Two Way ANOVA
Introduction to Statistics Using R (Psychology 9041B)

Paul Gribble

Winter, 2016

1 Two-way between subjects ANOVA

In factorial between-subjects designs, subjects are assigned to groups that represent a factorial combination of the levels of more than one factor. Let's assume we have two factors. The first factor, we will call **Biofeedback**, has two levels: *present, absent*, and the second factor, **Drug**, also has two levels, *present, absent*. Twenty subjects are randomly assigned to one of four groups, and the dependent measure is blood pressure.

```
> bloodpressure <- c(158,163,173,178,168,188,183,198,178,193,
+         186,191,196,181,176,185,190,195,200,180)
> biofeedback <- factor(c(rep("present",10),rep("absent",10)))
> drug <- factor(rep(c(rep("present",5),rep("absent",5)),2))
> bpdata <- data.frame(bloodpressure, biofeedback, drug)
> print(bpdata)
```

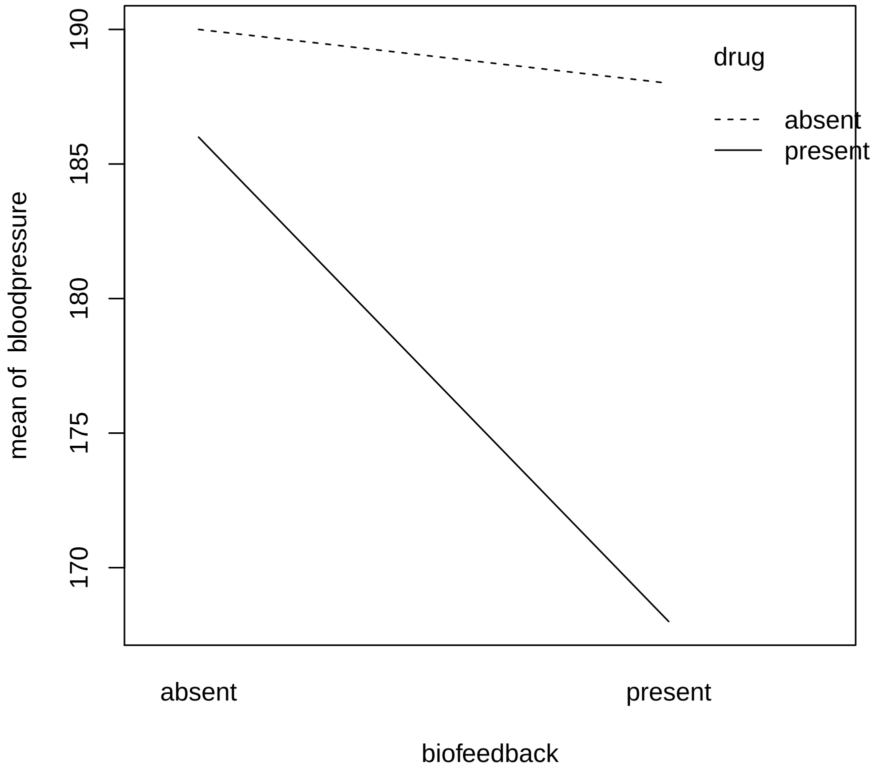
```
   bloodpressure biofeedback    drug
1            158     present present
2            163     present present
3            173     present present
4            178     present present
5            168     present present
6            188     present  absent
7            183     present  absent
8            198     present  absent
9            178     present  absent
10           193     present  absent
11           186      absent present
12           191      absent present
13           196      absent present
14           181      absent present
15           176      absent present
16           185      absent  absent
17           190      absent  absent
```

```
18            195      absent   absent
19            200      absent   absent
20            180      absent   absent
```

```
> summary(bpdata)
```

```
 bloodpressure    biofeedback      drug
 Min.   :158.0   absent :10   absent :10
 1st Qu.:177.5   present:10   present:10
 Median :184.0
 Mean   :183.0
 3rd Qu.:191.5
 Max.   :200.0
```

```
> interaction.plot(biofeedback, drug, bloodpressure)
```



The two-way ANOVA tests three omnibus effects: the main effect of each factor, and the interaction effect between the two factors. The full model is therefore that each observation

${Y_{ijk}}^{1}$ is made up of a grand mean μ, the main effect of biofeedback (α_j), the main effect of drug (β_k) and the interaction effect of blood pressure by drug ($\alpha\beta_{jk}$), plus unaccounted for variance (ϵ_{ijk}):

$$Y_{ijk} = \mu + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \epsilon_{ijk} \tag{1}$$

Since there are three omnibus tests, there are three restricted models — one for each effect. The full model is alway the same:

$$\begin{aligned}
full: \quad Y_{ijk} &= \mu + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \epsilon_{ijk} && (2)\\
biofeedback: \quad Y_{ijk} &= \mu + \beta_k + (\alpha\beta)_{jk} + \epsilon_{ijk} && (3)\\
drug: \quad Y_{ijk} &= \mu + \alpha_j + (\alpha\beta)_{jk} + \epsilon_{ijk} && (4)\\
(biofeedback)(drug): \quad Y_{ijk} &= \mu + \alpha_j + \beta_k + \epsilon_{ijk} && (5)
\end{aligned}$$

Note how each effect is tested using a restricted model that omits the parameter associated with that effect. Another way of saying this: If there is no main effect of biofeedback, then the α parameter should be zero. If this is true, the restricted model (that doesn't contain α), should not fit the data significantly worse than the full model that includes α.

