μ_{θ_p}), heterogeneity parameters (τ_p^2), and a certain true correlation value ($\rho_{p_1 p_2}$, third type of dependency). Each study at level-2 (as in the standard random-effects model) has two effect sizes with corresponding sampling variances. The two effect sizes are correlated (ρ_s , with s for sampling errors) because each effect is collected on the same participants.

```
res <- vector(mode = "list", length = nrow(sim))
for(i in 1:length(res)){
  res[[i]] <- sim_study(theta = sim$es[i], nc = sim$nc[i], nt = sim$nt[i])
}

# using mapply
sim <- mapply(sim_study, sim$es, sim$nc, sim$nt)

# using sim_studies (with mapply inside)
sim <- sim_studies(sim$es, sim$nc, sim$nt, data = sim)
```

3 Three-level model

Compared to the standard two-level model, the three-level model estimates another source of heterogeneity (ω^2) representing the variability within studies. We can easily extend the two-level *random-effects* equation into equation (??) from the main paper adding another “adjustment” to the overall effect.

$$y_{ij} = \mu_{\theta} + \delta_i + \zeta_{ij} + \epsilon_{ij} \quad (\text{S1})$$

$$\delta_i \sim N(0, \tau^2) \quad (\text{S2})$$

$$\zeta_{ij} \sim N(0, \omega^2) \quad (\text{S3})$$

$$\epsilon_{ij} \sim N(0, \sigma_{\epsilon_i}^2) \quad (\text{S4})$$

The notation suggests that each effect size belongs to an experiment (effect size) (j) nested within a study (i). The variability between studies (level 3) is handled by the τ^2 parameter (see Equation (S2)) and the variability within studies (level 2) is handled by the ω^2 parameter (see Equation (S3)). Practically, each study is composed of the average effect size μ_θ plus a study-specific adjustment determined by τ^2 . Then each effect size is composed of the study-specific effect ($\mu_\theta + \delta_i$) plus the adjustment determined by ω^2 thus $\mu_\theta + \delta_i + \zeta_{ij}$. As in the two-level meta-analysis model there is also the random error component $\sigma_{\epsilon_i}^2$ (see Equation (S4)). The number of effect sizes within each study will determine the estimation precision of the ω^2 parameter and the number of studies will determine the estimation precision of the τ^2 parameter. The relationship between τ^2 and ω^2 can be expressed using the intraclass correlation coefficient (ICC) expressed as $ICC = \frac{\tau^2}{\tau^2 + \omega^2}$ representing the proportion of total heterogeneity associated with level 2 or level 3 (see Cheung, 2014). For example, an ICC of 0.5 suggests that 50% of heterogeneity is caused by between-studies variability. On the extreme, an ICC close to 1 suggests that the between-studies variability causes most heterogeneity (i.e., the effect sizes within each study are very similar).

```
set.seed(seed)
i <- 50 # Number of studies (level 3)
j <- 5 # Number of effect sizes within each study (level 2)
tau2 <- 0.3 # heterogeneity between studies
omega2 <- 0.1 # heterogeneity within studies
icc <- tau2 / (tau2 + omega2) # real ICC
n <- 30 # sample size for each group, within each study
mu <- 0.3 # real average effect size

delta_i <- rnorm(i, 0, sqrt(tau2))
zeta_ij <- rnorm(i*j, 0, sqrt(omega2))

sim <- tidyr::expand_grid(
  study = 1:i,
  effect = 1:j,
  nt = n,
  nc = n,
  mu = mu
)

sim$delta_i <- delta_i[sim$study]
sim$zeta_ij <- zeta_ij

head(sim)
```

```
## # A tibble: 6 x 7
##   study effect   nt   nc   mu delta_i zeta_ij
##   <int>   <int> <dbl> <dbl> <dbl>   <dbl>   <dbl>
## 1     1     1    30    30  0.3 -0.0459  0.210
## 2     1     2    30    30  0.3 -0.0459 -0.345
## 3     1     3    30    30  0.3 -0.0459 -0.134
```

```
## 4      1      4      30      30      0.3 -0.0459      0.374
## 5      1      5      30      30      0.3 -0.0459      0.501
## 6      2      1      30      30      0.3 -0.538      0.721
```

