Nested Pair-Design Hierarchical Model

A quick simulation study

Version 1.0

Holger Sennhenn-Reulen
Forest Inventory, Informatics & Biometrics,
Department of Growth and Yield,
Northwest German Forest Research Institute.

June 23, 2020

Contents

1 Organise R Session

```
## Organise R Session: ####
rm(list = ls())
library("countreg")
library("rcartocolor")
library("brms")
```

2 Simulation of Measurements

Design:

- ten sites
- one to five pairs per site
- one measurement for each element of a pair

$$y_i \sim \text{Normal}\left(\mu_i, \sigma^2\right)$$

index i for the basic observation units, population variance $\sigma^2 = 1$, and:

$$\mu_i = 0 + 0.5 \cdot \text{I}_{\{x_i = \text{Treatment}\}} + \gamma_i,$$

with indicator function I:

$$\text{I}_{\{\text{Condition}\}} = \begin{cases} 0, & \text{if condition is not met} \\ 1, & \text{if condition is met} \end{cases}$$

and

$$\gamma_i = \sum_{k=1}^{10} \left(\text{I}_{\{\text{site}_i = k\}} \left(\gamma_k + \sum_{j=1}^{n_k} \text{I}_{\left\{\text{site}_i = k \text{ and pair}_i = j\right\}} \gamma_{kj} \right) \right)$$

with

$$\gamma_k \sim \text{Normal}\left(0, \sigma^2_{\text{site}}\right), \qquad \gamma_{kj} \sim \text{Normal}\left(0, \sigma^2_{\text{pair}}\right).$$

```
set.seed(123456789)
sites <- LETTERS[1:10]
n_pairs_per_site <- sample(1:5, length(sites), replace = T)
df <- data.frame(site = NULL, pair = NULL, mu_k = NULL, mu_j = NULL,
                 stringsAsFactors = FALSE)
mu_k <- rnorm(length(sites))
for (k in 1:length(sites)) {
  mu_j <- rnorm(n_pairs_per_site[k])
  for (j in 1:n_pairs_per_site[k]) {
    df_k <- data.frame(site = LETTERS[k],
                       pair = letters[j],
                       x = c("Control", "Treatment"),
                       mu_k = mu_k[k],
                       mu_j = mu_j[j],
                       mu_x = c(0, 0.5),
                       stringsAsFactors = FALSE)
    df <- rbind(df, df_k)
  }
}
df$y <- df$mu_k + df$mu_j + df$mu_x + rnorm(nrow(df))
```

