Displays for Statistics 5303

Lecture 8

September 20, 2002

Christopher Bingham, Instructor

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

Class Web Page

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

Quick review of contrasts in MacAnova

```
Cmd> anova("logy=treat",fstat:T) # same as before
Model used is logy=treat
WARNING: summaries are sequential
                DF          SS          MS           F     P-value
CONSTANT         1      79.425      79.425  8653.95365  1.6145e-40
treat            4      3.5376     0.88441    96.36296  2.2419e-17
ERROR1          32     0.29369   0.0091779

Cmd> muhats <- tabs(logy,treat,mean:T) # sample means

Cmd> muhats - sum(muhats)/5 # direct computation of effects
(1)     0.49456     0.19081    -0.06044    -0.24365    -0.38127

Cmd> alphahats <- coefs(treat); alphahats #black box effects
(1)     0.49456     0.19081    -0.06044    -0.24365    -0.38127

Cmd> w <- vector(vector(1,1)/2,-vector(1,1,1)/3);w # contrast
(1)         0.5         0.5    -0.33333    -0.33333    -0.33333

Cmd> result <- contrast(treat,w); result
component: estimate
(1)     0.57114      Value of contrast
component: ss
(1)      2.9446      SS for contrast
component: se
(1)    0.031886      Standard error of contrast

Cmd> tstat <- result$estimate/result$se; tstat
(1)      17.912      t-statistic to test H0:sum(w*alphas)=0

Cmd> errorss <- SS[3]; errordf <- DF[3]; mse <- errorss/errordf

Cmd> vector(errorss, errordf, mse)
      ERROR1      ERROR1      ERROR1
     0.29369          32   0.0091779

Cmd> tstat <- result$estimate/result$se; tstat
(1)      17.912

Cmd> twotailt(tstat,errordf) # P-value (two tail)
(1)  3.0663e-18      Essentially 0

Cmd> fstat <- result$ss/mse; fstat # = 17.912^2
(1)      320.83      F-statistic with 1 d.f. in numerator

Cmd> 1 - cumF(fstat,1,errordf) # P-value (two tail)
(1)           0
```

Here's how you would compute a 95% confidence interval for

$$w(\{\alpha_i\}) = (\alpha_1 + \alpha_2)/2 - (\alpha_3 + \alpha_4 + \alpha_5)/3$$

```
Cmd> t_025 <- invstu(1 - .025, errordf); t_025
(1)      2.0369      t_32 probability point

Cmd> result$estimate + vector(-1,1)*t_025*result$se
(1)     0.50619     0.63609
```

Polynomial Contrast

I didn't previously discuss the use of tables of coefficients for equally spaced doses and equal sample sizes in Table D.6

For these data, the sample sizes differ and the temperatures are not equally spaced.

To illustrate the use of the tables, I am going to discard enough cases so that all sample sizes are 6.

The temperatures are almost equally spaced by 19. So I will use modified temperatures that are completely equally spaced by 19.

```
Cmd> tabs(logy,count:treat) # original sample sizes
(1)            8            8            8            7            6

Cmd> J <- vector(run(6),8+run(6),16+run(6),24+run(6),31+run(6))
```

Use J as a subscript to select first 6 cases in each group

```
Cmd> treat1 <- factor(treat[J]) # new treatment factor

Cmd> logy1 <- logy[J] # new response

Cmd> n1 <- tabs(logy1,treat1,count:T); n1 # new sample sizes
(1)            6            6            6            6            6

Cmd> temp1 <- run(175,251,19); temp1 # new temperatures
(1)          175          194          213          232          251

Cmd> temper1 <- temp1[treat1] # vector of length 30

Cmd> anova("logy1=P4(temper1)",fstat:T) # fit 4th order polynom
Model used is logy1=P4(temper1)
WARNING: summaries are sequential
                DF           SS           MS            F      P-value
CONSTANT         1       62.814       62.814   5938.94768   3.0168e-31
{temper1}        1       2.9526       2.9526    279.16183   4.4799e-15
{(temper1)^2}    1     0.061344     0.061344      5.79994     0.023727
{(temper1)^3}    1   0.00010667   0.00010667      0.01009      0.92081
{(temper1)^4}    1   0.00016095   0.00016095      0.01522      0.90281
ERROR1          25      0.26442     0.010577
```

The underlined values are the SS for the polynomial contrasts.