Each study i is repeated j times¹ along the δ_i while each effect has a unique ζ_{ij} . We can use the same approach as the main paper using the `sim_studies()` function.

```
set.seed(seed)
sim$es <- sim$mu + sim$delta_i + sim$zeta_ij
sim <- sim_studies(es = sim$es, nt = sim$nt, nc = sim$nc, data = sim)
```

Finally we fit a *three-level models* using the `rma.mv` from the `metafor` package. The syntax differs slightly from the `rma` function and is clearly explained here <https://www.metafor-project.org/doku.php/analyses:konstantopoulos2011>. The essential part is adding the nested random effect using the `random =` argument specifying that the `study` variable is nested within the `study` variable with `~ 1|study/experiment`.

```
res <- rma.mv(yi, vi, random = ~1|study/effect, data = sim)
summary(res)
```

```
##
## Multivariate Meta-Analysis Model (k = 250; method: REML)
##
##      logLik    Deviance      AIC      BIC      AICc
## -169.0058    338.0116    344.0116    354.5640    344.1096
##
## Variance Components:
##
##      estim      sqrt  nlvls  fixed      factor
## sigma^2.1  0.3106  0.5573    50     no      study
## sigma^2.2  0.0749  0.2737   250     no  study/effect
##
## Test for Heterogeneity:
## Q(df = 249) = 1770.2214, p-val < .0001
##
## Model Results:
##
## estimate      se      zval      pval      ci.lb      ci.ub
##   0.4098   0.0823   4.9800   <.0001   0.2485   0.5711   ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As explained above, the model estimates the average effect μ_θ and the two heterogeneity parameters. To create a more realistic simulation, we can create some variability in the number of experiments within the same paper, similar to what we did for the number of participants in the main paper. We simulated a three-level model, but it is possible to extend to n levels for a more complicated nested structure. In this case, we need more heterogeneity parameters (i.e., more nested “adjustments”), but the overall simulation setup is the same.

¹To create a more realistic simulation, the number of effect sizes for each paper could be heterogeneous (e.g., sampled from a Poisson distribution)

4 Multivariate model

Compared to a standard two-level random-effects model, the *multivariate model* must consider the correlation between effect sizes collected on the same pool of participants. Authors should report these correlations, but often we need to guess a plausible value for the meta-analysis. Compared to the *two-level* and *three-level* models, we now have multiple outcomes (p).

Following the example of the main paper, we have three memory tests collected on the same pool of participants. For this reason we have three real effects (μ_{θ_1} , μ_{θ_2} and μ_{θ_3}), three heterogeneity (τ_1^2 , τ_2^2 and τ_3^2) and the population-level correlations between these outcomes (ρ_{12} , ρ_{13} and ρ_{23}). Moreover, the sampling errors (ϵ_i) are now correlated; thus each study will have three sampling variances for each outcome ($\sigma_{\epsilon_1}^2$, $\sigma_{\epsilon_2}^2$, and $\sigma_{\epsilon_3}^2$), and the correlations between sampling errors (ρ_{s12} , ρ_{s13} and ρ_{s23})². Equations (S5), (S6), (S7), (S8) (S9) formalize the multivariate model for a single study i with three outcomes. The random-effects adjustments (θ_i) to the overall effect and sampling errors are now sampled from a multivariate normal distribution, respectively $\theta_i \sim \mathcal{MVN}(0, T^2)$ and $\theta_i \sim \mathcal{MVN}(0, V^2)$.

$$\begin{bmatrix} y_{1_i} \\ y_{2_i} \\ y_{3_i} \end{bmatrix} = \begin{bmatrix} \mu_{\theta_1} \\ \mu_{\theta_2} \\ \mu_{\theta_3} \end{bmatrix} + \begin{bmatrix} \delta_{1_i} \\ \delta_{2_i} \\ \delta_{3_i} \end{bmatrix} + \begin{bmatrix} \epsilon_{1_i} \\ \epsilon_{2_i} \\ \epsilon_{3_i} \end{bmatrix} \quad (\text{S5})$$

$$\begin{bmatrix} \delta_{1_i} \\ \delta_{2_i} \\ \delta_{3_i} \end{bmatrix} \sim \mathcal{MVN}(0, T) \quad (\text{S6})$$

$$\begin{bmatrix} \epsilon_{1_i} \\ \epsilon_{2_i} \\ \epsilon_{3_i} \end{bmatrix} \sim \mathcal{MVN}(0, V) \quad (\text{S7})$$

$$T = \begin{bmatrix} \tau_1^2 & & \\ \rho_{21}\tau_2\tau_1 & \tau_2^2 & \\ \rho_{31}\tau_3\tau_1 & \rho_{32}\tau_3\tau_2 & \tau_3^2 \end{bmatrix} \quad (\text{S8})$$

$$V = \begin{bmatrix} \sigma_{\epsilon_1}^2 & & \\ \rho_{s21}\sigma_{\epsilon_2}\sigma_{\epsilon_1} & \sigma_{\epsilon_2}^2 & \\ \rho_{s31}\sigma_{\epsilon_3}\sigma_{\epsilon_1} & \rho_{s32}\sigma_{\epsilon_3}\sigma_{\epsilon_2} & \sigma_{\epsilon_3}^2 \end{bmatrix} \quad (\text{S9})$$

We can use the `MASS::mvrnorm()` to generate data from a multivariate normal distribution in R. Compared to `rnorm()`, `MASS::mvrnorm()` requires the vector of means (in this case zero) and the variance-covariance matrix.

