Displays for Statistics 5303

Lecture 11

September 27, 2002

Christopher Bingham, Instructor

612-625-7023 (St. Paul)
612-625-1024 (Minneapolis)

Class Web Page
http://www.stat.umn.edu/~kb/classes/5303

Summary of XSD methods based on "significant" differences

These methods all are based on a significant difference XSD where X is L, H, B or some other letter.

For each method, you compute XSD as

$$XSD = C_X(\alpha) \times \hat{SE}[\overline{y_{i\bullet}} - \overline{y_{j\bullet}}],$$

where the estimated standard error is

$$\hat{SE}[\overline{y_{i\bullet}} - \overline{y_{j\bullet}}] = s_p\sqrt{\{1/n_i + 1/n_j\}}$$
$$= s_p\sqrt{(2/n)} \text{ when } n_i = n_j = n,$$

$s_p = \sqrt{\{MS_E\}}$ and $C_X(\alpha)$ is a constant which is specific to the method.

When the sample sizes $\{n_i\}$ are not all equal, the value of XSD may be different for different pairs of treatments.

Effects α_i and α_j are significantly different when $\left|\hat{\alpha}_{i\bullet} - \hat{\alpha}_{j\bullet}\right| \geq XSD$. Thus XSD is the minimum significant difference for the method.

LSD (**L**east **S**ignificant **D**ifference)

$$C_L(\alpha) = t_{\alpha/2,df_{error}} = t_{\alpha/2,N-g}$$

Use of the LSD controls the per comparison error rate, and when preceded by an F-test, controls the per experiment error rate. But it can have a large *strong experimentwise error rate* (consider the case when $\mu_1 << \mu_2 = \mu_3 = \ldots = \mu_g$)

BSD (**B**onferroni **S**ignificant **D**ifference)

$$C_B(\alpha) = t_{(\alpha/K)/2,df_{error}} = t_{(\alpha/K)/2,\, N-g}$$

where $K = g(g-1)/2$

This controls the *strong experimentwise error rate*, but can be quite conservative (actual significance << intended). Consider the case

$\mu_1, \ldots, \mu_{g-1}$ all very different but $\mu_g = \mu_{g-1}$

The S experimentwise error rate will be close to $\alpha/K << \alpha$.

HSD Honestly **S**ignificant **D**ifference

$$C_H(\alpha) = q_\alpha(g, df_{error})/\sqrt{2}$$

where $q_\alpha(g, df_{error})$ is an upper probability point of the Studentized range distribution.

Usually the formula for HSD is given as

$$HSD = q_\alpha(g, df_{error}) \times s_p/\sqrt{n}$$

SSD Scheffe **S**ignificant **D**ifference

$$C_S(\alpha) = \sqrt{\{(g-1) \times F_{\alpha,g-1,df_{error}}\}}$$

This controls the family wise error rate for the entire family of all contrasts.

In MacAnova, you can compute these XSD's by

```
Cmd> stderror <- sqrt(mse)*sqrt(2/n) # standard error
Cmd> lsd <- invstu(1-alpha/2,dferror)*stderror # LSD
Cmd> K <- g*(g-1)/2 # number of pairwise comparisons
Cmd> bsd <- invstu(1-(alpha/K)/2,dferror)*stderror # BSD
Cmd> hsd <- (invstudrng(1-alpha,g,dferror)*stderror/sqrt(2)#HSD
Cmd> ssd <- sqrt((g-1)*invF(1 - alpha, g-1, dferror)) # SSD
```

