Mixed models

Mixed models in R

Showing similarities between `aov_car`, `aov_4`, and `mixed` from the `afex` package and `lme4()` from the `lme4` package.

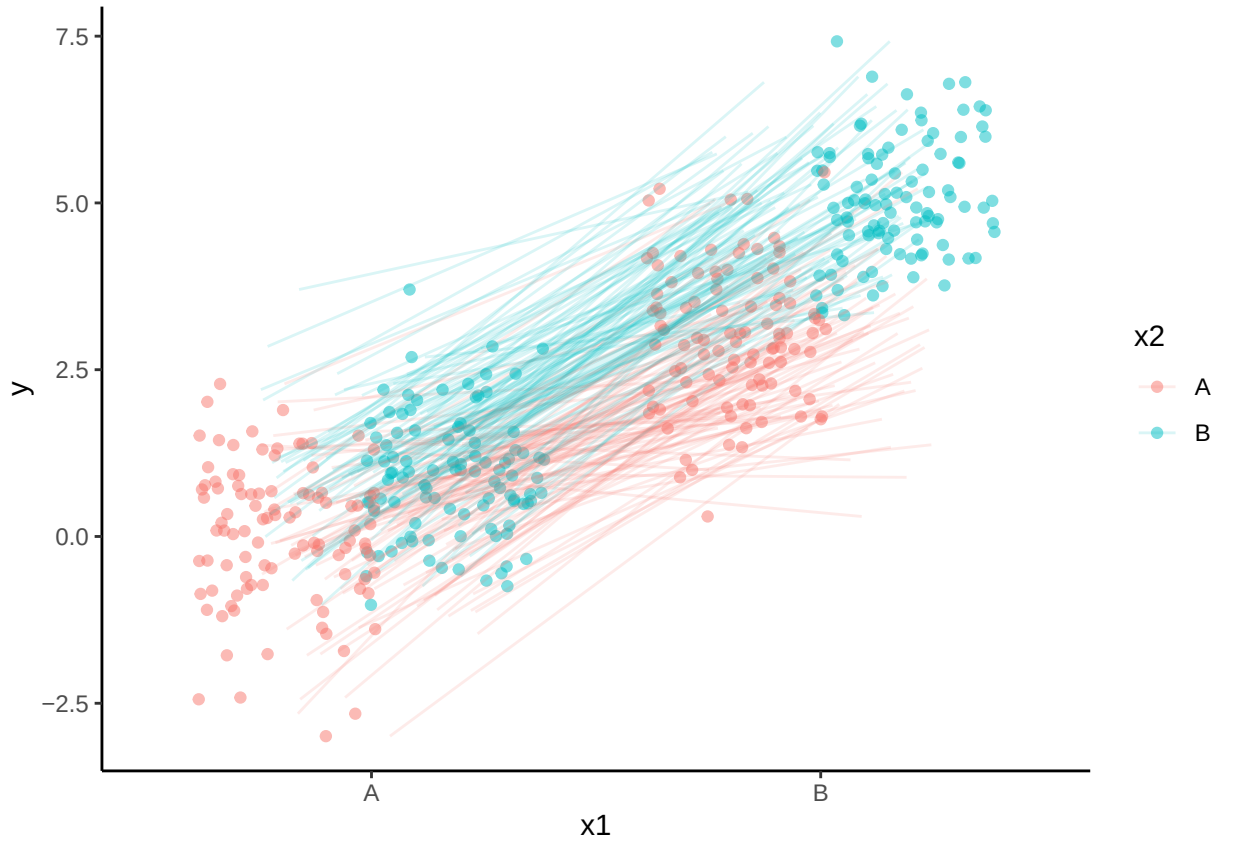
Simulate some data

- `y` our normally distributed DV
- `x1` a categorical within-subjects IV
- `x2` another categorical within-subjects IV
- `id` subject identifier

```
set.seed(42)
n_sub = 100
data <- tibble(y = c(rnorm(n_sub,0), rnorm(n_sub,1),rnorm(n_sub,3),rnorm(n_sub,5)),
            x1  = rep(c("A", "B"), each = n_sub*2),
            x2 = rep(c("A", "B"), each = n_sub, 2),
            id = rep(1:n_sub, 4)) %>%
  mutate(id = as.factor(id))
head(data)
```

```
## # A tibble: 6 x 4
##        y x1    x2    id
##    <dbl> <chr> <chr> <fct>
## 1  1.37  A     A     1
## 2 -0.565 A     A     2
## 3  0.363 A     A     3
## 4  0.633 A     A     4
## 5  0.404 A     A     5
## 6 -0.106 A     A     6
```

```
ggplot(data, aes(x1, y, group=x2, col=x2)) +
  geom_point(alpha=.5, position = position_jitterdodge()) +
  geom_path(aes(x = x1, y = y, group = interaction(x2, id)), alpha=.15, position=position_dodge(width=.5
  theme_classic()
```



aov_car

```
mod_car <- data %>%
  aov_car(y ~ x1 + x2 + Error(id/x1:x2), data = .)
mod_car
```

```
## Anova Table (Type 3 tests)
##
## Response: y
##   Effect    df  MSE           F ges p.value
## 1     x1 1, 99 0.93 1351.11 *** .77  <.0001
## 2     x2 1, 99 0.79  269.79 *** .37  <.0001
## 3  x1:x2 1, 99 1.07   31.57 *** .08  <.0001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

aov_4

```
mod_4 <- data %>%
  aov_4(y ~ x1 * x2 + (x1*x2|id), data = .)
mod_4
```

```
## Anova Table (Type 3 tests)
##
## Response: y
##   Effect    df  MSE           F ges p.value
## 1     x1 1, 99 0.93 1351.11 *** .77  <.0001
## 2     x2 1, 99 0.79  269.79 *** .37  <.0001
## 3  x1:x2 1, 99 1.07   31.57 *** .08  <.0001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

mixed

```
mod_mixed <- data %>%
  mixed(y ~ x1 * x2 + (1|id), data = .)
```

```
## Contrasts set to contr.sum for the following variables: id
```

```
## Fitting one lmer() model.
```

```
