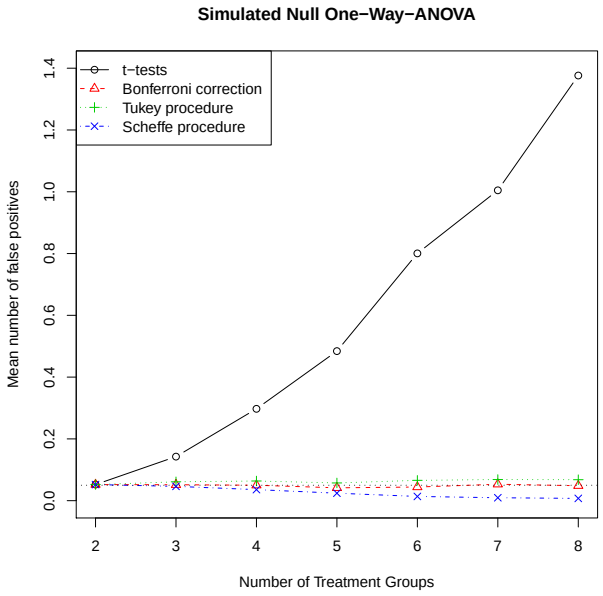
2/9/2010 36-402/608 ADA-II H. Seltman
Breakout #9 Comments

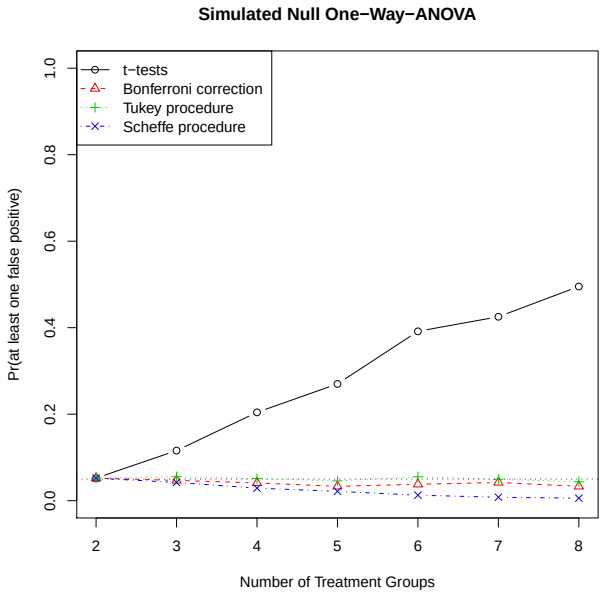
The results below are from simulation of experiments with one factor, which represents between I=2 and and I=8 different treatments to which subjects are randomized. For each experiment the null hypothesis, that $\mu_1 = \cdots = \mu_I$, is true. For each I, 1900 experiments are simulated, so the error for events that occur 5% of the time is +/-1%.

For each I, the number of possible paired tests is I(I-1)/2, which ranges from 1 to 28. For each experiment, all paired t-tests are run, and the number of positive (which must all be false positive) tests is recorded. Over all 1900 experiments the table below shows the mean number of positive tests per experiment and the mean of the indicator variable that at least one test was positive (FWAOP=fraction with at least one positive test). Similarly for each experiment the Bonferroni correction is used to control for multiple testing (based on looking at all pairs), the Tukey procedure is run for all pairs, and the Scheffe procedure is run (which is valid for all possible contrasts simultaneously).