```
set.seed(seed)
p <- 3 # number of outcomes
taus2 <- c(0.3, 0.1, 0.2) # vector of taus
rho <- 0.6 # correlation between outcomes
Tcmat <- rho + diag(1 - rho, nrow = p) # correlation matrix
Tcmat
```

²We used the subscript s for sampling errors to distinguish between between-outcomes correlations and correlations between sampling errors

```
##      [,1] [,2] [,3]
## [1,]  1.0  0.6  0.6
## [2,]  0.6  1.0  0.6
## [3,]  0.6  0.6  1.0

# correlation matrix to variance-covariance matrix
Tvcov <- diag(sqrt(taus2)) %*% Tcmat %*% diag(sqrt(taus2))
round(Tvcov, 3)
```

```
##      [,1] [,2] [,3]
## [1,] 0.300 0.104 0.147
## [2,] 0.104 0.100 0.085
## [3,] 0.147 0.085 0.200
```

```
multi_sim <- MASS::mvrnorm(n = 100, mu = rep(0, p), Sigma = Tvcov)
head(multi_sim)
```

```
##      [,1]      [,2]      [,3]
## [1,] -0.02016864  0.410185486 -0.13194700
## [2,]  0.54175504 -0.069826019  0.50505458
## [3,]  1.14644044  0.336528536  0.50838866
## [4,]  0.38035421 -0.482370414  0.01473713
## [5,]  0.52888869 -0.004733355  0.05339601
## [6,] -0.52921867 -0.413480727 -0.34537516
```

```
# check the result, close to the simulated values
apply(multi_sim, 2, mean) # mean
```

```
## [1] -0.0003956354  0.0066828825 -0.0863630452
```

```
apply(multi_sim, 2, var) # variance
```

```
## [1] 0.29557227 0.09757498 0.18773351
```

```
cor(multi_sim) # correlation
```

```
##      [,1]      [,2]      [,3]
## [1,] 1.0000000 0.6002067 0.6356537
## [2,] 0.6002067 1.0000000 0.6074336
## [3,] 0.6356537 0.6074336 1.0000000
```

The previous code simulated a *compound symmetry* (CS) structure where all variances (τ^2) and covariances ($\rho\tau_1\tau_2$) are the same. There could be different structures of the T matrix that are clearly explained in the `metafor::rma.mv()` documentation (see <https://www.io/metafor/reference/rma.mv.html#specifying-random-effects>). For the sake of simplicity we will use the CS structure. We can use the `sim_T()` function that given the number of studies k , outcomes p , the vector of τ^2 and the correlation ρ generates the random-effect adjustments³.

³The function allow to specify heterogeneous τ^2 values where the CS structure assume the same τ^2 for each outcome. Using a fixed ρ and different τ^2 will create what `rma.mv()` calls a *heteroscedastic compound symmetric structure* thus a matrix where the correlation is the same between outcomes and each outcome has a different τ^2 value


