STAT 506, Spring 2017: Homework 2

- Consider the fruit fly data of Problem 3.2. Test

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4 \text{ and } \tfrac{1}{4}(\mu_1 + \mu_2 + \mu_3 + \mu_4) - \mu_5 = 0,$$

  at the 5% level using a joint F-test, protecting the experimentwise error rate *if we weren't data snooping*. Note that this can be written as 4 contrasts and placed into a matrix as in the first R example in the Chapter 4 notes; the matrix will have 4 columns of length 5. The first three contrasts are $\mu_1 - \mu_2 = 0$, $\mu_2 - \mu_3 = 0$, and $\mu_3 - \mu_4 = 0$; the last one is $\frac{1}{4}(\mu_1 + \mu_2 + \mu_3 + \mu_4) - \mu_5$. Now repeat this using Scheffé CI's, which protects the SFER, even after data snooping.

- Pr. 5.1. Do your conclusions change from the previous problem?

- Ex 4.2. Also test that the contrast is zero at the 5% level.

- Ex. 4.3. This problem is a bit tricky. The experimental unit is *substrate* and the measurement unit is *pin*, so it's best to summarize the four pin strengths with one overall summary (pp. 8–9 in the textbook); the mean pin strength makes sense here. Use the following data for this problem instead of the data in the `oehlert` package. I'll show how I got the modified data set in R below.

```
strength=c(7.3550,7.5900,7.7750,6.2350,7.0000,6.7900,7.6250,5.7975,
 8.1050,8.3600,8.0100,8.2675)
oper=factor(c(1,1,1,2,2,2,3,3,3,4,4,4))
```

  `strength` is the mean pin strength across the four pins for each of the 12 randomly assigned substrates, and `oper` is the operator number.

- Ex. 4.4(a).

- Ex. 5.3. Hint: there are two "all pairwise comparisons" procedures that protect SFER. Summarize the results via the `lines` function. Use the data defined above for Ex. 4.3.

- Ex. 5.4.

- Ex. 5.5. Hint: you are only being asked to compare to control; Dunnett is best here, implemented in `compare.to.control`.

Only if you are interested..., here is R Voodoo to turn `pr03.1` data into something appropriate. Get the mean of the four pin strengths for each substrate assigned to an operator:

```
> attach(pr03.1)
> s=tapply(strength,list(oper,substrate),mean)
> s
      1      2      3
1 7.355 7.5900 7.7750
2 6.235 7.0000 6.7900
3 7.625 5.7975 8.1050
4 8.360 8.0100 8.2675
```

So operator 1 has data has strengths 7.355, 7.590, and 7.775 lbs., operator 2 6.235, 7.000, 6.790, etc. We can flatten the matrix into a data vector like this:

```
> strength=as.vector(t(s))
> strength
 [1] 7.3550 7.5900 7.7750 6.2350 7.0000 6.7900 7.6250 5.7975 8.1050 8.3600
[11] 8.0100 8.2675
```

The `t()` function converts a table to its "transpose" – it makes the rows into the columns and the columns into the rows. Now match up each strength with the correct operator:

```
> oper=factor(rep(1:4,each=3))
> oper
 [1] 1 1 1 2 2 2 3 3 3 4 4 4
Levels: 1 2 3 4

strength # as above
oper # as above
```