# 12.4b Appendix: Multiple Comparisons Using R

**by EV Nordheim, MK Clayton & BS Yandell, December 9, 2003**

Here we briefly indicate how R can be used to conduct multiple comparison after ANOVA. We illustrate the most frequently used methods, protected $T$-tests and the Bonferroni method, using the drug data.

```
> drug = read.table("http://www.stat.wisc.edu/~st571-1/data/drug.dat")
> group = factor(drug$V1)
> y = drug$V2
```

Recall that column `V1` contains the treatment group number and column `V2` contains the drug response.

Suppose we have already run an ANOVA on these data (see the Appendix for Chapter 11). Since the results were highly significant (p-value = 0.00023), we can now conduct the protected $T$-tests using the command `pairwise.t.test`. The third argument (`"none"`) specifies that no adjustment is made to p-values.

```
> pairwise.t.test(y, group, "none")

        Pairwise comparisons using t tests with pooled SD

data:  y and group

  1       2      3      4
2 0.1158  -      -      -
3 4.8e-05 0.0025 -      -
4 0.0499  0.6186 0.0124 -
5 0.0003  0.0133 0.5622 0.0509

P value adjustment method: none
```

This output is rather condensed, showing the lower half of a table of p-values comparing means. For instance the p-value for $T$-test of treatment 1 against treatment 2 is 0.116. In order to turn this into a useful display, it is helpful to first order the treatments by their means and then successively identify which pairs are not significantly different using the above table. While it is possible to automate this, it is beyond the scope of this course and there is currently no command to do that. Here is the resulting comparison table for the drug treatments, which was compiled by hand:

```
      mean  LSD
3 5.600000 a
5 5.725000 ab
4 6.185714  bc
2 6.300000   c
1 6.683333    d
```

The Bonferroni method can be applied in a similar manner using `"bonferroni"` (do not capitalize it!) as the third argument:

```
> pairwise.t.test(y, group, "bonferroni")

        Pairwise comparisons using t tests with pooled SD

data:  y and group

  1       2       3       4
2 1.00000 -       -       -
3 0.00048 0.02464 -       -
4 0.49895 1.00000 0.12413 -
5 0.00302 0.13279 1.00000 0.50930

P value adjustment method: bonferroni
```

The corresponding summary compiled by hand is

```
      mean BON
3 5.600000 a
5 5.725000 ab
4 6.185714 abc
2 6.300000  bc
1 6.683333   c
```

For those adventurous, here is some R code to help create such summary tables. The home-made command `p.table` below has the group names, the means in ascending order, and a table of 0s (no significant difference at specified `level`) and 1s (significant difference). Comments begin with `#`.

```
> p.table = function(x, g, p.adjust.method = "none", ..., level = 0.05) {
+     ## fill out p-value table
+     p = pairwise.t.test(x, g, p.adjust.method, ...)$p.value
+     p[is.na(p)] = 0
+     p = rbind(0, cbind(p, 0))
+     p = p + t(p)
+     ## 0 = no difference, 1 = difference
+     p = 1 * (p <= level)
+     diag(p) = 0
+     ## get means and find their order
+     m = tapply(x, g, mean)
+     o = order(m)
+     p = p[o, o]
+     dimnames(p) = list(names(m[o]), names(m[o]))
+     cbind(mean = m[o], p)
+ }
```

Now we run this command on the drug data using options for protected $T$-tests and Bonferroni comparisons. Note there is now only a modest amount of work to turn this into a small table with `"abc"` annotation.

```
> p.table(y, group, "none", 0.05)

      mean 3 5 4 2 1
3 5.600000 0 0 1 1 1
5 5.725000 0 0 0 1 1
4 6.185714 1 0 0 0 1
2 6.300000 1 1 0 0 0
1 6.683333 1 1 1 0 0
```

```
> p.table(y, group, "bonferroni", 0.05)

      mean 3 5 4 2 1
3 5.600000 0 0 0 1 1
5 5.725000 0 0 0 0 1
4 6.185714 0 0 0 0 0
2 6.300000 1 0 0 0 0
1 6.683333 1 1 0 0 0
```