Stop Using Repeated Measures ANOVA (working title)

Maya Mathur & Mike Frank

Broad Outline

1. RM-ANOVA is the same as but generally worse than LMM
2. Simulations show this
3. LMM is not a panacea, there are some other possibilities, inc. GEE and BLMM

Intro

Pro of other methods:

Con of LMM: - a little more complicated - convergence problems

We won't even get into generalized lm

Theory

1. Here are the assumptions of RM-ANOVA. It's a special case of LMM under the following restrictions.
2. You can relax these in LMM, or GEE, or Bayes LMM.
3. Simulations show this
4. LMM is not a panacea. There are some other possibilities, inc. GEE and BLMM
5. Each of these alternatives has pros and cons, and they address slightly different questions.

Table: pros vs. cons

1. When estimation of random intercepts/slopes themselves or of their variance is of interest (e.g., when they represent specific stimuli), LMM makes sense
2. When these are treated as nuisance parameters (e.g., specific subjects), consider GEE
3. LMM has strong parametric assumptions, unlike OLS and its direct extension, GEE. In practice, it is usually fairly robust.
4. Some of LMM's best properties (e.g., straightforward handling of missing data) are only achieved at the direct cost of above parametric assumptions.
5. Brief mention of conditional/marginal interpretation problem with non-ID link function
6. Bayesian methods make most sense when requiring arbitrary flexibility (pretty sure Gelman text has good examples). (Con: as far as I know, mainstream Bayesian methods are going to be fully parametric as well.)
7. Reiterate that any theoretical weakness of LMM is a weakness of RANOVA as well because latter is a special case. Computationally, when specification is equivalent, maybe RANOVA is useful for closed-form p-values and convergence?

Practical issues of RM-ANOVA: Simulations

Data

1. simulate from models
2. mental abacus - longitudinal
3. another applied dataset

Issue: Power/nominal alpha

- Under H_A, RANOVA might be less powerful?

- Under H_0, we should see nominal performance for both.

Issue: Missing data

- If LMM's distributional assumptions hold, then it automatically handles MAR missing data unbiasedly because it can just maximize the likelihood anyway.
- RANOVA uses closed-form sums of squares, so presumably mainstream software will do complete-cases. Point to existing missing data literature for why this is usually horrible.
- Simulation: Fit equivalent RANOVA and LMM specifications to MAR missing data; RANOVA will be biased and inefficient.

Issue: Distributional assumptions/specification flexibility

- RANOVA requires categorical covariates
- When truly continuous covariates are dichotomized, this is not great[**?**]
- Probably does not require simulation.

Issue: Pre-aggregation

- Does not lead to the same model specification! Leads to different distributional assumptions.
- E.g., if there is correlation within both subjects and items, and you fit the model to subject means rather than observations, then you are assuming normality on the item intercepts but not on the subjects.
- Simulation: Generate data with large number of observations/subject and relatively few subjects. Pre-aggregated should lose power.

Mental abacus example

```
ma <- read_csv("data/zenith.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   placeval = col_double(),
##   ravens = col_double(),
##   wiat = col_double(),
##   woodcock = col_double(),
##   arith = col_double(),
##   mental.rot.letters.prop = col_double(),
##   commute.prop = col_double(),
##   mental.rot.shapes.prop = col_double(),
##   wholegroupsums = col_double(),
##   age = col_double(),
##   condition = col_character(),
##   class = col_character(),
##   verbalwm = col_double(),
##   spatialwm = col_double(),
##   ans = col_double()
## )

## See spec(...) for full column specifications.
```

```
lmer_mod <- lmer(arith ~ condition * year + (year | subnum),
                 data = ma)
summary(lmer_mod)
```

```
## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula: arith ~ condition * year + (year | subnum)
##    Data: ma
##
## REML criterion at convergence: -1580.7
##
## Scaled residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -3.05565 -0.57333 -0.00666  0.57125  2.77786
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  subnum   (Intercept) 0.000848 0.02912
##           year        0.001074 0.03278  1.00
##  Residual             0.004577 0.06765
## Number of obs: 788, groups:  subnum, 206
##
## Fixed effects:
##                          Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)              0.126802   0.006401 324.000000  19.810  < 2e-16
## conditioncontrol        -0.025777   0.008961 325.800000  -2.877  0.00428
## year                     0.103582   0.004630 216.900000  22.372  < 2e-16
## conditioncontrol:year   -0.020799   0.006399 215.000000  -3.250  0.00134
##
## (Intercept)             ***
## conditioncontrol        **
## year                    ***
## conditioncontrol:year **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cndtnc year
## condtncntrl -0.714
## year        -0.166  0.118
## cndtncntrl:  0.120 -0.165 -0.723
```

```
lmm_coef <- coef(lmer_mod)
```

```r
aov_mod <- aov(arith ~ condition * year + Error(subnum / year), data = ma)
summary(aov_mod)
```

```
##
## Error: subnum
##           Df   Sum Sq  Mean Sq
## condition  1 0.002993 0.002993
##
## Error: subnum:year
##           Df Sum Sq Mean Sq
## condition  1  6.576   6.576
##
## Error: Within
##                 Df Sum Sq Mean Sq F value   Pr(>F)
## condition        1  0.632  0.6323   53.92 5.24e-13 ***
## year             1  2.194  2.1940  187.11  < 2e-16 ***
## condition:year   1  0.132  0.1324   11.29 0.000818 ***
## Residuals      782  9.170  0.0117
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# rm_coef(coef(aov_mod))
```

```r
# blmer_mod <- brm(arith ~ condition * year + (year | subnum),
#                  data = ma, family = "gaussian")
#
# summary(blmer_mod)
```

Simulations

- basic power on the same dataset
- missing data
- pre-aggregation

References