12.6: Multiple Comparisons and Post Hoc Tests

Any time you run an ANOVA with more than two groups, and you end up with a significant effect, the first thing you'll probably want to ask is which groups are actually different from one another. In our drugs example, our null hypothesis was that all three drugs (placebo, Anxifree and Joyzepam) have the exact same effect on mood. But if you think about it, the null hypothesis is actually claiming *three* different things all at once here. Specifically, it claims that:

- Your competitor's drug (Anxifree) is no better than a placebo (i.e., $\mu_A=\mu_P$)
- Your drug (Joyzepam) is no better than a placebo (i.e., $\mu_J=\mu_P$)
- Anxifree and Joyzepam are equally effective (i.e., $\mu_J=\mu_A$)

If any one of those three claims is false, then the null hypothesis is also false. So, now that we've rejected our null hypothesis, we're thinking that *at least* one of those things isn't true. But which ones? All three of these propositions are of interest: you certainly want to know if your new drug Joyzepam is better than a placebo, and it would be nice to know how well it stacks up against an existing commercial alternative (i.e., Anxifree). It would even be useful to check the performance of Anxifree against the placebo: even if Anxifree has already been extensively tested against placebos by other researchers, it can still be very useful to check that your study is producing similar results to earlier work.

When we characterise the null hypothesis in terms of these three distinct propositions, it becomes clear that there are eight possible "states of the world" that we need to distinguish between:

possibility:	is $\mu_P=\mu_A$?	is $\mu_P=\mu_J$?	is $\mu_A=\mu_J$?	which hypothesis?
1	✓	✓	✓	null
2	✓	✓		alternative
3	✓		✓	alternative
4	✓			alternative
5		✓	✓	alternative
6		✓		alternative
7			✓	alternative
8				alternative

By rejecting the null hypothesis, we've decided that we *don't* believe that #1 is the true state of the world. The next question to ask is, which of the other seven possibilities *do* we think is right? When faced with this situation, its usually helps to look at the data. For instance, if we look at the plots in Figure 14.1, it's tempting to conclude that Joyzepam is better than the placebo and better than Anxifree, but there's no real difference between Anxifree and the placebo. However, if we want to get a clearer answer about this, it might help to run some tests.

12.6.1 Running "pairwise" t-tests

How might we go about solving our problem? Given that we've got three separate pairs of means (placebo versus Anxifree, placebo versus Joyzepam, and Anxifree versus Joyzepam) to compare, what we could do is run three separate t-tests and see what happens. There's a couple of ways that we could do this. One method would be to construct new variables corresponding the groups you want to compare (e.g., `anxifree` , `placebo` and `joyzepam`), and then run a t-test on these new variables:

```
t.test( anxifree, placebo, var.equal = TRUE )   # Student t-test

anxifree <- with(clin.trial, mood.gain[drug == "anxifree"])  # mood change due to anx
placebo <- with(clin.trial, mood.gain[drug == "placebo"])    # mood change due to pla
```

or, you could use the `subset` argument in the `t.test()` function to select only those observations corresponding to one of the two groups we're interested in:

```
t.test( formula = mood.gain ~ drug,
        data = clin.trial,
        subset = drug %in% c("placebo","anxifree"),
        var.equal = TRUE
)
```

See Chapter 7 if you've forgotten how the `%in%` operator works. Regardless of which version we do, R will print out the results of the t-test, though I haven't included that output here. If we go on to do this for all possible pairs of variables, we can look to see which (if any) pairs of groups are significantly different to each other. This "lots of t-tests idea" isn't a bad strategy, though as we'll see later on there are some problems with it. However, for the moment our bigger problem is that it's a *pain* to have to type in such a long command over and over again: for instance, if your experiment has 10 groups, then you have to run 45 t-tests. That's way too much typing.