Example (from Ex. 3.3). Study of resistance to trampling. 20 lanes (EU's) in New Hampsire alpine meadows, with 5 treatments, 0, 25, 75, 200, 500 walking passes by 70 kg person. Response is average vegetation height one year later.

```
Cmd> data33 <- read("","pr3.3",quiet:T) # Problem 3.3 data
Read from file "TP1:Stat5303:Data:OeCh03.dat"

Cmd> treat <- factor(data33[,1]) # create treatment factor

Cmd> height <- vector(data33[,2])

Cmd> anova("height = treat",fstat:T)
Model used is height = treat
              DF          SS          MS           F     P-value
CONSTANT       1      2782.4      2782.4  1349.49826           0
treat          4      243.16       60.79    29.48371  5.9878e-07
ERROR1        15      30.928      2.0618

Cmd> g <- 5; n <- 4; dferror <- 15 # or dferror <- DF[3]

Cmd> mse <- SS[3]/DF[3]; mse
      ERROR1
      2.0618

Cmd> stderror <- sqrt(2*mse/n); stderror
(1)      1.0153   Estimated SE[alphahat_i - alphahat_j]

Cmd> alpha <- .05 # signficance level

Cmd> lsd <- invstu(1-alpha/2, dferror)*stderror

Cmd> K <- g*(g-1)/2; K
(1)          10

Cmd> bsd <- invstu(1 - (alpha/K)/2, dferror)

Cmd> hsd <- invstudrng(1 - alpha, g, dferror)*stderror/sqrt(2)

Cmd> ssd <- sqrt((g-1)*invF(1 - alpha, g-1, dferror))*stderror

Cmd> vector(lsd, bsd, hsd, ssd) # 4 XSD's
(1)      2.1641       3.286      3.1354      3.5497
```


Here are summaries of comparisons of all pairs by these methods:

```
Cmd> pairwise("treat",.05,lsd:T)
  |      5    -3.79
  |      4    -2.79
    |    3     0.18
    |    2    0.205
         1     6.21

Cmd> pairwise("treat",.05,bsd:T)
  |      5    -3.79
  | |    4    -2.79
    |    3     0.18
    |    2    0.205
         1     6.21

Cmd> pairwise("treat",.05,hsd:T)
  |      5    -3.79
  | |    4    -2.79
    |    3     0.18
    |    2    0.205
         1     6.21
```

There is no Scheffe option for `pairwise()`, so I did it "by hand":

```
Cmd> sort(ybars)[-5] + ssd
(1)       11.55       12.55       15.525       15.55

Cmd> sort(ybars)
(1)           8           9      11.975          12          18
```

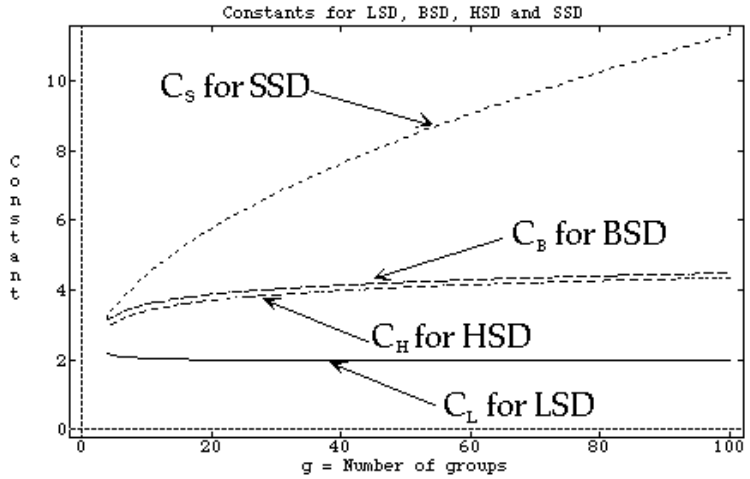
Here are just the constants C_X:

```
Cmd> c <- vector(invstu(1-alpha/2, dferror),\
       invstu(1 - (alpha/K)/2, dferror),\
       invstudrng(1 - alpha, g, dferror)/sqrt(2),\
       sqrt((g-1)*invF(1 - alpha, g-1, dferror)))

Cmd> vector(c,labels:vector("LSD","BSD","HSD","Scheffe"))
        LSD         BSD         HSD     Scheffe
     2.1314       3.286       3.088       3.496
```