```
sim_T <- function(k, p, taus2, rho){
  rhoM <- rho + diag(1 - rho, nrow = p)
  tau2M <- diag(sqrt(taus2)) %*% rhoM %*% diag(sqrt(taus2))
  c(t(MASS::mvrnorm(k, rep(0, p), tau2M)))
}
```

```
set.seed(seed)
# compound symmetry
sim_T(k = 10, p = 3, taus2 = c(0.2, 0.2, 0.2), rho = 0.7)
```

```
## [1] -0.137783060  0.006924584  0.030317250 -0.400980223
## [5] -0.301731906 -0.476820366 -0.599321253 -1.017707804
## [9] -0.633051729 -0.101130503 -0.181352426  0.059109337
## [13] -0.267400421 -0.117154089 -0.375628328  0.047702666
## [17]  0.692241336  0.569012955 -0.534720462 -0.313042059
## [21] -0.248710208  0.206484487  0.507666832  0.487816335
## [25]  0.007204637 -0.511638473  0.025313912  0.222197914
## [29] -0.721417315 -0.062528264
```

```
# heteroscedastic compound symmetry
sim_T(k = 10, p = 3, taus2 = c(0.1, 0.3, 0.2), rho = 0.7)
```

```
## [1]  0.20547847  0.04380675  0.14946731 -0.50545878
## [5] -0.69452948 -0.34453427  0.15527888  0.35994483
## [9]  0.43542728  0.21001803  0.01567005 -0.10341940
## [13]  0.45596919  0.22106059  0.20538547  0.54161212
## [17]  0.02342131  0.15806043 -0.39421052 -0.59991781
## [21]  0.41745313 -0.32040433 -0.75720059 -0.20041835
## [25] -0.18732400 -0.57476731 -0.33691260  0.18553571
## [29] -0.09171771  0.15354564
```

For the sampling errors, the matrix V^2 for the study, i is essentially a $p_i \times p_i$ matrix, where the diagonal contains the sampling variances, and off-diagonal elements are the covariances. Crucially, sampling errors are correlated within a study but uncorrelated between studies. For this reason, stacking all matrices from k studies will create a block variance-covariance matrix where the correlations between sampling errors of different studies are fixed to zero. Figure S3 depicts the idea of the block-variance covariance matrix for three studies with a different number of outcomes. The block variance-covariance matrix can be easily created using the `metafor::vcalc()` function.

We have all pieces to simulate the multivariate model. We can use the `sim_study_m()` that follows the same idea of the `sim_study()` function but extends to multiple outcomes. We generate a single study with p outcomes where sampling errors are correlated according to a compound symmetry structure⁴. In practical terms, we assume that the correlation at the participant level between multiple outcomes is r and fixed between studies.

```
sim_study_m <- function(mus, nt, nc, r){
  p <- length(mus)

  # variance covariance matrix with sigma2 = 1
```

⁴Also for the V matrix we can assume different variance-covariance structures. In this example we are using a CS structure

$$V = \begin{bmatrix} \text{Study 1} & & \\ \sigma_{\epsilon_{1p_1}}^2 & \sigma_{\epsilon_{1p_2}}^2 & \sigma_{\epsilon_{1p_3}}^2 & & & \\ \sigma_{\epsilon_{1p_2}}^2 & \sigma_{\epsilon_{1p_1}}^2 & \rho_{s_{p_2p_1}} & \sigma_{\epsilon_{1p_2}}^2 & & \\ \sigma_{\epsilon_{1p_3}}^2 & \sigma_{\epsilon_{1p_1}}^2 & \rho_{s_{p_3p_2}} & \sigma_{\epsilon_{1p_2}}^2 & \rho_{s_{p_3p_2}} & \sigma_{\epsilon_{1p_3}}^2 \\ 0 & 0 & 0 & 0 & \text{Study 2} & \\ & & & & \sigma_{\epsilon_{2p_1}}^2 & \\ 0 & 0 & 0 & 0 & 0 & \text{Study 3} \\ & & & & & \sigma_{\epsilon_{3p_1}}^2 \\ & & & & & \sigma_{\epsilon_{3p_2}}^2 & \sigma_{\epsilon_{3p_1}}^2 & \rho_{s_{p_2p_1}} & \sigma_{\epsilon_{3p_2}}^2 \end{bmatrix}$$

Figure S3: Example of a block variance-covariance matrix for three hypothetical studies. In the example, there are three outcomes ($p = 3$), but not all studies have all outcomes. For this reason, the matrix represents each study as a “block,” and covariances between different studies are fixed to zero (because sampling errors are not correlated for different participants). The first study (pink) has three outcomes, thus a 3×3 matrix where the diagonal contains the sampling variances and off-diagonal elements are the sampling covariances. The second study has only one outcome; thus, the matrix is a single value for the sampling variance. The last study follows the same logic as the first study, with only two outcomes.