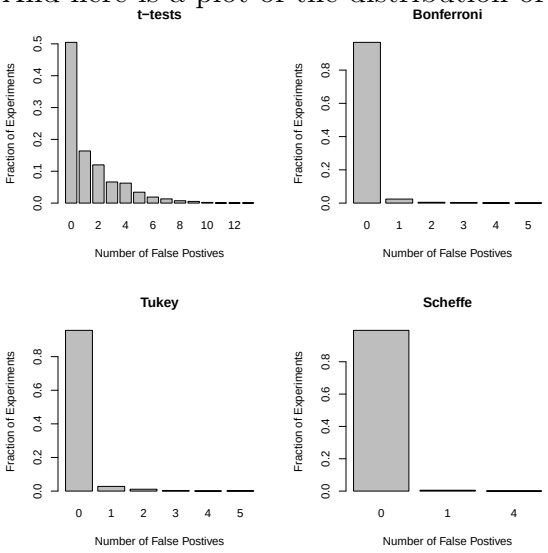
	---t.test---		-Bonferroni-		---Tukey---		---Scheffe--	
I	Count	FWAOP	Count	FWAOP	Count	FWAOP	Count	FWAOP
2	0.052	0.052	0.052	0.052	0.052	0.052	0.052	0.052
3	0.143	0.116	0.052	0.047	0.061	0.055	0.046	0.042
4	0.297	0.204	0.050	0.041	0.064	0.051	0.036	0.029
5	0.484	0.270	0.042	0.033	0.057	0.046	0.024	0.022
6	0.801	0.392	0.044	0.038	0.066	0.055	0.014	0.013
7	1.005	0.425	0.053	0.042	0.068	0.049	0.009	0.008
8	1.376	0.495	0.048	0.033	0.068	0.044	0.007	0.006

Here are plots of all of the counts and of the FWAOP values:





And here is a plot of the distribution of counts for I=8:



Question 1: What do the plots show?

On the right we see that as the number of treatment groups in an ANOVA rises, comparing all pairs with multiple uncorrected testing leads to a higher and higher chance of falsely claiming a difference when none really exists (false positive, or type-1 error). But the three correction procedures all maintain type-1 error. Subtly, the Scheffe procedure is too conservative for just testing all pairs, and thus will have lower power than necessary.

The plot on the left shows that, while the correction procedures have an average number of false positives of around 5%, the naive multiple testing procedure produces at least one false positive (per experiment) when the number of levels of the factor is at least 7. This is seen in more detail in the third plot: the correct procedures rarely give more than one false positive (per experiment), but the naive procedure often produces 2, 3 or more false positives when there are 8 treatments.

Example to demonstrate tests and code:

```
dtf = read.csv("Experiment.csv")
# treat     score
# A:8   Min.   : 7.905
# B:8   1st Qu.: 9.594
# C:8   Median :10.100
# D:8   Mean   :10.074
# E:8   3rd Qu.:10.538
# F:8   Max.   :12.225
e0 = aov(score~treat, dtf)
summary(e0)
#             Df Sum Sq Mean Sq F value    Pr(>F)
```

```
# treat        5 69.372 13.8744  6.4438 0.0001581 ***
# Residuals   42 90.431  2.1531
```

The standard procedure is to "focus" your power by used contrasts that are planned (i.e., chosen *before* looking at the data.

For this experiment, the (orthogonal) planned contrasts, based on the underlying science, are to compare ABC vs. DEF, AB vs. C, DE vs. F, A vs. B, and D vs. E.

```
library(gmodels) # for fit.contrast()
planC = rbind(ABCvsDEF = c(1/3,1/3,1/3,-1/3,-1/3,-1/3),
              ABvsC = c(1/2,1/2,-1,0,0,0), AvsB = c(1,-1,0,0,0,0),
              DEvsF = c(0,0,0,1/2,1/2,-1), DvsE = c(0,0,0,1,-1,0))
round( fit.contrast(e0, "treat", planC), 3)
#               Estimate Std. Error t value Pr(>|t|)
# treatABCvsDEF   -0.411      0.424  -0.971    0.337
# treatABvsC      -0.720      0.635  -1.134    0.263
# treatAvsB       -3.999      0.734  -5.450    0.000
# treatDEvsF       0.161      0.635   0.253    0.801
# treatDvsE        0.345      0.734   0.470    0.641
```

Question 2: What do we conclude? Why can't we do more testing the same way?

We see evidence of a significant difference in population means of score for treatment A vs. B (< 0.0005), but not significant differences for ABC vs DEF, AB vs C, DE vs F, or D vs E.

Note: if R calculated $p \geq 0.0005$ it would have rounded to 0.001.

We have 'used up" our 4 df on the four contrasts. Any additional testing cannot be a decomposition of the treatment SS, so it re-uses the same information and is not protected by the F value for the overall null hypothesis that all treatment means are equal (F=6.55, 0.00016).