You can get the contrasts themselves or their standard errors this way, but that's OK since you would seldom need them.

Let's find the SS using the orthogonal polynomial contrast coefficients for g = 5 from Table D.6 on p. 630.

Here I entered them into a matrix (table) with 5 rows, with contrasts down columns rather than in rows as in the table.

```
Cmd> WP <- matrix(enter(-2 -1 0 1 2   2 -1 -2 -1 2  \
        -1 2 0 -2 1  1 -4 6 -4 1), 5)

Cmd> WP # each column is a set of contrast weights
(1,1)          -2           2          -1           1
(2,1)          -1          -1           2          -4
(3,1)           0          -2           0           6
(4,1)           1          -1          -2          -4
(5,1)           2           2           1           1
```

Do ANOVA so contrast can work.

```
Cmd> anova("logy1=treat1",silent:T)

Cmd> contrast(treat1, WP[,1]) # linear contrast
component: estimate
(1)     -2.2183
component: ss
(1)      2.9526
component: se
(1)     0.13277

Cmd> for(i,1,4){ss <- contrast(treat1,WP[,i])$ss
        print(paste("SS for temper1^",i," = ",ss,sep:""))
     }
SS for temper1^1 = 2.9526          -2,-1,0,1,2  linear
SS for temper1^2 = 0.061344        2,-1,0,-1,2  quadratic
SS for temper1^3 = 0.00010667      -1,2,0,-2,1  cubic
SS for temper1^4 = 0.00016095      1,-4,6,-4,1  quartic
```

These match the SS from **anova()** output.

Multiple Comparisons

The ANOVA F-test is just the beginning. It tests the null hypothesis that all the treatment means are the same, or equivalently, that all the treatment have the same effects.

H_0: $\mu_1 = \mu_2 = \ldots = \mu_g$

H_a: $\mu_i \neq \mu_j$ for at least one pair $i \neq j$

or

H_0: $\alpha_1 = \alpha_2 = \ldots = \alpha_g$

H_a: $\alpha_i \neq \alpha_j$ for at least one pair $i \neq j$

When you reject H_0, what should you do next?

Your goal is to understand the pattern of treatment means, often with several specific questions in mind.

Often you would like to determine, for any two treatments, whether their effects are significantly different.

And this is easy to do for any fixed pair of treatments, chosen before looking at the data, say treatment i and treatment j. You just test $H_0^{(ij)}: \alpha_i - \alpha_j = 0$ using a t-test based on $\overline{y}_{i\bullet} - \overline{y}_{j\bullet} = \hat{\alpha}_i - \hat{\alpha}_j$.

What is the defining property of the test?

$$\boxed{\text{When } \mu_i = \mu_j,\ P(\text{reject } H_0^{(ij)}) = \alpha}$$

where α is the chosen significance level, say $\alpha = .05$ or $\alpha = .01$.

Significance level α is an *error rate*, specifically a **type I error rate**.

This is the error rate for a single contrast and hence, in this context, is called the **per comparison error rate**.

Suppose you nominated two contrasts to test, say $\alpha_1 - \alpha_2$ (w = {1 -1, 0, 0, ...}) and $\alpha_3 - \alpha_4$ (w = {0, 0, 1, -1, 0, ...}).

That is, you want to test $H_0^{(12)}:\alpha_1 - \alpha_2 = 0$ and $H_0^{(34)}:\alpha_3 - \alpha_4 = 0$.

- These contrasts are orthogonal for any sample sizes
- Hence they are independent.

The t-statistics won't be exactly independent because they both have $s_p = \sqrt{MSE}$ in the denominator, but they should be almost independent.