```
Vcm <- r + diag(1 - r, nrow = p)

yc <- MASS::mvrnorm(nt[1], rep(0, p), Vcm)
yt <- MASS::mvrnorm(nc[1], mus, Vcm)

ytm <- apply(yt, 2, mean)
ycm <- apply(yc, 2, mean)

ytv <- apply(yt, 2, var)
ycv <- apply(yc, 2, var)

yi <- ytm - ycm
vi <- ytv/nt[1] + ycv/nc[1]

data.frame(yi = yi, vi = vi)
}

set.seed(seed)
mus <- c(0.1, 0, 0.7) # three real average outcomes
n <- 30 # number of participant per study
r <- 0.6 # correlation between sampling errors
sim_study_m(mus, nt = n, nc = n, r = r)

##          yi          vi
## 1  0.19664472 0.08077794
## 2 -0.03510052 0.07668513
## 3  0.43822286 0.05596794
```

As in the main paper, we can create a dataframe with simulation parameters and iterating the `sim_study_m()` producing a meta-analysis dataframe.

```

set.seed(seed)
k <- 100 # number of studies
p <- 3 # number of outcomes per study
rho <- 0.7 # population level correlation between outcomes
r <- 0.5 # correlation between sampling errors
n <- 30 # number of participants per group per study

mus <- c(0.1, 0, 1) # population level real effect sizes
taus2 <- c(0.3, 0.3, 0.3) # population level real tau2s

# creating the dataframe structure
sim <- expand_grid(
  id = 1:k,
  p = 1:p,
  nt = n,
  nc = n
)

head(sim)

```

```

## # A tibble: 6 x 4
##   id     p   nt   nc
##   <int> <int> <dbl> <dbl>
## 1     1     1    30    30
## 2     1     2    30    30
## 3     1     3    30    30
## 4     2     1    30    30
## 5     2     2    30    30
## 6     2     3    30    30

```

```

# adding the real effects and the random-effect adjustments
sim$mu <- rep(mus, k)
sim$delta_i <- sim_T(k, p, taus2, rho)

# average effect + random-effect adjustment
sim$es <- sim$mu + sim$delta_i

# splitting to list of studies for improving the simulation speed
siml <- split(sim, sim$id)

# simulating the effect sizes for each study
resl <- lapply(siml, function(x) sim_study_m(x$es, x$nt, x$nc, r))
res <- dplyr::bind_rows(resl) # combine everything
sim <- cbind(sim, res) # append to the sim dataframe

head(sim)

```

```

##   id p nt nc mu   delta_i      es      yi
## 1  1 1 30 30 0.1 -0.4351134 -0.3351134 -0.07086765
## 2  1 2 30 30 0.0  0.2623641  0.2623641  0.44818811
## 3  1 3 30 30 1.0  0.2958867  1.2958867  1.51695135
## 4  2 1 30 30 0.1  0.8548680  0.9548680  0.91525720
## 5  2 2 30 30 0.0  0.3180528  0.3180528  0.32884179

```

```
## 6  2 3 30 30 1.0  0.2717055  1.2717055  1.45530564
##          vi
## 1 0.07212637
## 2 0.06335647
## 3 0.06970182
## 4 0.07463058
## 5 0.06451803
## 6 0.07696529
```

Now, we can use the `metafor::rma.mv()` function to fit the multivariate random-effects model. Compared to the multilevel model, the `metafor::rma.mv()` for a multivariate model uses the block variance-covariance matrix as the `vi` argument and the variable representing each outcome as a moderator estimating the average effect size. We can create the block variance-covariance matrix. Given that we simulated the raw data, we can calculate and use the observed correlations (within the `sim_study_m()` function) and use a different correlation for each study within the `vcalc()` function (see <https://www.metafor-project.org/doku.php/analyses:berkey1998> for an example). Often, these values are not reported; thus, we need to guess a plausible correlation to compute the block variance-covariance matrix. Here we use the r value from our simulation and use the same for each study. We know the underlying model in this case, so our guessed value is correct.

