Workshop part 3.0

Intro to Linear Modeling in R

Lm() review: We have already been introduced to lm(), which allows a fairly straightforward analysis to be set up. Remember the basics of the modeling syntax:

y ~ x

Where y is the dependent variable and x is the independent variable (either continuous or a factor).

Key syntax is needed for interactions:

Symbol	Example	Meaning
+	+X	include this variable
−	−X	delete this variable
:	X:Z	include the interaction between these variables
*	X*Y	include these variables and the interactions between them
|	X | Z	conditioning: include x given z
^	(X + Z + W)^3	include these variables and all interactions up to three way
I	I(X*Z)	as is: include a new variable consisting of these variables multiplied
1	X − 1	intercept: delete the intercept (regress through the origin)

I.e.

```
mod.1 = lm(y ~ x1*x2*x3)  # all main factors, two way and three way interactions

mod.2 = lm(y ~ x1*x2*x3 – x2:x3)  # same but without that particular two way interaction
```

Alternative forms of lm: We have been using lm() by specifying the dataframe at the same time as the variable, e.g.

```
mod.1 = lm(duncan$prestige ~ duncan$type)
```

But an alternative form that is very common is:

```
mod.2 = lm(prestige ~ type, data = duncan)
```

In that case you can call the variable names without specifying the dataset until afterward.

Use of aov() : Aov() works like lm() and uses the same syntax for specifying a model, mostly; but it outputs an F table that is more recognizable for psychologists.

```
mod.3 = aov(prestige ~ type, data = duncan)
```

```
summary(mod.3)
```

Rather than providing tests of the coefficients etc. as summary() does for lm() output, the aov() output is the F table. The command anova (little a) does the same summary for output of the lm() command:

```
mod.2 = lm(prestige ~ type, data = duncan)

anova(mod.2)
```

Anova() vs anova() : Anova() (the capital A is important!) is a function from the car() library which can calculate Type III sums of squares. By default it returns Type II. But if we use contr.sum, contr.helmert, or contr.poly to set the contrasts on the factors, it will be able to return the Type III SS. (This doesn't work on continuous variables, only grouping factors.)

For example:

```
mod.5 = lm(stress$stressreduction ~ stress$treatment*stress$gender)
```

and

```
mod.6 = lm(stressreduction ~ treatment*gender, data=stress,
  contrasts=list(treatment=contr.sum, gender=contr.sum))
```

Are very similar in the overall pattern of significant effects, but their presentation is not the same.

(Basic background on Type I, II, and III sums of squares: Type I sum of squares are "sequential." In essence the factors are tested in the order they are listed in the model. Type III are "partial." In essence, every term in the model is tested in light of every other term in the model. That means that main effects are tested in light of interaction terms as well as in light of other main effects. Type II are similar to Type III, except that they preserve the principle of marginality. This means that main factors are tested in light of one another, but not in light of the interaction term. When data are balanced and the design is simple, types I, II, and III will give the same results. –from: *An R Companion for the Handbook of Biological Statistics*, by Salvatore S. Mangiafico (https://rcompanion.org/rcompanion/d_04.html)).

Repeated measure or within subjects factors: So far, we've only worked with between-subjects data. Quite commonly in psychology the same subject is assessed under all the conditions of a factor in the experiment (a within-subjects factor).

There are a number of ways to do analyses of data from within-subjects designs in R, ranging from the most simple to full-out generalized additive mixed models (GAMMs). We are going to keep it simple here!

The aov() function will allow within-subjects analyses, as will lme() from the nlme library, as will lmer() from the lme4 library. The lme4 library is probably the most common approach to random factors in a model, and it generalizes nicely into the more complex approaches.

I am working with the aov() examples from this tutorial: https://personality-project.org/r/r.guide/r.anova.html

1. Repeated measures within subjects: aov(y ~ x + Error(subj/x)) if using the aov() formulation

    Using the lmer() formulation: lmer(y~ x + (1|subj))

Note that Anova() by default uses the chi-square which makes comparing to the F output of aov() difficult. Use the test.statistic = "F" option in Anova() to get the F table.

2. Two-way fully repeated measures within subjects: aov(y ~ x1*x2 + (Error(subj/(x1*x2)))

    Using the lmer() formulation: lmer(y ~ x1*x2 +(1|subj) + (1|x1) + (1|x2))

3. Mixed model, with x1 and x2 as between and x2 and x3 as within subjects:

    aov(y ~ x1*x2*x3*x4 + Error(subj/x3*x4) +(x1*x2))

    That last x1*x2 term is needed because x3 and x4 are categorized by those non-nested factors.

    Using the lmer() formulation: lmer(y ~x1*x2*x3*x4 +(1|subj) + (1|x1) + (1|x2))

For a comparison of aov, lme and lmer across a range of common within-subjects designs, see http://dwoll.de/rexrepos/posts/anovaMixed.html. Note that the different techniques do not always give the same results, largely due to differences in their default algorithms and choices of sums of squares.