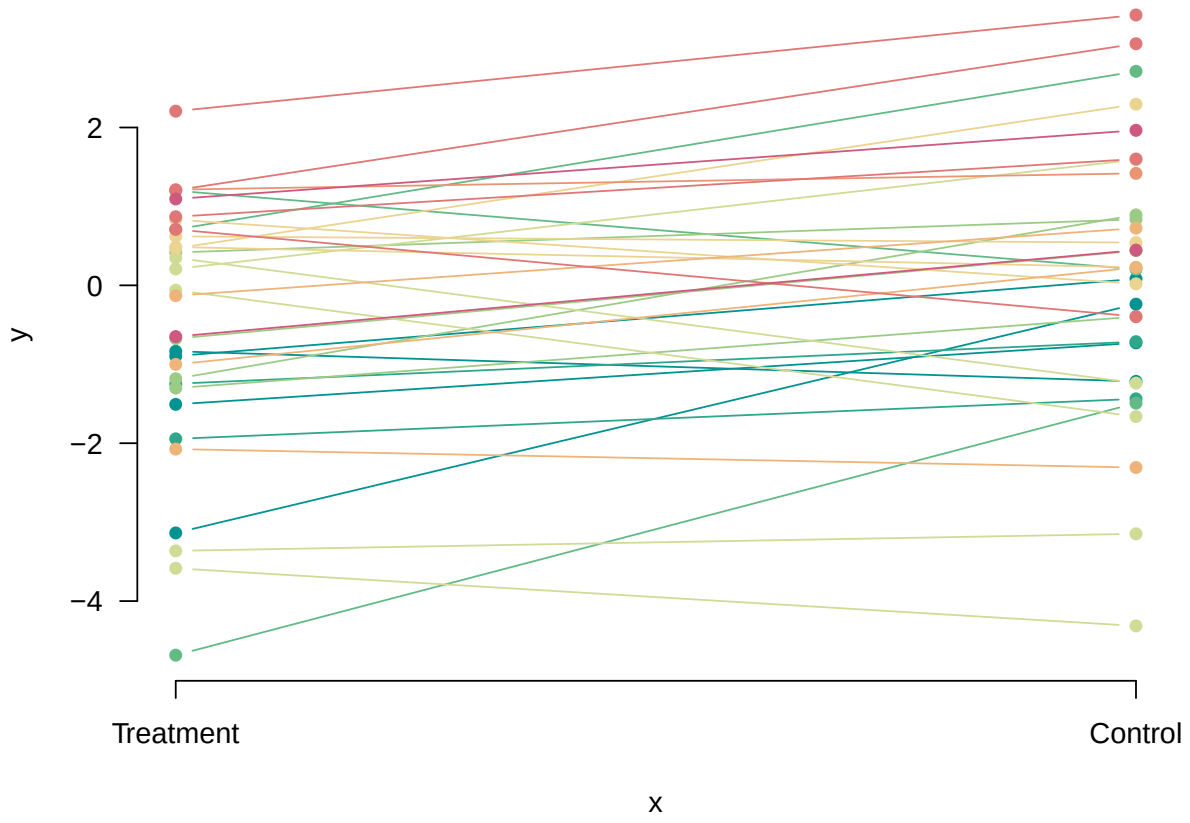

Figure 1: Simulated data of pairs (one element treatment with one element control) one ten different sited (colors).

3 Modeling

For comparison, a reduced model is constructed without including the structural information by the pair design:

$$\gamma_i = \sum_{k=1}^{10} \mathrm{I}_{\{\mathrm{site}_i = k\}} \gamma_k$$

In such a model, the present variation by γ_{k_j} is not correctly incorporated. As per site, the variation by pairs is symmetrical and central at 0, this non-inclusion should lead to an inflation of the population variation. This model compared to the correctly constructed model as defined by the simulation build-up. As the strength of the true *signal* of interest – the influence of x on the expected value of y – is not changed between the models, such an inflation of the estimated population variation should lead to a lower precision in the estimation of the signal.
In R (R Core Team, 2019) add-on package `brms` (Bürkner, 2017)– which very flexibly sets up Stan (Stan Development Team, 2019) regression models and provides an extensive toolbox to prepare and handle the results of such models – these models are – with defining a suitable set of priors – generated by:

```
frmla_0 <- bf(y ~ x + (1 | site))
frmla <- bf(y ~ x + (1 | site/pair))
pr_0 <- get_prior(frmla_0, data = df, family = "normal")
pr <- get_prior(frmla, data = df, family = "normal")
rm(df)
pr_0[pr_0$class == "b" & pr_0$coef == "xTreatment", "prior"] <- "normal(0, 2)"
pr[pr$class == "b" & pr$coef == "xTreatment", "prior"] <- "normal(0, 2)"
dummy_model_0 <- brm(frmla_0, data = df, family = "normal",
                     prior = pr_0, chains = 0)
dummy_model <- brm(frmla, data = df, family = "normal",
                   prior = pr, chains = 0)
```