Main effects

Each test of a main effect of a factor is a test of whether there is an effect of that factor on the dependent variable when the data are averaged over the levels of the other factor. Is there an effect of biofeedback, when we average over (essentially ignoring) the levels of the drug factor?

Interaction Effect

The interaction effect is a test of whether the effect of one factor (e.g. biofeedback) on the dependent variable is *different* depending on the level of the other factor (e.g. drug). Another way of saying this, is that if the effects of the two factors (biofeedback and drug) are simply additive, then there will be no interaction effect. So an interaction effect, if present, reflects the extent to which there is a non-linear (multiplicative) effect of the two factors together.

This can be seen in the example data above. The effect of biofeedback alone is to reduce blood pressure from 190 (when neither biofeedback nor drug are present), to 188. This reflects a decrease of 2 units of blood pressure. The effect of drug alone is to reduce blood pressure from 190 to 186 (a reduction of 4 units). If there was no interaction effect, then when both biofeedback and drug are present their effects should simply sum, and reduce blood pressure by 6 units (to 194). In fact, when both a present, blood pressure is reduced to 168, a much larger reduction.

[1] the ith subject in the jth level of α and the kth level of β

Workflow

I recommend that the first effect you should examine (even though it usually appears last in the list of effects in an ANOVA output table), is the interaction effect. If it is significant, then you should probably ignore the two main effects tests. If the interaction effect is significant, this means that the effect of one factor on the dependent variable is different depending on the level of the other factor. Thus why would you average over the levels of the second factor, knowing that the levels of the second factor make a difference, i.e. that they in fact modulate the effect of the first factor? There may be a situation where it makes sense to do so, but it's a bit difficult to imagine.

An example in R

We use the `aov()` command to analyse the data, and the `summary()` command to output the results in a familiar looking ANOVA table format.

Note that we issue a strange command `options()` with some strange looking arguments. This is important as it instructs R to compute effects assuming that contrast weights sum to zero. If we don't tell R to do this, it will do something else, and we will not get results that we expect. See `http://goo.gl/3GHTB` for some more details about this and how it relates to the different ways of computing sums-of-squares as discussed in the Maxwell & Delaney text, chapter 7.

```
> options(contrasts = c("contr.sum", "contr.poly"))
> myanova <- aov(bloodpressure ~ biofeedback*drug)
> summary(myanova)

                 Df Sum Sq Mean Sq F value  Pr(>F)
biofeedback       1    500   500.0    8.00 0.01211 *
drug              1    720   720.0   11.52 0.00371 **
biofeedback:drug  1    320   320.0    5.12 0.03792 *
Residuals        16   1000    62.5
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We see that all three omnibus tests are significant, there is a main effect of biofeedback, there is a main effect of drug, and there is an interaction between biofeedback and drug. Since there is a significant interaction effect, the next step is to conduct followup tests to investigate where the differences lie. There are many possibilities for how to proceed. You could perform so-called *simple effects* analyses, or you could proceed directly to pairwise tests. I typically go straight to the pairwise tests. There are also many ways to control for Type-I error. I typically use Tukey tests. Fortunately there is a simple command in `R` to do this:

```
> TukeyHSD(myanova,which="biofeedback:drug")

  Tukey multiple comparisons of means
    95% family-wise confidence level
```

```
Fit: aov(formula = bloodpressure ~ biofeedback * drug)

$`biofeedback:drug`
                               diff      lwr       upr     p adj
present:absent-absent:absent     -2 -16.3051 12.305099 0.9775889
absent:present-absent:absent     -4 -18.3051 10.305099 0.8534038
present:present-absent:absent   -22 -36.3051 -7.694901 0.0022719
absent:present-present:absent    -2 -16.3051 12.305099 0.9775889
present:present-present:absent  -20 -34.3051 -5.694901 0.0051230
present:present-absent:present  -18 -32.3051 -3.694901 0.0115535
```

The second parameter to the `TukeyHSD()` function instructs it which omnibus test we are interested in examining for the pairwise tests.

Statistical Power

The `power.anova.test()` function we saw earlier in the course only applies to balanced, single-factor designs. There is a package called `pwr` that contains functions we can use to do power analysis for multi-factor designs. To install it, do the following: `install.packages("pwr")` once, to download the package to your computer. Then in a given `R` session, to load in the library, type `library(pwr)`.

To do power calculations we will use the `pwr.f2.test()` function, which enables us to compute the power for *any* F test. Let's say we have a two-factor design, with factors A (three levels) and B (two levels). Let's say that we expect the effect size for the main effect of A to be medium (0.25 according to Cohen, 1988). Let's say we have 5 subjects in each group — the df for our mean-squared term in our ANOVA will be $3 \times 2 \times (5-1) = 24$. Here is how we would use `pwr.f2.test()` to compute the power of the test for a main effect of factor A, for an α level of 0.05:

```
> library(pwr)
> pwr.f2.test(u=2, v=24, f2=(0.25*0.25), sig.level=0.05, power=NULL)

     Multiple regression power calculation

              u = 2
              v = 24
             f2 = 0.0625
      sig.level = 0.05
          power = 0.1775164
```

The `u` and `v` parameters in the `pwr.f2.test()` function correspond to the df-numerator and df-denominator terms in the ANOVA. The `f2` parameter is the effect size squared (Cohen, 1988).

To ask how many subjects we would require to have a significant main effect of A, at a power of 0.90 we would issue this command:

```
> pwr.f2.test(u=2, v=NULL, f2=(0.25*0.25), sig.level=0.05, power=0.90)

     Multiple regression power calculation

              u = 2
              v = 202.4912
             f2 = 0.0625
      sig.level = 0.05
          power = 0.9
```

We see that `v = 202.4912` which means that we would need 203 degrees of freedom in the denominator of our F test. This means we would need $1 + [203/(3 \times 2)] = 35$ subjects per group in our 3 x 2 design. Remember,

$$df_{residuals} = a \times b \times (n - 1) \tag{6}$$

where a is the number of levels of factor A, b is the number of levels of factor B, and n is the number of subjects in each group.

Complications

When your experiment has equal number of subjects in each group, it is known as a *balanced design*. An *unbalanced design* has unequal numbers of subjects in each group. This turns out to be a big nuisance when it comes to the computations underlying the ANOVA. You should read the chapter in Maxwell & Delaney for the details, but essentially the problem is that when you have unequal numbers of subjects in each group, the main effects and interaction effects are no longer orthogonal to each other. This means that effects of one can erroneously show up as effects of another (if you don't do something to correct for the situation).

The issue of unbalanced designs, and the resulting effects on calculations of the ANOVA, require that you understand something about different ways of computing so-called *sums of squares*. Remember that we encountered the idea of sums of squares in our earlier discussions of ways of computing the error associated with full versus restricted models. Part of the traditional ANOVA calculations involve computing the sums of squares for each omnibus test in an experimental design (the SS usually appears in the first column of numbers in a traditional ANOVA output table).

There are three common ways that sums of squares are computed, and they are known as Type-I, Type-II and Type-III sums of squares. You should read Maxwell and Delaney for a detailed discussion of the ways that these differ ... but there are three things you need to know, as a bottom line. First, the recommended sums of squares are the Type-III sums of squares. Second, the way that R computes sums of squares by default, is Type-I sums of squares. Third, Type-I and Type-III sums of squares for omnibus effects (main effects and interactions) will only differ when you have an unbalanced design. We won't go into the details of this here, please read the text.

When your design is balanced (same number of subjects in each cell of your experimental design), then you don't have to worry, just proceed as usual. When however you have an unbalanced design, you have to do something special in R to force it to compute the ANOVA using Type-III sums of squares. There are **two ways** of doing this.

The first way is to use the drop1() command following the usual aov() function:

```
> myanova <- aov(bloodpressure ~ biofeedback*drug)
> drop1(myanova, . ~ ., test="F")

Single term deletions

Model:
bloodpressure ~ biofeedback * drug
                 Df Sum of Sq  RSS    AIC F value   Pr(>F)
<none>                        1000 86.240
biofeedback       1       500 1500 92.350    8.00 0.012109 *
drug              1       720 1720 95.087   11.52 0.003706 **
biofeedback:drug  1       320 1320 89.793    5.12 0.037917 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The results are the same as before because this design is balanced ... this is just a demo of how to do it IF the design was unbalanced. The drop1() command performs F-tests of the full model versus restricted models, where the restricted models are constructed by removing each term in the full model, one at a time. By default, R doesn't compute the ANOVA this way, but computes the Type-I sums of squares, which involves incrementally *adding* terms to the restricted model. When the design is balanced, they are both valid and produce the same results. When the design is unbalanced, the Type-I method produces something strange and not useful (see Maxwell & Delaney for an explanation of why Type-I sums of squares are strange and not useful).

Important: R will not warn you that you have an unbalanced design — it will go ahead and compute the Type-I sums of squares and report the results to you. It's your responsibility to know what procedure you're applying to your data, and to ensure that it's the correct one.

The second method is to use the Anova() function in the car package. First install the package on your computer — in R, type: install.packages("car"). Then:

```
> library(car)
> Anova(myanova, type="III")

Anova Table (Type III tests)

Response: bloodpressure
                 Sum Sq Df  F value    Pr(>F)
(Intercept)      669780  1 10716.48 < 2.2e-16 ***
biofeedback         500  1     8.00  0.012109 *
drug                720  1    11.52  0.003706 **
biofeedback:drug    320  1     5.12  0.037917 *
```

```
Residuals          1000 16
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We will be using the `Anova()` function later in the course for repeated-measures designs, so you may as well install it now.

Again, please read the chapter in Maxwell & Delaney for the full details about the difference between Type-I, Type-II and Type-III sums of squares and how they relate to unbalanced designs.