Here is a demo of how to conduct the bootstrap analysis, more simply (no need to make your data wide!)

**1. Fit your repeated-measures model with lmer**

library(lme4)

data(obk.long, package = "afex") # data from the afex package

fit\_mixed <- lmer(value ~ treatment \* gender \* phase \* hour + (1|id),

data = obk.long)

Note that I assume here data is aggregated (one value per cell/subject), as it would be in a rmANOVA, as so it is sufficient to model only a random intercept.

**2. Define the contrast(s) of interest**

For this step we will be using emmeans to get the estimates of the pairwise differences between the treatment groups within each phase of the study:

library(emmeans)

# get the correct reference grid with the correct ultivariate levels!

rg <- ref\_grid(fit\_mixed, mult.levs = rm\_levels)

rg

## 'emmGrid' object with variables:

## treatment = control, A, B

## gender = F, M

## phase = fup, post, pre

## hour = 1, 2, 3, 4, 5

# get the expected means:

em\_ <- emmeans(rg, ~ phase \* treatment)

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

em\_

## phase treatment emmean SE df lower.CL upper.CL

## fup control 4.33 0.603 13.2 3.03 5.64

## post control 4.08 0.603 13.2 2.78 5.39

## pre control 4.25 0.603 13.2 2.95 5.55

## fup A 7.25 0.661 13.2 5.82 8.68

## post A 6.50 0.661 13.2 5.07 7.93

## pre A 5.00 0.661 13.2 3.57 6.43

## fup B 7.29 0.505 13.2 6.20 8.38

## post B 6.62 0.505 13.2 5.54 7.71

## pre B 4.17 0.505 13.2 3.08 5.26

##

## Results are averaged over the levels of: gender, hour

## Degrees-of-freedom method: kenward-roger

## Confidence level used: 0.95

# run pairwise tests between the treatment groups within each phase

c\_ <- contrast(em\_, "pairwise", by = 'phase')

c\_

## phase = fup:

## contrast estimate SE df t.ratio p.value

## control - A -2.9167 0.895 13.2 -3.259 0.0157

## control - B -2.9583 0.787 13.2 -3.760 0.0061

## A - B -0.0417 0.832 13.2 -0.050 0.9986

##

## phase = post:

## contrast estimate SE df t.ratio p.value

## control - A -2.4167 0.895 13.2 -2.700 0.0445

## control - B -2.5417 0.787 13.2 -3.230 0.0166

## A - B -0.1250 0.832 13.2 -0.150 0.9876

##

## phase = pre:

## contrast estimate SE df t.ratio p.value

## control - A -0.7500 0.895 13.2 -0.838 0.6869

## control - B 0.0833 0.787 13.2 0.106 0.9938

## A - B 0.8333 0.832 13.2 1.002 0.5885

##

## Results are averaged over the levels of: gender, hour

## P value adjustment: tukey method for comparing a family of 3 estimates

# extract the estimates

est\_names <- c("fup: control - A", "fup: control - B", "fup: A - B",

"post: control - A", "post: control - B", "post: A - B",

"pre: control - A", "pre: control - B", "pre: A - B")

est\_values <- summary(c\_)$estimate

names(est\_values) <- est\_names

est\_values

## fup: control - A fup: control - B fup: A - B post: control - A

## -2.91666667 -2.95833333 -0.04166667 -2.41666667

## post: control - B post: A - B pre: control - A pre: control - B

## -2.54166667 -0.12500000 -0.75000000 0.08333333

## pre: A - B

## 0.83333333

**3. Run the bootstrap**

Now let’s wrap this all in a function that accepts the fitted model as an argument:

treatment\_phase\_contrasts <- function(mod){

rg <- ref\_grid(mod, mult.levs = rm\_levels)

# get the expected means:

em\_ <- emmeans(rg, ~ phase \* treatment)

# run pairwise tests between the treatment groups within each phase

c\_ <- contrast(em\_, "pairwise", by = 'phase')

