# PSYC 7720 Lab

Lab 13 Activity

Andrew Graves, Department of Psychology, University of Virginia

## Directions:

A. Download the **Pixel** dataset from the *nlme* package in R. Call ?Pixel at the R console to obtain more information about this dataset. You will be fitting various random effects structures with the *lmerTest* package in R, which is a direct wrapper of the *lme4* package we have been using up until this point. The outcome variable will be *pixel* for each model, and the observational unit will be *dog*. Call ranef() on the model for each question. Do your best to handle convergence warnings if you receive them.
B. Answer the following questions and save the code you used in an R script.
C. You have until the end of lab to complete.

```
library(tidyverse)
library(nlme)
library(lmerTest)
options(contrasts = c("contr.sum","contr.poly"))

# Set a reasonable optimizer
my_opt <- lmerControl(optimizer = "bobyqa",
                      optCtrl = list(maxfun = 2e6))

data(Pixel)
```

## Questions:

1. In exactly two function calls, make all the column names of **Pixel** lowercase, and rename the dataset **dat**. For review, specify a repeated measurements two-way ANOVA with *side* and *day* as the repeated measurements "factors" (keep *day* as numeric for convergence purposes).

```
# Here is a useful blogpost for addressing convergence issues:
# https://joshua-nugent.github.io/allFit/

dat <- Pixel %>%
  rename_all(str_to_lower)

set.seed(42)
mod1 <- lmer(pixel ~ day * side + (1|dog/day) + (1|dog/side) + (1|dog/day:side),
                     data = dat, control = lmerControl(optimizer = "optimx",
                     check.nobs.vs.nlev = "ignore", check.nobs.vs.nRE = "ignore",
                     optCtrl = list(method = "hjkb", maxit = 1e9)))
rand_ef <- ranef(mod1)
map(rand_ef, head)
```

```
## $`day:side:dog`
```

```
##       (Intercept)
## 0:L:1  -5.9472877
## 0:L:2   0.5151689
## 0:L:3   1.3788978
## 0:L:4  -5.0615703
## 0:R:1   5.2513602
## 0:R:2  -1.7115796
##
## $`day:dog`
##     (Intercept)
## 0:1   -3.125958
## 0:2   -5.374023
## 0:3  -26.432606
## 0:4  -50.383073
## 1:1    1.811111
## 1:2   -4.300240
##
## $`side:dog`
##      (Intercept)
## L:1   -10.015885
## L:10   27.805818
## L:2    -8.768246
## L:3   -13.274729
## L:4    -3.574414
## L:5    -1.192570
##
## $dog
##    (Intercept) (Intercept) (Intercept)
## 1   -4.4556883  -14.519921  -15.588505
## 10   2.7480402    8.955143    9.614191
## 2   -3.7292882  -12.152773  -13.047149
## 3   -1.8230734   -5.940919   -6.378137
## 4    0.5344775    1.741722    1.869903
## 5    2.8479963    9.280874    9.963894
```

2. Specify additive fixed effects of *day* and *side*. Specify a random slope for *day* and a random intercept for *dog*. Make sure *day* is numeric for this question and the remainder of the exercise.

```
mod2 <- lmer(pixel ~ day + side + (day|dog), data = dat,
             control = my_opt)
ranef(mod2)
```

```
## $dog
##    (Intercept)         day
## 1   -46.689156  0.56547436
## 10   28.860472 -0.48907165
## 2   -40.847817  0.72641162
## 3   -21.640460  0.47120176
## 4    -1.626615  0.69803341
## 5    33.175365 -0.66746126
## 6     8.404725 -0.37442533
## 7    48.234889 -1.17892577
## 8     2.252070  0.06741901
## 9   -10.123473  0.18134386
##
```

```
## with conditional variances for "dog"
```

3. Fit the same model from 2, but do not estimate the correlation between the random slope and random intercept.

```
mod3 <- lmer(pixel ~ day + side + (day||dog), data = dat)
ranef(mod3)
```

```
## $dog
##    (Intercept)          day
## 1   -42.649829 -0.050800282
## 10   26.534368  0.018403540
## 2   -35.939727  0.001401480
## 3   -17.627199  0.015271267
## 4     4.050118  0.126271769
## 5    27.588457 -0.015984854
## 6     5.306989 -0.044581352
## 7    38.061551 -0.055216019
## 8     2.904338  0.011039706
## 9    -8.229067 -0.005805255
##
## with conditional variances for "dog"
```

4. Fit the same model from 2, but remove the random intercept entirely.

```
mod4 <- update(mod2, pixel ~ day + side + (0 + day|dog))
ranef(mod4)
```

```
## $dog
##           day
## 1  -4.4596336
## 10  4.0584799
## 2  -3.4348829
## 3  -1.6425702
## 4   0.3152442
## 5   2.4419165
## 6   0.1587435
## 7   2.8685276
## 8   0.2339671
## 9  -0.5397923
##
## with conditional variances for "dog"
```

5. Which out of the models from questions 2 through 4 is most conservative and why?

- Model 2 is the most conservative because it estimate the most parameters, as well as shrinks the degrees of freedom for the fixed effects to the largest degree. Model 2 is arguably the most desirable random effects structure, as it follows the principle of keeping it maximal. We can safely interpret the fixed effect of day. In order to interpret the fixed effect of side, we need to specify it as a random effect as well.

6. Redo the procedures from questions 2 through 5, but specify random slopes for *day* **AND** *side*.

```
# Random slopes for side and day within dog
mod6a <- update(mod2, pixel ~ day + side + (side + day|dog))
ranef(mod6a)
```

```
## $dog
##    (Intercept)      side1        day
```

```
## 1  -46.2216295  -1.364587  0.3595935
## 10  25.6257976  19.298276  0.0997153
## 2  -41.2315421  -1.117774  0.7061734
## 3  -21.6441508  -8.972562  0.4322341
## 4   -4.0561067  -3.479300  0.9815675
## 5   35.1561951  -5.425574 -0.8829874
## 6    9.2571660   5.272361 -0.4886792
## 7   54.5463737 -15.786292 -1.8863929
## 8    0.5897232   3.467645  0.2567095
## 9  -12.0218264   8.107806  0.4220663
##
## with conditional variances for "dog"
```

```
# Uncorrelated random parameters for side and day within dog
# This model is not identifiable with any of the off-the shelf optimizers!
# That's ok because we want mod6a anyways.
mod6b <- update(mod2, pixel ~ day + side + (side + day||dog))

# Remove the random intercept but keep the random slopes
mod6c <- update(mod2, pixel ~ day + side + (0 + side + day||dog))
ranef(mod6c)
```

```
## $dog
##          sideL       sideR         day
## 1  -43.5827603 -41.4622410 -0.21747318
## 10  45.1716988   7.4447318  0.12345518
## 2  -38.1451736 -35.9651725  0.08763162
## 3  -27.9371716  -9.8193789  0.16104647
## 4   -4.5783809   3.5797398  0.67017596
## 5   23.8488872  34.3961708 -0.13956973
## 6   13.2487743   2.0516512 -0.30759940
## 7   27.0226258  57.1578577 -0.44381697
## 8    5.5990965  -0.8789691  0.09946981
## 9   -0.6475963 -16.5043898 -0.03331976
##
## with conditional variances for "dog"
```