

Specifics of Model

Here are the specifics (together with a conceptual example) of a model in which one would (or at least might) want to constrain σ to be 0. Consider a number N of students sitting a (say literacy) test on a number (n) of occasions. The response is the test score. There is a “student” random effect and a “time” fixed effect. The mathematical model for the structure that I envisage is:

$$Y_{ij} = \mu_i + S_{ij} \quad (1)$$

where μ_i is the population mean score at time i ($i = 1, \dots, n$) and S_{ij} is the departure from the population mean of the score of the j^{th} student at time i . I assume that the S_{ij} are Gaussian, with mean 0 and covariance matrix $\Sigma = [\sigma_{k\ell}]$ where $\sigma_{k\ell} = \text{cov}(S_{kj}, S_{\ell j})$. Of course S_{kj_1} and $S_{\ell j_2}$ are independent for $j_1 \neq j_2$.

Note that in this setting we actually have a straightforward multivariate observations structure. The fixed effects can be estimated as the column means of the $N \times n$ data matrix Y where the i^{th} row of Y consists of the observations for the i^{th} student. The covariance matrix can be estimated simply as $\text{var}(Y)$.

The objective in fitting the model using `lmer()` is really to be able to compare what one knows to be the right answer with what one gets out of `lmer()` so as to check that one is getting, in this simple instance, the syntax of the call to `lmer()` correct.

However `lmer()` won't let you fit that model. It insists, in effect, on tacking on an “overall error” term, E_j . That is, it tries to fit

$$Y_{ij} = \mu_i + S_{ij} + E_j \quad (2)$$

with the E_j having mean 0 and variance σ^2 . But this of course makes the model unidentifiable. For any value of σ^2 , model (2) with the covariance matrix of the S_{ij} given by

$$\Sigma_1 = \Sigma - \sigma^2 I$$

is equivalent to model (1), the only constraint being that $\Sigma - \sigma^2 I$ must be positive definite. There will be a range of values of σ^2 which satisfy this constraint.

To (attempt to) fit the model specified by (2) with `lmer()` it seems that one would use the syntax:

```
fit <- lmer(y ~ time + (time | student))
```

This does indeed return a result which seems to be sensible, but there is a worrisome aspect to it in that it contains and is dependent upon an estimate of the meaningless parameter σ .

The estimates of the “time” effect and their standard errors appear to be correct, i.e. they agree (after appropriate transformation) with estimates obtained by “multivariate analysis”.

The point estimates can be obtained by `m%*%apply(Y,2,mean)` and the standard errors by `sqrt(diag(m%*%var(Y)%*%t(m))/sqrt(N))` where `m` is the matrix

```

      [,1] [,2] [,3]
[1,]    1    0    0
[2,]   -1    1    0
[3,]   -1    0    1

```

which serves to convert the “multivariate” parameterization (μ_1, μ_2, μ_3) to the “linear models” parameterization $(\mu, \alpha_2, \alpha_3)$.

One can extract the covariance matrix (say “CM”) for the student effect using `VarCorr()` and transform it by the reverse of the transformation referred to above (i.e. by `solve(m)%*%CM%*%t(solve(m))`) so as to account for the difference in parameterizations.

The result differs from `var(Y)` (to within “numerical accuracy”) by a constant-diagonal matrix whose entries are equal to the square of the “sc” attribute of `VarCorr(fit)`. That is, the “sc” attribute seems to be an estimate of (the ill-defined) parameter σ . The `Variance` values associated with the `Random effects` in the `lmer()` output `fit` are equal to `diag(CM)`. That is they are affected by the value of the meaningless parameter estimate `sc = $\hat{\sigma}$` . This seems wrong to me; I think that they should actually be `diag(CM + s2*m%*%t(m))` where `s2` is equal to `sc^2` if you’re going to have `sc` kicking around at all. More properly, `CM` should have the value it would have if `sc` were 0, i.e. `m%*%var(Y)%*%t(m)`.

One could muck about making adjustments of the output of `lmer()` in respect of these `Variance` values but to do so would be at one’s peril.

It would be much nicer to be able to fit model (1) directly.