

Multilevel modeling worksheet: Exercise 3**Random slopes**

Q6 Sometimes (perhaps often) it makes sense to fit models with random slopes – that is models where the effects of interest vary across higher level units. For the pitch data the higher level units are people (it is a repeated measures model) and so this is equivalent to allowing individual differences in the effect of a predictor. For example, what if males don't always raise (or lower) pitch to the same degree when rating an attractive face? Perhaps some males raise their pitch and some males lower their pitch?

To fit a random slope model to the pitch data start with a two-level model with *base* and *attract* as predictors (but without *Face* nested within *Participant*) and put the predictor you want to model as random in addition to or in place of the constant for the random effect: `(1 + attract|Participant)` or `(1 + attract|Participant)` are equivalent. Again assign the model to a new object such as `pitch.rs` (or similar).

a) Compare the fit of the `pitch3.fe` and `pitch.rs` models using a LRT. *What does this comparison suggest?*

b) Obtain a 95% CI for the effect of *attract* via profiling. Note down any problems you encounter.

Q7 Now use the `MCMCglmm` package.¹

Start by running a small number of iterations to see if everything is OK.

```
library(MCMCglmm)
nsims <- 75000
pitch.mcmc.rs <- MCMCglmm(pitch ~ base + attract, random= ~
  us(1+attract):Participant, nitt=nsims, data=pitch.dat)
```

a) Do you encounter any problems? If so, make a note of them.

¹ Install it within R Studio or use `install.packages('MCMCglmm')` from the R console.

Try a simpler variance-covariance structure (with no covariance term):

```
pitch.mcmc.rs <- MCMCglmm(pitch ~ base + attract, random= ~
  idh(1+attract):Participant, nitt=nsims, data=pitch.dat)
```

b) Plot the MCMC trace using plot(pitch.mcmc.rs) and hitting return to cycle through the graphs. What you think?

c) Obtain the summary of the model. What is the 95% HPD for the effect of attract?

Optional

You might like to plot the random effects. There are various way to do this, but a caterpillar plot is one common way. One option is to use the lattice package:

```
library(lattice)
dotplot(ranef(pitch.rs, condVar=TRUE))
qqmath(ranef(pitch.rs, condVar=TRUE))
```

However, there are lots of other ways. For instance there is a pretty function written for the very popular ggplot2 function that illustrates how you can customize the approach. First you need to copy and past the ggCaterpillar() function from the author (who posted it on stackexchange):

<http://stackoverflow.com/questions/13847936/in-r-plotting-random-effects-from-lmer-lme4-package-using-qqmath-or-dotplot>

Then you can just run the following commands:

```
library(ggplot2)
ggCaterpillar(ranef(pitch.rs, condVar=TRUE))
```