The naive "all t-tests approach" starts with finding biggest mean difference, doing a t-test, and repeating until a non-significant one is found.

```
with(dtf, tapply(score, treat, mean))
#        A        B        C        D        E        F
# 10.04231 14.04113 12.76204 12.91898 12.57440 12.58586

with(dtf, t.test(score[treat=="A"], score[treat=="B"]))
# t = -7.0826, df = 13.972, p-value = 5.547e-06
with(dtf, t.test(score[treat=="A"], score[treat=="D"]))
```

```
# t = -3.7533, df = 11.791, p-value = 0.002837
with(dtf, t.test(score[treat=="A"], score[treat=="C"]))
# t = -3.5927, df = 11.912, p-value = 0.003736
with(dtf, t.test(score[treat=="A"], score[treat=="F"]))
# t = -3.9848, df = 13.549, p-value = 0.001440
with(dtf, t.test(score[treat=="A"], score[treat=="E"]))
# t = -4.0317, df = 13.67, p-value = 0.001294
with(dtf, t.test(score[treat=="B"], score[treat=="E"]))
# t = 2.3791, df = 13.466, p-value = 0.03276
with(dtf, t.test(score[treat=="B"], score[treat=="F"]))
# t = 2.3212, df = 13.322, p-value = 0.03673
with(dtf, t.test(score[treat=="B"], score[treat=="C"]))
# t = 1.7112, df = 11.598, p-value = 0.1136
```

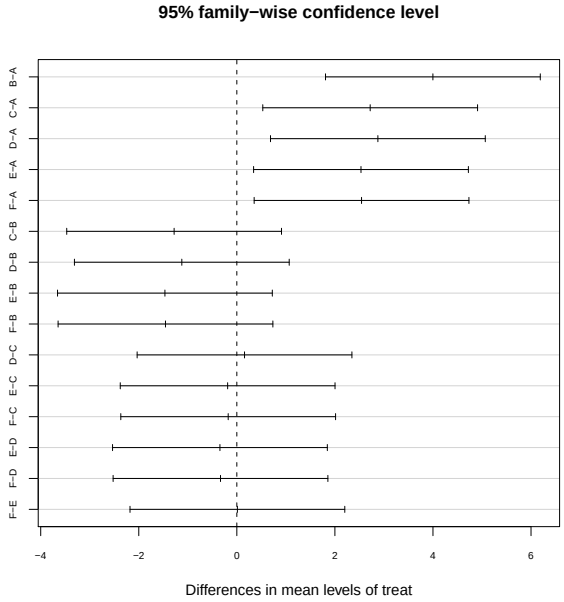
Question 3: Why is this equivalent to testing *all* pairs?

A human being always looks for the biggest difference. If your restrict snooping to a particular pair, you will not catch some other pair when it is the largest difference. To catch any pair that happens to be large, you have to be willing to look at all pairs.

To test (snoop) beyond the planned comparisons (or if there were no planned comparisons, which is generally a bad idea), *if* you decide (before looking at the data) that only paired tests are worthwhile, then the Tukey procedure is used. Ignore any differences already tested as planned. This is based on the distribution of the maximum of paired differences under the ANOVA null hypothesis.

```
> print(TukeyHSD(e0),digits=3)
  Tukey multiple comparisons of means
    95% family-wise confidence level
Fit: aov(formula = score ~ treat, data = dtf)
$treat
       diff    lwr   upr p adj
B-A  3.9988  1.809 6.189 0.000
C-A  2.7197  0.530 4.910 0.007
D-A  2.8767  0.686 5.067 0.004
E-A  2.5321  0.342 4.722 0.015
F-A  2.5435  0.353 4.734 0.015
C-B -1.2791 -3.469 0.911 0.512
D-B -1.1221 -3.312 1.068 0.648
E-B -1.4667 -3.657 0.723 0.360
F-B -1.4553 -3.645 0.735 0.369
D-C  0.1569 -2.033 2.347 1.000
E-C -0.1876 -2.378 2.003 1.000
F-C -0.1762 -2.366 2.014 1.000
E-D -0.3446 -2.535 1.846 0.997
```

```
F-D -0.3331 -2.523 1.857 0.997
F-E  0.0115 -2.179 2.202 1.000
> plot(TukeyHSD(e0, cex.axis=0.7)) # Without cex.axis, some labels are hidden.
```



Question 4: How do you interpret the plot?