## boundary (singular) fit: see ?isSingular
```

```
## [DONE]
## Calculating p-values. [DONE]
```

```
summary(mod_mixed)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: y ~ x1 * x2 + (1 | id)
##    Data: data
##
## REML criterion at convergence: 1111.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.14418 -0.66832 -0.00715  0.66362  2.89869
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  id       (Intercept) 1.234e-17 3.512e-09
##  Residual             9.260e-01 9.623e-01
## Number of obs: 400, groups:  id, 100
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)   0.03251    0.09623 396.00000   0.338    0.736
## x1B           2.95712    0.13609 396.00000  21.729  < 2e-16 ***
## x2B           0.88000    0.13609 396.00000   6.466 2.96e-10 ***
## x1B:x2B       1.16330    0.19246 396.00000   6.044 3.46e-09 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) x1B    x2B
## x1B     -0.707
## x2B     -0.707  0.500
## x1B:x2B  0.500 -0.707 -0.707
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

lmer

```
mod_lmer <- data %>%
  lmer(y ~ x1 * x2 + (1|id), data = .)
```

```
## boundary (singular) fit: see ?isSingular
```

```
summary(mod_lmer)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: y ~ x1 * x2 + (1 | id)
##    Data: .
##
## REML criterion at convergence: 1111.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.14418 -0.66832 -0.00715  0.66362  2.89869
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  id       (Intercept) 1.234e-17 3.512e-09
##  Residual             9.260e-01 9.623e-01
## Number of obs: 400, groups:  id, 100
##
## Fixed effects:
##             Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)   0.03251    0.09623 396.00000   0.338    0.736
## x1B           2.95712    0.13609 396.00000  21.729  < 2e-16 ***
## x2B           0.88000    0.13609 396.00000   6.466 2.96e-10 ***
## x1B:x2B       1.16330    0.19246 396.00000   6.044 3.46e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) x1B    x2B
## x1B     -0.707
## x2B     -0.707  0.500
## x1B:x2B  0.500 -0.707 -0.707
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Compare marginal means