```
# create the block variance-covariance matrix
# cluster is the study (i.e. each "block")
# obs identify each outcome (i.e., each "block" number of rows/columns)
# r is the fixed sampling errors correlation
V <- vcalc(vi = vi, cluster = id, obs = p, rho = r, data = sim)

# first two studies (notice the off diagonal zeros for the correlation between different studies)
round(V, 3)[1:6, 1:6]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 0.072 0.034 0.035 0.000 0.000 0.000
## [2,] 0.034 0.063 0.033 0.000 0.000 0.000
## [3,] 0.035 0.033 0.070 0.000 0.000 0.000
## [4,] 0.000 0.000 0.000 0.075 0.035 0.038
## [5,] 0.000 0.000 0.000 0.035 0.065 0.035
## [6,] 0.000 0.000 0.000 0.038 0.035 0.077
```

```
# outcome to character with a meaningful prefix
sim$p <- paste0("outcome", sim$p)

# fitting the model
res <- rma.mv(yi, V, mods = ~ 0 + p, random = ~p|id, data = sim, struct = "CS")
summary(res)
```

```
##
## Multivariate Meta-Analysis Model (k = 300; method: REML)
##
##      logLik   Deviance      AIC      BIC      AICc
## -204.4749   408.9499   418.9499   437.4185   419.1560
##
## Variance Components:
##
## outer factor: id (nlvls = 100)
```

```
## inner factor: p (nlvls = 3)
##
##          estim    sqrt  fixed
## tau^2      0.2828  0.5318    no
## rho        0.6750              no
##
## Test for Residual Heterogeneity:
## QE(df = 297) = 1368.8098, p-val < .0001
##
## Test of Moderators (coefficients 1:3):
## QM(df = 3) = 642.5366, p-val < .0001
##
## Model Results:
##
##          estimate      se      zval      pval      ci.lb
## poutcome1    0.0735  0.0593   1.2402  0.2149  -0.0427
## poutcome2   -0.0863  0.0591  -1.4599  0.1443  -0.2022
## poutcome3    1.0456  0.0591  17.6841 <.0001   0.9297
##          ci.ub
## poutcome1  0.1897
## poutcome2  0.0296
## poutcome3  1.1614 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output is more complicated compared to previous models. The **Variance Components** is the estimation of the T^2 matrix. In the two-level random-effects model, we have a single τ^2 while in the multivariate model, we have a full variance-covariance matrix. Using the `struct = "CS"` argument, we are forcing the `rma.mv()` function to use a compound symmetry structure, thus estimating a single τ^2 and a single ρ . Again, this is consistent with our simulation but could be a stringent assumption in a real-world analysis. The crucial part is the `random = ~ p|id` argument that specifies the random-effects structure. The syntax is clearly explained here <https://wviechthb.github.io/metafor/reference/rma.mv.html#specifying-random-effects>. The basic idea is to specify the random effect structure as `~ inner|outer` where `inner` is the outcome variable and `outer` is the paper variable. The `mods = ~ 0 + p` argument specifies to estimate the average effect for each outcome (i.e., the vector of average effects). We are removing the intercept (`~ 0 + p` or equivalently `~ p - 1`) and the model will estimate the mean for each level of the `p` variable.

We could use also another structure for the T matrix. We know that our data are generated using a CS structure thus using another structure will be less appropriate. In reality, is not known which is the real data generation model thus we can try different structure. Setting the `struct = "UN"` (unstructured) argument, `rma.mv` try to estimate a variance-covariance matrix where all correlations and variances need to be estimated. Compared to the previous model, we are now estimating 3 heterogeneity and three correlation parameters.