Using the Tukey procedure to correct for the post-hoc testing of all pairs, we see a significant difference for the first 5 pairs, which have corrected 95% CIs that do not cross zero.

Question 5: Multiple choice: with the Tukey procedure the power to detect a true difference is larger/smaller/same compared to all t-tests.

With any multiple correction procedure power is smaller; this is the price we must pay to protect our type 1 error rate.

Now suppose you have a fixed set of contrasts (pairs, triples, etc.) that you plan to test. The Bonferroni procedure says that if there are m tests, then test them at $\alpha' = \alpha/m$. If some of the m are planned, use the `fit.contrast()` results for those tests. E.g. we plan to test all pairs plus all "2 vs. 3" comparisons, with none having higher interest than any others. There are `choose(6,2)`=15 pairs and 6·`choose(5,2)`=60 ways to set up 2 vs. 3 comparisons, so $m = 75$.

Reject only those tests with $p \leq (0.05/75) = 0.000667$. You could list all 75 tests and test them using `fit.contrast()` with either 75 function calls, or with some cleverly thought about sets of orthogonal contrasts. Practically you might just test those that look likely to show large differences, e.g., for pairs we can reinterpret the t-tests above (or slightly better use contrasts with the pooled SE, as below), and for the "2 vs. 3" tests, start with two highest vs. three lowest and the three highest vs. two lowest:

```
fit.contrast(e0, "treat", rbind(A.B=c(1,-1,0,0,0,0), A.C=c(1,0,-1,0,0,0)))
#          Estimate Std. Error  t value Pr(>|t|)
# treatA.B -3.99882    0.73368 -5.45037  0.00000
```

```
# treatA.C -2.71972    0.73368 -3.70698  0.00061
fit.contrast(e0, "treat", rbind(A.D=c(1,0,0,-1,0,0), A.E=c(1,0,0,0,-1,0)))
#           Estimate Std. Error   t value     Pr(>|t|)
# treatA.D -2.876669  0.7336774 -3.920890 0.0003200433
# treatA.E -2.532089  0.7336774 -3.451230 0.0012846660
fit.contrast(e0, "treat", rbind(BD.AEF=c(-1/2,1/3,0,-1/2,1/3,1/3),
                                AE.BCD=c(1/2,-1/3,-1/3,-1/3,1/2,0)))
#              Estimate Std. Error   t value     Pr(>|t|)
# treatBD.AEF  1.586482  0.4735867  3.349929 0.0017165505
# treatAE.BCD -1.932358  0.4735867 -4.080262 0.0001967149
```

Question 6: Which contrast would you try next?

From the 5 sample means (above), AE vs. BDE is also likely to be significant.

For the intense data snooping approach of *all* possible contrasts, the Scheffe procedure is used. (It may also be used instead of Bonferroni for a fixed size set of contrasts if it has a more favorable cutoff.) The cutoff value for statistical significance on the F scale is, in R code: $(I-1)*\text{qF}(1-\alpha, I-1, N-I)$, where N is the total sample size, and $N-I$ is the within-group df. We can take the square root to get on the t-test scale:

```
> sqrt(5 * qf(0.95, 5, 42))
3.491198
```

So we are "safe" rejecting any contrasts with t-values less than -3.49 or greater than +3.49 if we are an intense snooper.

For example we might be (post hoc) interested in A vs all others, and we can fairly test this with:

```
> fit.contrast(e0, "treat", rbind(A.Oth=c(1,-1/5,-1/5,-1/5,-1/5,-1/5)))
            Estimate Std. Error   t value Pr(>|t|)
treatA.Oth -2.934168  0.5683041 -5.163025 6.26e-06
```

Because $-5.16 \leq -3.49$ we can say that we have sufficient evidence to claim that the population mean of A is lower than the average of the other population means.