To help keep the typing to a minimum, R provides a function called `pairwise.t.test()` that automatically runs all of the t-tests for you. There are three arguments that you need to specify, the outcome variable `x`, the group variable `g`, and the `p.adjust.method` argument, which "adjusts" the p-value in one way or another. I'll explain p-value adjustment in a moment, but for now we can just set `p.adjust.method = "none"` since we're not doing any adjustments. For our example, here's what we do:

```
pairwise.t.test( x = clin.trial$mood.gain,   # outcome variable
                 g = clin.trial$drug,         # grouping variable
                 p.adjust.method = "none"     # which correction to use?
 )
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  clin.trial$mood.gain and clin.trial$drug
##
##          placebo anxifree
## anxifree 0.15021 -
## joyzepam 3e-05   0.00056
##
## P value adjustment method: none
```

One thing that bugs me slightly about the `pairwise.t.test()` function is that you can't just give it an `aov` object, and have it produce this output. After all, I went to all that trouble earlier of getting R to create the `my.anova` variable and – as we saw in Section 14.3.2 – R has actually stored enough information inside it that I should just be able to get it to run all the pairwise tests using `my.anova` as an input. To that end, I've included a `posthocPairwiseT()` function in the `lsr` package that lets you do this. The idea behind this function is that you can just input the `aov` object itself,[208] and then get the pairwise tests as an output. As of the current writing, `posthocPairwiseT()` is actually just a simple way of calling `pairwise.t.test()` function, but you should be aware that I intend to make some changes to it later on. Here's an example:

```
posthocPairwiseT( x = my.anova, p.adjust.method = "none" )
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mood.gain and drug
##
##          placebo anxifree
## anxifree 0.15021 -
## joyzepam 3e-05   0.00056
##
## P value adjustment method: none
```

In later versions, I plan to add more functionality (e.g., adjusted confidence intervals), but for now I think it's at least kind of useful. To see why, let's suppose you've run your ANOVA and stored the results in `my.anova`, and you're happy using the Holm correction (the default method in `pairwise.t.test()`, which I'll explain this in a moment). In that case, all you have to do is type this:

```
posthocPairwiseT( my.anova )
```

and R will output the test results. Much more convenient, I think.

12.6.2 Corrections for multiple testing

In the previous section I hinted that there's a problem with just running lots and lots of t-tests. The concern is that when running these analyses, what we're doing is going on a "fishing expedition": we're running lots and lots of tests without much theoretical guidance, in the hope that some of them come up significant. This kind of theory-free search for group differences is referred to as ***post hoc analysis*** ("post hoc" being Latin for "after this").[209]

It's okay to run post hoc analyses, but a lot of care is required. For instance, the analysis that I ran in the previous section is actually pretty dangerous: each *individual* t-test is designed to have a 5% Type I error rate (i.e., α=.05), and I ran three of these tests. Imagine what would have happened if my ANOVA involved 10 different groups, and I had decided to run 45 "post hoc" t-tests to try to find out which ones were significantly different from each other, you'd expect 2 or 3 of them to come up significant *by chance alone*. As we saw in Chapter 11, the central organising principle behind null hypothesis testing is that we seek to control our Type I error rate, but now that I'm running lots of t-tests at once, in order to determine the source of my ANOVA results, my actual Type I error rate across this whole *family* of tests has gotten completely out of control.

The usual solution to this problem is to introduce an adjustment to the p-value, which aims to control the total error rate across the family of tests (see Shaffer 1995). An adjustment of this form, which is usually (but not always) applied because one is doing post hoc analysis, is often referred to as a ***correction for multiple comparisons***, though it is sometimes referred to as "simultaneous inference". In any case, there are quite a few different ways of doing this adjustment. I'll discuss a few of them in this section and in Section 16.8, but you should be aware that there are many other methods out there (see, e.g., Hsu 1996).

12.6.3 Bonferroni corrections

The simplest of these adjustments is called the ***Bonferroni correction*** (Dunn 1961), and it's very very simple indeed. Suppose that my post hoc analysis consists of m separate tests, and I want to ensure that the total probability of making *any* Type I errors at all is at most α.[210] If so, then the Bonferroni correction just says "multiply all your raw p-values by m". If we let p denote the original p-value, and let p'_j be the corrected value, then the Bonferroni correction tells that:

$$p' = m \times p$$

And therefore, if you're using the Bonferroni correction, you would reject the null hypothesis if $p' < \alpha$. The logic behind this correction is very straightforward. We're doing m different tests; so if we arrange it so that each test has a Type I error rate of at most α/m, then the *total* Type I error rate across these tests cannot be larger than α. That's pretty simple, so much so that in the original paper, the author writes:

The method given here is so simple and so general that I am sure it must have been used before this. I do not find it, however, so can only conclude that perhaps its very simplicity has kept statisticians from realizing that it is a very good method in some situations (pp 52-53 Dunn 1961)

To use the Bonferroni correction in R, you can use the `pairwise.t.test()` function,[211] making sure that you set `p.adjust.method = "bonferroni"` . Alternatively, since the whole reason why we're doing these pairwise tests in the first place is because we have an ANOVA that we're trying to understand, it's probably more convenient to use the `posthocPairwiseT()` function in the `lsr` package, since we can use `my.anova` as the input:

```
posthocPairwiseT( my.anova, p.adjust.method = "bonferroni")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mood.gain and drug
##
##          placebo anxifree
## anxifree 0.4506  -
## joyzepam 9.1e-05 0.0017
##
## P value adjustment method: bonferroni
```

If we compare these three p-values to those that we saw in the previous section when we made no adjustment at all, it is clear that the only thing that R has done is multiply them by 3.

12.6.4 Holm corrections

Although the Bonferroni correction is the simplest adjustment out there, it's not usually the best one to use. One method that is often used instead is the ***Holm correction*** (Holm 1979). The idea behind the Holm correction is to pretend that you're doing the tests sequentially; starting with the smallest (raw) p-value and moving onto the largest one. For the j-th largest of the p-values, the adjustment is *either*

$$p'_j = j \times p_j$$

(i.e., the biggest p-value remains unchanged, the second biggest p-value is doubled, the third biggest p-value is tripled, and so on), *or*

$$p'_j = p'_{j+1}$$

whichever one is *larger*. This might sound a little confusing, so let's go through it a little more slowly. Here's what the Holm correction does. First, you sort all of your p-values in order, from smallest to largest. For the smallest p-value all you do is multiply it by m, and you're done. However, for all the other ones it's a two-stage process. For instance, when you move to the second smallest p value, you first multiply it by m−1. If this produces a number that is bigger than the adjusted p-value that you got last time, then you keep it. But if it's smaller than the last one, then you copy the last p-value. To illustrate how this works, consider the table below, which shows the calculations of a Holm correction for a collection of five p-values:

raw p	rank j	p×j	Holm p
.001	5	.005	.005
.005	4	.020	.020
.019	3	.057	.057
.022	2	.044	.057
.103	1	.103	.103

Hopefully that makes things clear.

Although it's a little harder to calculate, the Holm correction has some very nice properties: it's more powerful than Bonferroni (i.e., it has a lower Type II error rate), but – counterintuitive as it might seem – it has the *same* Type I error rate. As a consequence, in practice there's never any reason to use the simpler Bonferroni correction, since it is always outperformed by the slightly more elaborate Holm correction. Because of this, the Holm correction is the default one used by `pairwise.t.test()` and `posthocPairwiseT()`. To run the Holm correction in R, you could specify `p.adjust.method = "Holm"` if you wanted to, but since it's the default you can just to do this:

```
posthocPairwiseT( my.anova )
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mood.gain and drug
##
##          placebo anxifree
## anxifree 0.1502  -
## joyzepam 9.1e-05 0.0011
##
## P value adjustment method: holm
```

As you can see, the biggest p-value (corresponding to the comparison between Anxifree and the placebo) is unaltered: at a value of .15, it is exactly the same as the value we got originally when we applied no correction at all. In contrast, the smallest p-value (Joyzepam versus placebo) has been multiplied by three.

12.6.5 Writing up the post hoc test

Finally, having run the post hoc analysis to determine which groups are significantly different to one another, you might write up the result like this:

Post hoc tests (using the Holm correction to adjust p) indicated that Joyzepam produced a significantly larger mood change than both Anxifree (p=.001) and the placebo ($p=9.1\times10^{-5}$). We found no evidence that Anxifree performed better than the placebo (p=.15).

Or, if you don't like the idea of reporting exact p-values, then you'd change those numbers to p<.01, p<.001 and p>.05 respectively. Either way, the key thing is that you indicate that you used Holm's correction to adjust the p-values. And of course, I'm assuming that elsewhere in the write up you've included the relevant descriptive statistics (i.e., the group means and standard deviations), since these p-values on their own aren't terribly informative.