The R-code given on the next page repeatedly runs simulation and estimation.

```
R <- 20
results <- vector("list", R)
set.seed(123456789)
for (r in 1:R) {
  n_pairs_per_site <- sample(1:5, length(sites), replace = T)
  df <- data.frame(site = NULL, pair = NULL, mu_k = NULL, mu_j = NULL,
                   stringsAsFactors = FALSE)
  mu_k <- rnorm(length(sites))
  for (k in 1:length(sites)) {
    mu_j <- rnorm(n_pairs_per_site[k])
    for (j in 1:n_pairs_per_site[k]) {
      df_k <- data.frame(site = LETTERS[k],
                         pair = letters[j],
                         x = c("Control", "Treatment"),
                         mu_k = mu_k[k],
                         mu_j = mu_j[j],
                         mu_x = c(0, 0.5),
                         stringsAsFactors = FALSE)
      df <- rbind(df, df_k)
    }
  }
  df$y <- df$mu_k + df$mu_j + df$mu_x + rnorm(nrow(df))
  fit_0 <- update(dummy_model_0, newdata = df, recompile = FALSE,
                  control = list(adapt_delta = 0.8, max_treedepth = 10),
                  chains = 4, cores = 4, iter = 2000, warmup = 1000,
                  seed = 123456789)
  fit <- update(dummy_model, newdata = df, recompile = FALSE,
                control = list(adapt_delta = 0.8, max_treedepth = 10),
                chains = 4, cores = 4, iter = 2000, warmup = 1000,
                seed = 123456789)
  tmp_0 <- fitted(fit_0, re_formula = NA, summary = FALSE,
                  newdata = data.frame(x = c("Control", "Treatment")))
  tmp <- fitted(fit, re_formula = NA, summary = FALSE,
                newdata = data.frame(x = c("Control", "Treatment")))
  res_0 <- apply(tmp_0, MAR = 1, FUN = diff)
  res <- apply(tmp, MAR = 1, FUN = diff)
  results[[r]] <- cbind(res_0, res,
                        as.matrix(fit_0)[, "sigma"], as.matrix(fit)[, "sigma"],
                        as.matrix(fit_0)[, "sd_site__Intercept"], as.matrix(fit)[, "sd_site__Intercept"])
}
```