Here are commands that compute the constants for g groups, g = 4, 5, ..., 100 with n = 4 observations per group.

```
Cmd> G <- run(4,100); n <- 4
Cmd> Edf <- n*G - G # vector of error df for all g's
Cmd> K <- G*(G-1)/2 # vector of Bonferronizing constants
Cmd> c_l <- invstu(1 - alpha/2,Edf) # for LSD
Cmd> c_b <- invstu(1 - (alpha/K)/2,Edf) # for BSD
Cmd> c_h <- invstudrng(1 - alpha, G, Edf)/sqrt(2) # for HSD
Cmd> c_s <- sqrt((G-1)*invF(1-alpha,G-1,Edf)) # for Scheffe
Cmd> lineplot(G,hconcat(c_l,c_b,c_h,c_s),ymin:0,xmin:0,\
     title:"Constants for LSD, BSD, HSD and SSD",\
     xlab:"g = Number of groups", ylab:"Constant")
```



Clearly HSD is preferred to BSD (it's smaller); SSD should not be used.

The reason the Scheffe method is so bad is because it is controlling the *family-wise* error for the *infinite* set of *all* contrasts $\sum w_i\alpha_i$, $\sum_i w_i = 0$.

Recall the standard error of a contrast is

$$\hat{SE}_w = \sqrt{\{\sum_i w_i^2/n_i\}}s_p$$

The family of confidence intervals

$$\sum_i w_i\alpha_i = \sum_i w_i\hat{\alpha}_i \pm \sqrt{\{(g-1)F_{1-\alpha,g-1,N-g}\}}\times\hat{SE}_w$$

has *simultaneous* coverage $1-\alpha$, that is

P(any interval does not cover its $\sum_i w_i\alpha_i$) = α and

P(*all* intervals cover their $\sum_i w_i\alpha_i$) = $1-\alpha$

It can be shown that

$\max_w(\sum_i w_i\hat{\alpha}_i/\hat{SE}_w) = \sqrt{\{(g-1)\times F\}}$, $F=MS_{trt}/MS_E$

The quantity on the left is the largest possible t-statistic for testing H_0:$\sum_i w_i\alpha_i$ so

$P(\max_w(\sum_i w_i\hat{\alpha}_i/\hat{SE}_w) > \sqrt{\{(g-1)\times F_{1-\alpha,g-1,N-g}\}}) = \alpha$

HSD is designed just for *pairwise* comparisons, although it can be extended to all contrasts.

You compare $\left|\sum_i w_i \hat{\alpha}_i\right|$ with

$$HSD_W \equiv q_\alpha(g, N-g)\{\sum |w_i|/2\}s_p$$

Note that $\{\sum |w_i| \times s_p\}/2$ is *not* a multiple of $\hat{SE}[\sum_i w_i \hat{\alpha}_i]$, the estimated standard error of $\sum_i w_i \hat{\alpha}_i$. You probably shouldn't use this; it's usually worse than Scheffe.

BSD is a particular case of the Bonferroni method for a family of K contrasts defined by weights $\{w_i^\ell\}$, $\ell = 1, \ldots, K$, *chosen in advance of seeing the data.*

You compare each observed contrast $\left|\sum_i w_i^\ell \hat{\alpha}_i\right|$ with $t_{(\alpha/K)/2, N-g} \hat{SE}_W$

LSD is a particular case of the naive method for all contrasts. You compare $\left|\sum_i w_i \hat{\alpha}_i\right|$ with $t_{\alpha/2, N-g} \hat{SE}_W$.

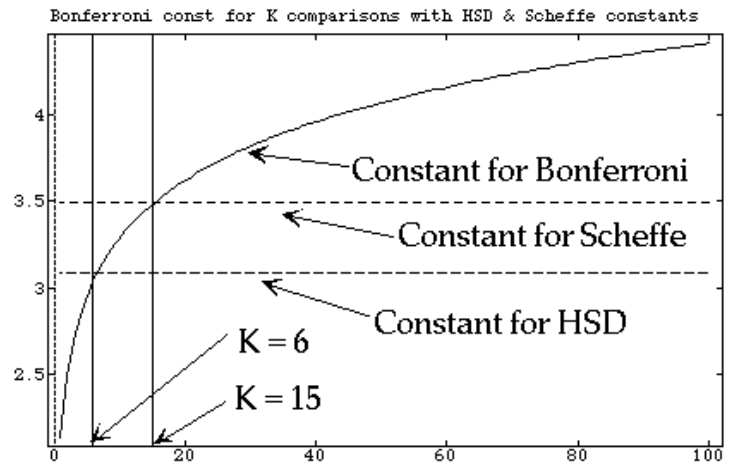
There are *only* two situations in which you should consider the Scheffe method:

1. You want to test K *picked in advance* contrasts where K is substantially greater than the number g(g-1)/2 of all pairwise contrasts.
2. You want to test one or more contrasts selected after seeing the data.

This graph on the next overhead compares the constants $t_{(\alpha/2)/K, dfe}$ (Bonferroni), $\sqrt{\{(g-1)F_{\alpha, g-1, dfe}\}}$ (Scheffe) and $q_\alpha(g, dfe)/\sqrt{2}$ (BSD) for K from 1 to 100.

g = 5 and dfe = 15 was assumed.

Only the Bonferroni constant depends on K.



The Bonferroni constant $t_{(\alpha/K)/2,N-g}$ for K contrasts is plotted against K together with the HSD constant for paired comparisons and the Scheffe constant for all contrasts in the case $g = 5$, $df_{error} = 15$.

It shows that when you are testing 6 or fewer pre-chosen contrasts, Bonferroni is better than HSD. If you are testing more than 15 contrasts, then Scheffe is better than Bonferroni.

A disadvantage of the HSD is that it assumes equal sample sizes.

A natural path to extend it to unequal sample sizes is to use the standard error

$$\hat{SE}[\overline{y_{i\bullet}} - \overline{y_{j\bullet}}] = \sqrt{\{s_p^2(1/n_i + 1/n_j)\}}$$

in place of $\sqrt{\{s_p^2 \times 2/n\}}$ in computing an Honestly Significant Difference

$$HSD_{ij} = (q_\alpha(g, df_{error})/\sqrt{2})\sqrt{\{s_p^2(1/n_i + 1/n_j)\}}$$

The use of HSD_{ij} is known as the Tukey-Cramer method.

Unlike the HSD in the equal sample size case, Tukey-Cramer is not an "exact" method, since the Sudentized range of means based on unequal sample sizes doesn't have the usual Studentized range distribution.

But it's close. I did a simulation with 10,000 replications with g = 4 and n = (5, 10, 15, 20). For each replication I computed I computed the "t" statistic

$$(\overline{y}_{(g)} - \overline{y}_{(1)})/\sqrt{\{s_p^2(1/n_{(g)} + 1/n_{(1)})\}}$$

where $\overline{y}_{(1)}$ and $\overline{y}_{(1)}$ were the the smallest and largest means based on $n_{(1)}$ and $n_{(g)}$ cases. It exceeded $q_{.05}(4,46)/\sqrt{2} = 2.666$ 469 times out of 10000, not significantly different from 500/10,000 = .05. In general, it's conservative, i.e. the error rate is a little less than the intended α.

Sequential Methods

These are methods that use different significant differences at different stages.

First you compare $\hat{\alpha}_{(g)} - \hat{\alpha}_{(1)}$, where $\hat{\alpha}_{(1)} = \min(\{\hat{\alpha}_i\})$ and $\hat{\alpha}_{(g)} = \max(\{\hat{\alpha}_i\})$, with the significant difference for g groups.

If they are not significantly different, you stop and say there are no significant differences and draw a line under all ordered effects or means.

If $\hat{\alpha}_{(g)} - \hat{\alpha}_{(1)}$ *is* significantly different from 0, you compare $\hat{\alpha}_{(g-1)} - \hat{\alpha}_{(1)}$ and $\hat{\alpha}_{(g)} - \hat{\alpha}_{(2)}$ with the significant difference for g - 1 groups. If neither is is significant you stop with a pattern of lines like

<u>1 2 3 . . . . . g-1 g</u>

If $\hat{\alpha}_{(g-1)} - \hat{\alpha}_{(1)}$ is significantly different from 0, you repeat this process comparing $\hat{\alpha}_{(g-2)} - \hat{\alpha}_{(1)}$ and $\hat{\alpha}_{(g-1)} - \hat{\alpha}_{(2)}$ with the significant difference for g-2 groups.