We could also use another structure for the T^2 matrix. We know that our data are generated using a CS structure; thus, using another structure will be less appropriate. In reality, it is not known which is the real data generation model; thus, we can try different structures. Setting the `struct = "UN"` (unstructured) argument, `rma.mv` tries to estimate a variance-covariance matrix where all correlations and variances need to be estimated. We are now estimating three heterogeneity and three correlation parameters compared to the previous model.

```
# fitting the model
res <- rma.mv(yi, V, mods = ~ 0 + p, random = ~p|id, data = sim, struct = "UN")
summary(res)
```

```

##
## Multivariate Meta-Analysis Model (k = 300; method: REML)
##
##      logLik    Deviance      AIC      BIC      AICc
## -200.8649    401.7297    419.7297    452.9733    420.3569
##
## Variance Components:
##
## outer factor: id (nlvls = 100)
## inner factor: p  (nlvls = 3)
##
##      estim    sqrt  k.lvl  fixed    level
## tau^2.1    0.2427  0.4926   100     no  outcome1
## tau^2.2    0.3244  0.5695   100     no  outcome2
## tau^2.3    0.2811  0.5302   100     no  outcome3
##
##      rho.otc1  rho.otc2  rho.otc3    otc1  otc2  otc3
## outcome1      1              -    100    100
## outcome2    0.5869      1      no      -    100
## outcome3    0.7035    0.7430      1    no    no    -
##
## Test for Residual Heterogeneity:
## QE(df = 297) = 1368.8098, p-val < .0001
##
## Test of Moderators (coefficients 1:3):
## QM(df = 3) = 789.4812, p-val < .0001
##
## Model Results:
##
##      estimate      se      zval      pval      ci.lb
## poutcome1    0.0737  0.0558   1.3219  0.1862  -0.0356
## poutcome2   -0.0858  0.0625  -1.3722  0.1700  -0.2084
## poutcome3    1.0467  0.0590  17.7480 <.0001   0.9311
##      ci.ub
## poutcome1  0.1831
## poutcome2  0.0368
## poutcome3  1.1623 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

5 Extra

5.1 I^2 with homogeneous vs heterogeneous sample sizes

In the paper we suggested that is not straightforward to simulate a meta-analysis fixing the I^2 as a simulation parameter with heterogeneous sample sizes. Here we show that when sample sizes are generated from a probability distribution (e.g., Gaussian) we can fix the I^2 value using the \tilde{v} calculated using the expected value of the sample size distribution. For example, if the sample sizes of our k studies is generate from a Poisson distribution $n_i \sim \mathcal{P}(\lambda)$, then \tilde{v} can be calculated using λ as *typical* sample size (i.e., as if the sample sizes were homogeneous). The following code simulate this scenario showing the distribution of I^2 with

heterogeneous sample sizes⁵.

```
set.seed(seed)
tau2 <- 0.2 # real tau2 value
navg <- 40 # average sample size
nmin <- 10 # minimum sample size
k <- 30 # number of studies
k <- 30 # number of studies
nsim <- 1000 # number of simulations
mu <- 0.3 # real average effect size

# here we simulate some effect sizes using the aggregated
# data approach, thus from the summary statistics distributions

I2 <- rep(0, nsim)

for(i in 1:nsim){
  ni <- nmin + rpois(k, navg - nmin)