For x1

```
emmeans(mod_car, ~x1)
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  x1 emmean     SE  df lower.CL upper.CL
##  A   0.473 0.0678 198    0.339    0.606
##  B   4.011 0.0678 198    3.878    4.145
##
## Results are averaged over the levels of: x2
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

```
emmeans(mod_4, ~x1)
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  x1 emmean     SE  df lower.CL upper.CL
##  A   0.473 0.0678 198    0.339    0.606
##  B   4.011 0.0678 198    3.878    4.145
##
## Results are averaged over the levels of: x2
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

```
emmeans(mod_mixed, ~x1)
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  x1 emmean    SE  df lower.CL upper.CL
##  A   0.473 0.068 297    0.339    0.606
##  B   4.011 0.068 297    3.877    4.145
##
## Results are averaged over the levels of: x2
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

```
emmeans(mod_lmer, ~x1)
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  x1 emmean    SE  df lower.CL upper.CL
##  A   0.473 0.068 297    0.339    0.606
##  B   4.011 0.068 297    3.877    4.145
##
## Results are averaged over the levels of: x2
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

For x2

```
emmeans(mod_car, ~x2)
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  x2 emmean     SE  df lower.CL upper.CL
##  A    1.51 0.0653 197     1.38     1.64
##  B    2.97 0.0653 197     2.84     3.10
##
## Results are averaged over the levels of: x1
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

```
emmeans(mod_4, ~x2)
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  x2 emmean     SE  df lower.CL upper.CL
##  A    1.51 0.0653 197     1.38     1.64
##  B    2.97 0.0653 197     2.84     3.10
##
## Results are averaged over the levels of: x1
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

```
emmeans(mod_mixed, ~x2)
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  x2 emmean    SE  df lower.CL upper.CL
##  A    1.51 0.068 297     1.38     1.64
##  B    2.97 0.068 297     2.84     3.11
##
## Results are averaged over the levels of: x1
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

```
emmeans(mod_lmer, ~x2)
```

```
## NOTE: Results may be misleading due to involvement in interactions

##  x2 emmean    SE  df lower.CL upper.CL
##  A    1.51 0.068 297     1.38     1.64
##  B    2.97 0.068 297     2.84     3.11
##
## Results are averaged over the levels of: x1
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

For the x1:x2 interaction

```r
emmeans(mod_car, ~x1*x2)
```

```
##  x1 x2 emmean     SE  df lower.CL upper.CL
##  A  A  0.0325 0.0962 392   -0.157    0.222
##  B  A  2.9896 0.0962 392    2.800    3.179
##  A  B  0.9125 0.0962 392    0.723    1.102
##  B  B  5.0329 0.0962 392    4.844    5.222
##
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

```r
emmeans(mod_4, ~x1*x2)
```

```
##  x1 x2 emmean     SE  df lower.CL upper.CL
##  A  A  0.0325 0.0962 392   -0.157    0.222
##  B  A  2.9896 0.0962 392    2.800    3.179
##  A  B  0.9125 0.0962 392    0.723    1.102
##  B  B  5.0329 0.0962 392    4.844    5.222
##
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

```r
emmeans(mod_mixed, ~x1*x2)
```

```
##  x1 x2 emmean     SE  df lower.CL upper.CL
##  A  A  0.0325 0.0962 396   -0.157    0.222
##  B  A  2.9896 0.0962 396    2.800    3.179
##  A  B  0.9125 0.0962 396    0.723    1.102
##  B  B  5.0329 0.0962 396    4.844    5.222
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

```r
emmeans(mod_lmer, ~x1*x2)
```

```
##  x1 x2 emmean     SE  df lower.CL upper.CL
##  A  A  0.0325 0.0962 396   -0.157    0.222
##  B  A  2.9896 0.0962 396    2.800    3.179
##  A  B  0.9125 0.0962 396    0.723    1.102
##  B  B  5.0329 0.0962 396    4.844    5.222
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```