For each comparison, you have type I error rate α.

Suppose both $H_0^{(12)}$ and $H_0^{(34)}$ are true, that is, $\mu_1 = \mu_2$ and $\mu_3 = \mu_4$.

What is P(you make some type I error), that is, the probability you erroneously reject $H_0^{(12)}$, $H_0^{(34)}$, or both?

Because of the almost independence,

$$P(\text{reject one or both}) =$$
$$1 - P(\text{not reject either})$$
$$\cong 1 - (1 - \alpha)^2 = 2\alpha - \alpha^2$$

For $\alpha = .05$ this is $.10 - .0025 = .0975$.

This is the **per two independent comparisons error rate**. It's much larger than the per single comparison error rate

If E_1 and E_2 are two events (outcomes that may or may not occur) in a probability model, then

$$P(E_1 \textit{ or } E_2) \leq P(E_1) + P(E_2)$$

This is the **Bonferroni inequality**.

If $E_1 = \{\text{reject } H_0^{(ij)}\}$ and $E_2 = \{\text{reject } H_0^{(k\ell)}\}$, it guarantees that the per two comparisons error rate $\leq 2\times\alpha$.

More generally, the Bonferroni inequality for K events, E_1, E_2, ..., E_K states that $P(E_1 \textit{ or } E_2 \textit{ or} \ldots \textit{or } E_K) \leq \sum_{1\leq i\leq K} P(E_i)$.

This guarantees that the per K comparisons error rate, each of which is at significance level α is $\leq K\times\alpha$.

That is, if you test K *true* null hypotheses, the probability of rejecting one or more is bounded by $K\times\alpha$. In most cases, the probability is a lot closer to $K\alpha$ than to α.

Suppose you are interested in comparing all K = g(g-1)/2 pairs of effects. Even if every $H_0^{(ij)}$ is true (can happen only when $\alpha_1 = \alpha_2 = \ldots = \alpha_g$), for any testing procedure, there is some probability that you would make at least one type I error.

The probability of making at least one type I error would be the *experimentwise error rate* for the method used.

If you used t-tests with significance level α and they were all independent (they're not), the experimentwise error rate would be

$$1 - (1 - \alpha)^{g(g-1)/2}$$

```
Cmd> alpha <- .05; g <- 5

Cmd> 1 - (1 - alpha)^(g*(g-1)/2)
(1)     0.40126
```

This is a lot bigger than 5%. The Bonferroni upper bound for the experimentwise error rate is $(g(g-1)/2)\alpha$

```
Cmd> (g*(g-1)/2)*alpha
(1)         0.5
```

The Bonferroni method of multiple comparisons for a family of comparisons with K contrasts, uses α/K as the α-level for each comparison, where α is the desired family-wise error rate.

An equivalent way to do it is to multiply each ordinary P-value by K, obtaining what is sometimes called a *Bonferronized* P-value.

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

Cmd> data33[1,] # first case; shows col. 1 is the factor
(1,1)           1       20.7

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

Cmd> longevity <- vector(data33[,2]) # create response vector

Cmd> list(treat) # g = 5
treat           REAL    20    1     FACTOR with 5 levels

Cmd> tabs(longevity,treat,count:T) # n1 = n2 = n3 = n4 = n5 = 4
(1)             4           4           4           4           4

Cmd> anova("longevity=treat",fstat:T)
Model used is longevity=treat
               DF          SS          MS           F     P-value
CONSTANT        1      2782.4      2782.4  1349.49826  4.1416e-16
treat           4      243.16       60.79    29.48371  5.9878e-07
ERROR1         15      30.928      2.0618
```

The F-statistic shows there is very strong evidence the means differ.

```
Cmd> tabs(longevity,treat,mean:T) # sample treatment means
(1)          18         12      11.975          9          8
```

There are g(g-1)/2 = 10 pairwise comparisons.