You do something similar $\hat{\alpha}_{(g)} - \hat{\alpha}_{(2)}$ is significantly different from 0.

You keep stepping down, doing the same thing at each stage, but using the significant difference appropriate to the number of groups that might be homogeneous.

The SNK (Student-Neuman-Keuls method) is of this form. At each stage, when you are checking whether $\hat{\alpha}_{(i)} - \hat{\alpha}_{(j)}$ is significant, it is compared with $(q_{\alpha}(i-j+1, N-g)/\sqrt{2})\sqrt{\{s_p^2(1/n_{(i)}+1/n_{(j)})\}}$, the HSD for $g' = i - j + 1$ groups.

Here is an illustration, using the same data as previously.

```
Cmd> ybars <- tabs(height,treat,means:T)

Cmd> ybar_sorted <- sort(ybars);ybar_sorted
(1)           8           9      11.975          12          18

Cmd> id <- grade(ybars);id # treatment numbers
(1)           5           4           3           2           1

Cmd> mse <- SS[3]/DF[3] # mean square error s^2

Cmd> hsd5 <- sqrt(mse/n)*invstudrng(1 - alpha,5,15); hsd5
(1)      3.1354
```

hsd5 is the HSD for g = 4, df_{error} =15.

```
Cmd> ybar_sorted[5] - ybar_sorted[1] # largest difference
(1)          10
```

Since 10 > 3.1354, 18 and 8 are significantly different and you go on to check the range of the two sets of 4 consecutive means

```
Cmd> hsd4 <- sqrt(mse/n)*invstudrng(1 - alpha,4,15); hsd4
(1)      2.9265
```

hsd4 is the HSD for g = 4, df_{error} =15.

```
Cmd> J1 <- run(2);J2 <- run(4,5)

Cmd> ybar_sorted[J2] - ybar_sorted[J1]
(1)           4           9
```

These are 12 - 8 and 18 - 9. Both exceed 2.93 so 12 and 8 are significantly different as are 18 and 9.

We continue on, looking at subsets of size 3:

```
Cmd> hsd3 <- sqrt(mse/n)*invstudrng(1 - alpha,3,15); hsd3
(1)      2.6374
```

hsd3 is the HSD for g = 3, df_{error} =15.

```
Cmd> J1 <- run(3);J2 <- run(3,5)

Cmd> ybar_sorted[J2] - ybar_sorted[J1]
(1)       3.975           3       6.025
```

These are 11.975 - 8, 12 - 9 and 18-11.975. All exceed 2.6374 so no group of 3 means appears to be homogeneous.

```
Cmd> hsd2 <- sqrt(mse/n)*invstudrng(1 - alpha,2,15); hsd2
(1)      2.1643
```

hsd2 is the HSD for g = 2, df_{error} =15.

```
Cmd> J1 <- run(4);J2 <- run(2,5)

Cmd> ybar_sorted[J2] - ybar_sorted[J1]
(1)           1       2.975       0.025           6
```

1 = $\overline{y}_{(2)\bullet} - \overline{y}_{(1)\bullet} < 2.1643$ and hence the two smallest means are not significantly different. Similarly. $\overline{y}_{(4)\bullet} - \overline{y}_{(3)\bullet} = .025 < 2.1643$. The other differences > 2.1643. You can summarize this by

```
Cmd> ybar_sorted
(1)           8           9      11.975          12          18
```


Here's the black box way to get to the same point.

```
Cmd> pairwise("treat",.05,snk:T)
  |       5    -3.79
  |       4    -2.79
    |     3     0.18
    |     2    0.205
          1     6.21
```

Note that the SNK method gives the same pattern of significant differences as the LSD method (not usually the case).

Unlike the LSD method, it has a low strong experimentwise error rate and low FDR = false discovery rate.