# extract the estimates

est\_names <- c("fup: control - A", "fup: control - B", "fup: A - B",

"post: control - A", "post: control - B", "post: A - B",

"pre: control - A", "pre: control - B", "pre: A - B")

est\_values <- summary(c\_)$estimate

names(est\_values) <- est\_names

est\_values

}

# test it

treatment\_phase\_contrasts(fit\_mixed)

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

## fup: control - A fup: control - B fup: A - B post: control - A

## -2.91666667 -2.95833333 -0.04166667 -2.41666667

## post: control - B post: A - B pre: control - A pre: control - B

## -2.54166667 -0.12500000 -0.75000000 0.08333333

## pre: A - B

## 0.83333333

Finally, we will use lme4::bootMer to get the bootstrapped estimates!

treatment\_phase\_results <-

bootMer(fit\_mixed, treatment\_phase\_contrasts, nsim = 50) # R = 599 at least

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

summary(treatment\_phase\_results) # original vs. bootstrapped estimate (bootMed)

##

## Number of bootstrap replications R = 50

## original bootBias bootSE bootMed

## fup: control - A -2.916667 0.017263 0.77841 -2.801902

## fup: control - B -2.958333 -0.017880 0.86119 -3.025705

## fup: A - B -0.041667 -0.035143 0.98850 -0.066474

## post: control - A -2.416667 0.031072 0.82654 -2.383370

## post: control - B -2.541667 -0.024860 0.82351 -2.520263

## post: A - B -0.125000 -0.055932 1.03670 -0.216929

## pre: control - A -0.750000 -0.065397 0.73276 -0.851533

## pre: control - B 0.083333 0.024664 0.78592 0.111930

## pre: A - B 0.833333 0.090061 0.95015 0.994195

confint(treatment\_phase\_results, type = "perc") # does include zero?

## 2.5 % 97.5 %

## fup: control - A -5.062951 -1.2782764

## fup: control - B -4.985715 -1.0325666

## fup: A - B -2.348035 2.1660820

## post: control - A -4.451445 -0.5162071

## post: control - B -4.840519 -1.1705024

## post: A - B -2.349137 2.3025369

## pre: control - A -2.427992 0.8830127

## pre: control - B -1.915388 1.7159931

## pre: A - B -1.530049 2.7527436

Results indicate that the Control group is lower than both treatment groups in the post and fup (follow -up) phases.

If we wanted p-values, we could use this little function:

boot\_pvalues <- function(x, side = c(0, -1, 1)) {

# Based on:

# <https://blogs.sas.com/content/iml/2011/11/02/how-to-compute-p-values-for-a-bootstrap-distribution.html>

side <- side[1]

x <- as.data.frame(x$t)

ps <- sapply(x, function(.x) {

s <- na.omit(.x)

s0 <- 0

N <- length(s)

if (side == 0) {

min((1 + sum(s >= s0)) / (N + 1),

(1 + sum(s <= s0)) / (N + 1)) \* 2

} else if (side < 0) {

(1 + sum(s <= s0)) / (N + 1)

} else if (side > 0) {

(1 + sum(s >= s0)) / (N + 1)

}

})

setNames(ps,colnames(x))

}

boot\_pvalues(treatment\_phase\_results)

## fup: control - A fup: control - B fup: A - B post: control - A

## 0.03921569 0.03921569 0.94117647 0.03921569

## post: control - B post: A - B pre: control - A pre: control - B

## 0.03921569 0.74509804 0.23529412 0.94117647

## pre: A - B

## 0.27450980

These p-values can then be passed to p.adjust() for the p-value adjustment method of your choosing.

**Summary**

I’ve demonstrated (again!) how to run permutation tests on main effects / interactions, with follow-up analysis using the bootstrap method. Using this code as a basis for any analysis you might have in mind gives you all the flexibility of emmeans, which supports many (many) models!