  # effect sizes
  yi <- rnorm(k, rnorm(k, mu, sqrt(tau2)), sqrt(1/ni + 1/ni))

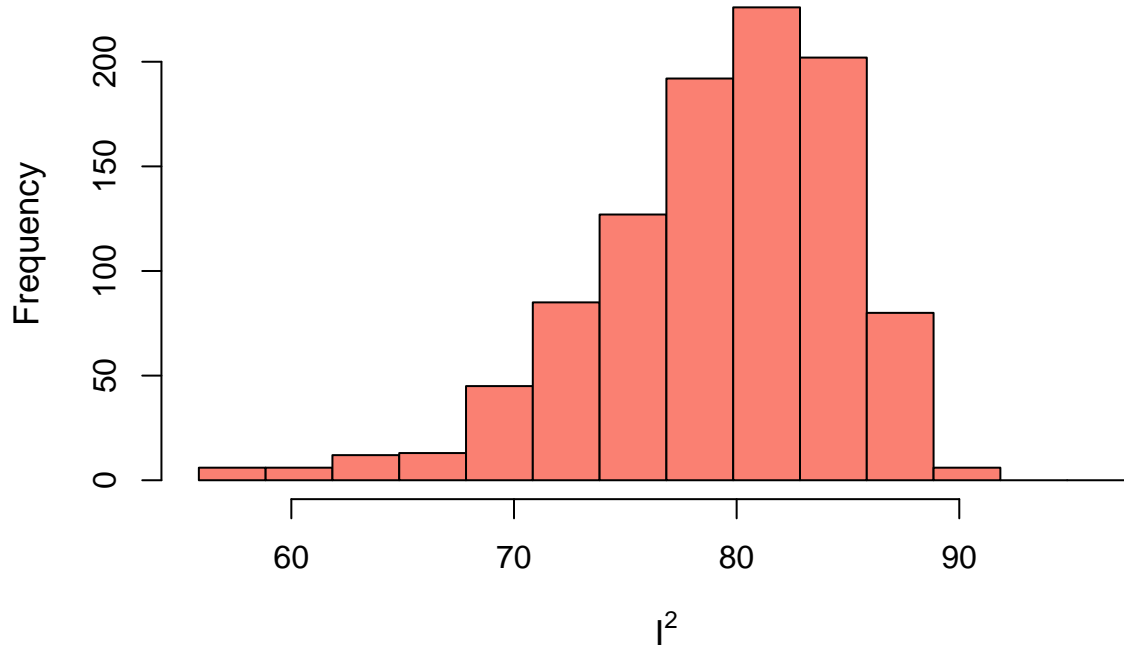
  # sampling variances
  vi <- (rchisq(k, ni + ni - 2) / (ni + ni - 2)) * (1/ni + 1/ni)

  # model
  fit <- rma(yi, vi)

  # extracting I2
  I2[i] <- fit$I2
}
```

⁵Compared to the simulations of the paper here we are simulating the effect size and the sampling variances from the respective sampling distributions (e.g., supplementary materials by Aert et al., 2019)

Simulated I^2 values (Mean = 79.138)



The I^2 calculated using the expected value of the sample sizes distribution ($\lambda = 40$) and assuming sample size homogeneity is:

```
v <- 1/navg + 1/navg
tau2 / (v + tau2)
```

```
## [1] 0.8
```

The value is very close to the mean of the simulated I^2 distribution. In conclusion, by generating the sample sizes from a probability distribution (e.g., Poisson) and using the Higgins and Thompson (2002) formula to calculate \tilde{v} we can somehow fix the I^2 with heterogeneous sample sizes using the expected value of the distribution.

6 Useful resources

- <https://www.jepusto.com/simulating-correlated-smnds/>
- <https://www.metafor-project.org/doku.php/metafor>
- <https://stat.ethz.ch/mailman/listinfo/r-sig-meta-analysis>
- https://bookdown.org/MathiasHarrer/Doing_Meta_Analysis_in_R/

7 References

Aert, R. C. M. van, Assen, M. A. L. M. van, & Viechtbauer, W. (2019). Statistical properties of methods based on the Q-statistic for constructing a confidence interval for the between-study variance in meta-

- analysis. *Res. Synth. Methods*, 10(2), 225–239. <https://doi.org/10.1002/jrsm.1336>
- Cheung, M. W.-L. (2014). Modeling dependent effect sizes with three-level meta-analyses: A structural equation modeling approach. *Psychol. Methods*, 19(2), 211–229. <https://doi.org/10.1037/a0032968>
- Cheung, M. W.-L. (2015). *Meta-Analysis: A structural equation modeling approach*. John Wiley & Sons.
- Cheung, M. W.-L. (2019). A guide to conducting a Meta-Analysis with Non-Independent effect sizes. *Neuropsychol. Rev.*, 29(4), 387–396. <https://doi.org/10.1007/s11065-019-09415-6>
- Higgins, J. P. T., & Thompson, S. G. (2002). Quantifying heterogeneity in a meta-analysis. *Stat. Med.*, 21(11), 1539–1558. <https://doi.org/10.1002/sim.1186>