4 Results

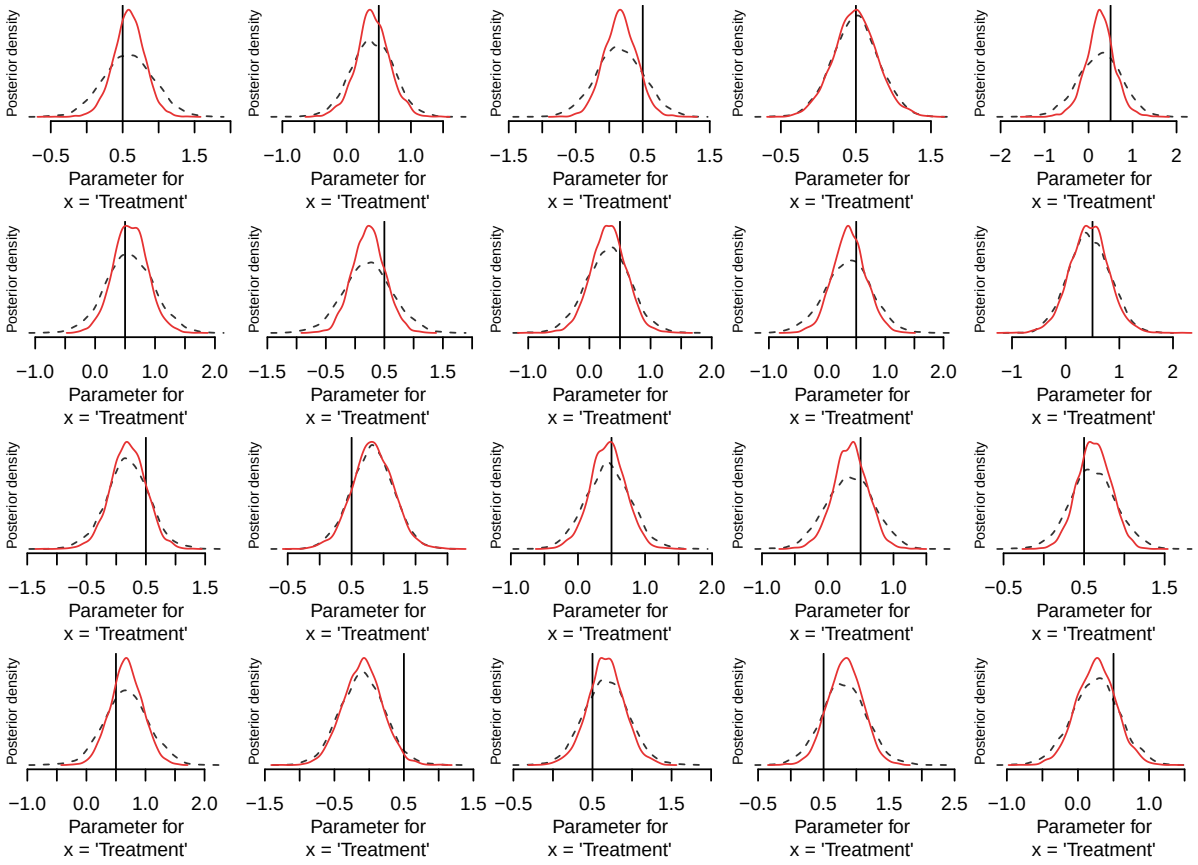

Figure 2: Posterior distribution – for each of 20 replications – for the change in expected value of y when changing x from *control* to *treatment* – underlying truth is 0.5. Each dashed black line is a kernel density estimator of the distribution estimated by model without including the structural information by the pair design, solid red line is the respective density estimator from the model including the structural pair information.

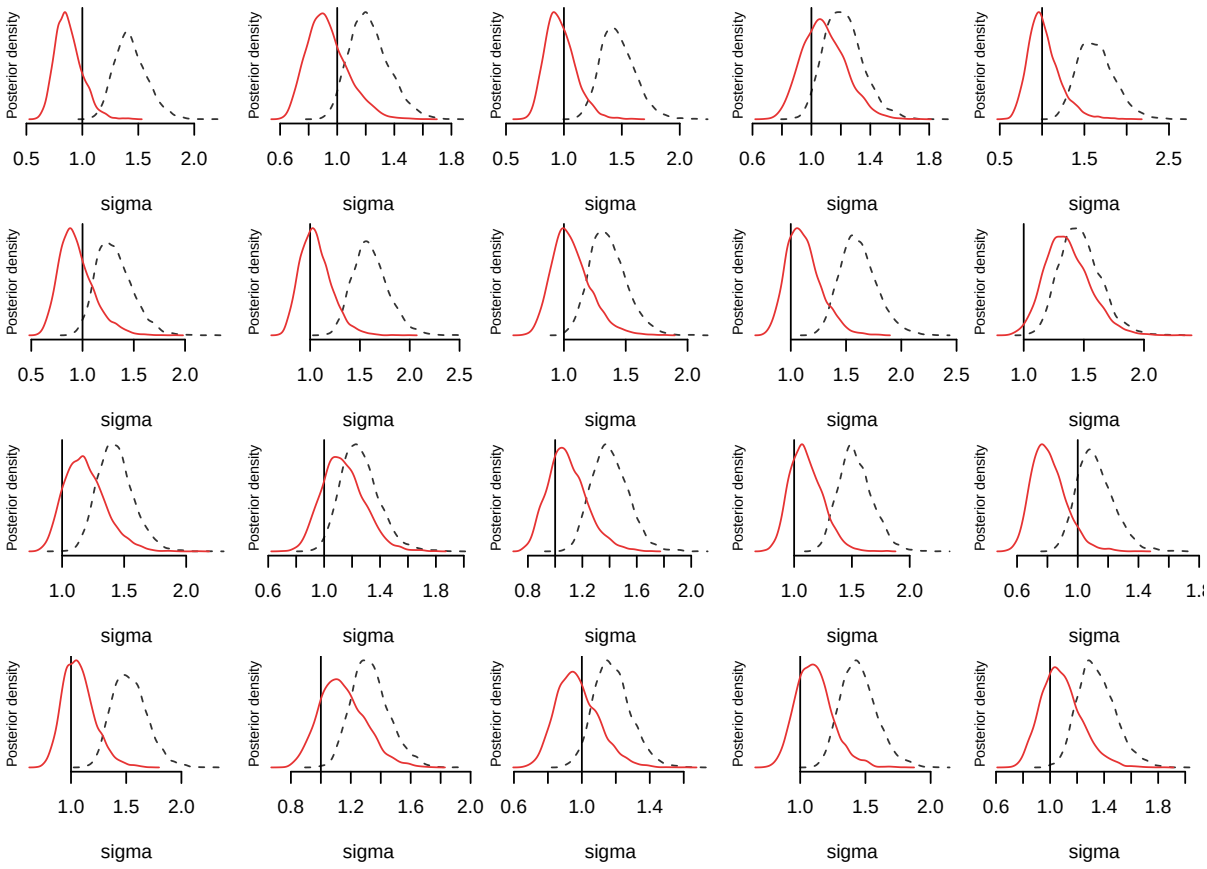

Figure 3: Posterior distribution – for each of 20 replications – for the population variation. Each dashed black line is a kernel density estimator of the distribution estimated by model without including the structural information by the pair design, solid red line is the respective density estimator from the model including this structural pair information.