```
Cmd> g <- 5; g*(g-1)/2
(1)          10
```

Here I enter a matrix whose columns define all 10 two-treatment comparisons

```
Cmd> W1 <- matrix(enter(1 -1 0 0 0   1 0 -1 0 0   1 0 0 -1 0\
    1 0 0 0 -1   0 1 -1 0 0   0 1 0 -1 0   0 1 0 0 -1\
    0 0 1 -1 0   0 0 1 0 -1   0 0 0 1 -1), 5)

Cmd> print(W1,format:"4.0f")
W:
(1,1)    1    1    1    1    0    0    0    0    0    0
(2,1)   -1    0    0    0    1    1    1    0    0    0
(3,1)    0   -1    0    0   -1    0    0    1    1    0
(4,1)    0    0   -1    0    0   -1    0   -1    0    1
(5,1)    0    0    0   -1    0    0   -1    0   -1   -1
```

I used a for loop in MacAnova to compute all 10 t-statistics using contrast():

```
Cmd> tstats <- rep(0,10) # place to put t-statistics

Cmd> for(i,1,10){
        result <- contrast(treat,W1[,i]) # uses column i of W
        tstats[i] <- result$estimate/result$se
    ;;}

Cmd> tstats
(1)      5.9093       5.934       8.864      9.8489    0.024622
(6)      2.9547      3.9396      2.9301      3.9149     0.98489

Cmd> pvals <- twotailt(tstats, DF[3]); pvals
(1)  2.8671e-05  2.7416e-05  2.3821e-07  6.1028e-08     0.98068
(6)   0.0098395   0.0013111    0.010344   0.0013786      0.3403
```

These are the ordinary P-values.

```
Cmd> 10*pvals
(1)  0.00028671  0.00027416  2.3821e-06  6.1028e-07      9.8068
(6)    0.098395    0.013111     0.10344    0.013786       3.403
```

These are Bonferronized P-values.

```
Cmd> t_025 <- invstu(1 - .025, DF[3]); t_025
(1)      2.1314      Ordinary critical value

Cmd> abs(tstats) > t_025 # T means signif. at ordinary 5% level
(1) T       T       T       T       F       T       T
(8) T       T       F    Grouped by left treatment

Cmd> bonf_t_025 <- invstu(1 - .025/10, DF[3]); bonf_t_025
(1)       3.286      Bonferronized critical value

Cmd> abs(tstats) > bonf_t_025 #significant by Bonferroni method
(1) T       T       T       T       F       F       T
(8) F       T       F
```

Here is a summary of the **ordinary** t-tests using underlining

```
Treatment   1   2   3   4   5
                -----   -----
```

Any treatments not significantly different are connected a line.

Here is a summary of the **Bonferronized** t-tests using underlining

```
Treatment   1   2   3   4   5
                ---------
                        -----
```

Another way to test the differences $\hat{\alpha}_i - \hat{\alpha}_j$ is to compare them with a precomputed significant difference = (critical value)×SE.

Such a difference for the Bonferroni method is called a **Bonferroni Significant Difference** or **BSD**. This is mainly used when all the sample sizes are the same so that all the standard errors are the same.

```
Cmd> se <- contrast(treat,W[,1])$se; se #  1 vs 2 contrast
(1)     1.0153

Cmd> contrast(treat,W1[,6])$se # 2 vs 4 contrast
(1)     1.0153      Same

Cmd> bsd <- bonf_t_025*se

Cmd> bsd # Bonferroni significant difference
(1)     3.3364
```

Any effect differences larger than BSD are significantly different from 0.

```
Cmd> diffs <- rep(0,10) # place to put differences

Cmd> for(i,1,10){ # compute them using contrast()
        diffs[i] <- contrast(treat,W1[,i])$estimate
    ;;}

Cmd> diffs # pairwise differences of alphahats
(1)          6      6.025          9         10       0.025
(6)          3          4      2.975      3.975           1
```

The underlined differences are greater than BSD = 3.3364.

Macro **pairwise()** summarizes the comparisons using vertical lines rather than horizontal lines.

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

BSD:T directs that the BSD is to be used. This is the same pattern as found before.