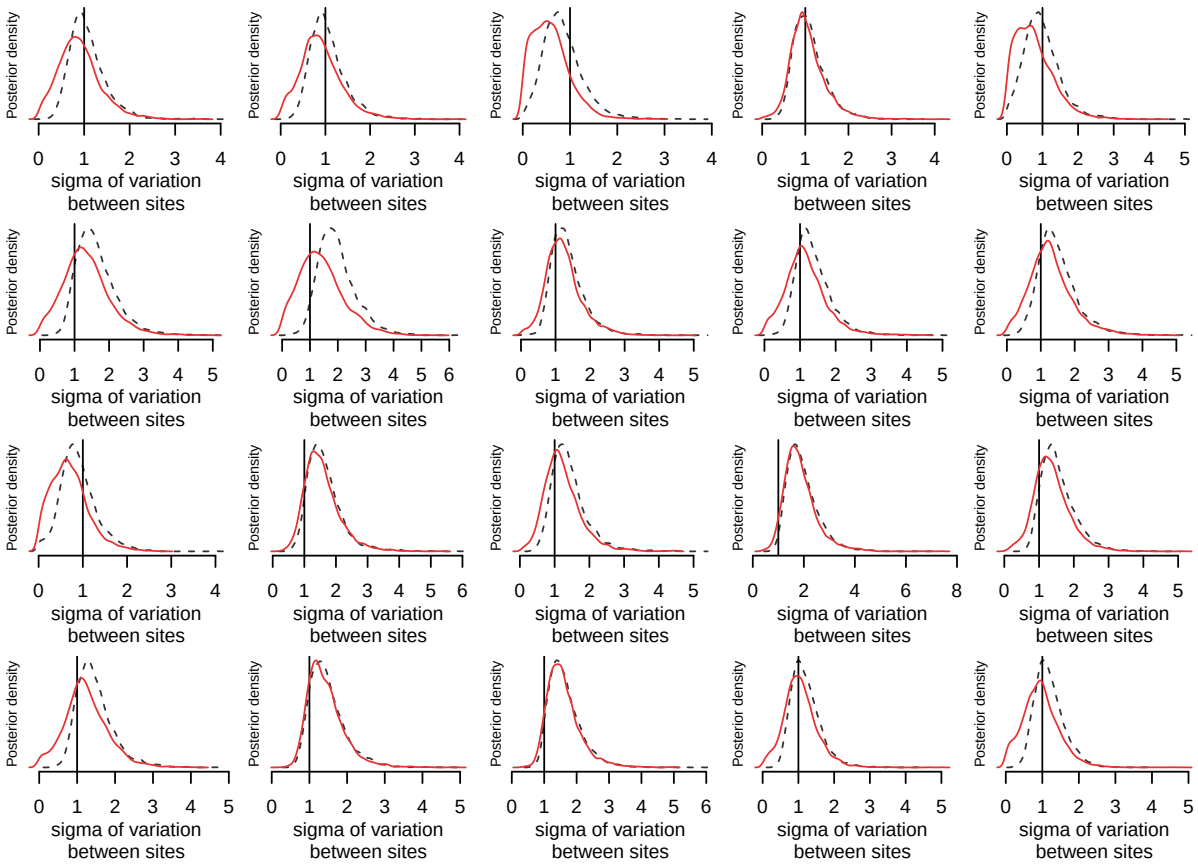

Figure 4: Posterior distribution – for each of 20 replications – for the variation between sites. Each dashed black line is a kernel density estimator of the distribution estimated by model without including the structural information by the pair design, solid red line is the respective density estimator from the model including this structural pair information.

References

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1):1–28.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Stan Development Team (2019). Stan: A C++ library for probability and sampling, version 2.19.2. http://mc-stan.org/.