Displays for Statistics 5303

Lecture 22

October 25, 2002

Christopher Bingham, Instructor

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

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

More on 1-dofna

The **1-dofna** (one degree of freedom for non-additivity) model for two factor data is

$$y_{ij} = \mu + \alpha_i + \beta_j + \gamma\alpha_i\beta_j + \varepsilon_{ij}$$

This has interaction of the simple form

$$\alpha\beta_{ij} = \gamma\alpha_i\beta_j,\ i = 1,\ldots,a,\ j = 1,\ldots,b$$

This is most useful when there is no replication providing an error term to test an interaction SS. Even when the model is not *exactly* true, a test of H_0: $\gamma = 0$ is a valid test of H_0: $\alpha\beta_{ij} = 0$ against H_a: $\alpha\beta_{ij} = \gamma\alpha_i\beta_j$, with $\gamma \neq 0$.

If the 1-dofna F is significant, there is significant interaction.

If a power transformation y → y^p might provide a scale with an additive model, you can estimate p by $\hat{p} = 1 - \hat{\mu}\hat{\gamma}$.

Fitting the 1-dofna model takes two steps:

1. Fit the additive model $\mu + \alpha_i + \beta_j$ and from it find the **fitted values**
$$\hat{\mu}_{ij} = \hat{\mu} + \hat{\alpha}_i + \hat{\beta}_j.$$
2. Compute $z_{ij} = (\hat{\mu}_{ij} - \hat{\mu})^2/2 = (\hat{\alpha}_i + \hat{\beta}_j)^2/2$
3. Fit the model with an additional term
$$y_{ij} = \mu + \alpha_i + \beta_j + \gamma z_{ij} + \varepsilon_{ij}$$

The F- or t-statistic for z is a test of H_0: $\gamma = 0$, that is H_0: model is additive.

Instead of z_{ij}, you can use $\tilde{z}_{ij} = \hat{\alpha}_i\hat{\beta}_j$.

Oehlert's recipe uses
$$z_{ij}^* = (\hat{\mu}_{ij} - \hat{\mu})^2/(2\hat{\mu})$$
and after you fit the model
$$y_{ij} = \mu + \alpha_i + \beta_j + \gamma^* z_{ij}^* + \varepsilon_{ij}$$
you estimate p by $\hat{p} = 1 - \hat{\gamma}^*$.

Snedecor and Cochran example

```
Cmd> sned15_9 <- vector(19.1,23.4,29.5,23.4,16.6,\
      50.1,166.1,223.9,58.9,64.6, 123,407.4,398.1,229.1,251.2)

Cmd> print(matrix(sned15_9,5,\
      labels:structure("Trap ","Night ")),format:"11.1f")
MATRIX:
              Night 1     Night 2     Night 3
Trap 1           19.1        50.1       123.0
Trap 2           23.4       166.1       407.4
Trap 3           29.5       223.9       398.1
Trap 4           23.4        58.9       229.1
Trap 5           16.6        64.6       251.2

Cmd> period <- factor(rep(run(5),3))

Cmd> trap <- factor(rep(run(3),rep(5,3)))

Cmd> anova("sned15_9=period + trap", fstat:T)
Model used is sned15_9=period + trap
                DF          SS          MS           F     P-value
CONSTANT         1  2.8965e+05  2.8965e+05    75.70780  2.3739e-05
period           4       52066       13016     3.40223     0.06611
trap             2  1.7333e+05       86667    22.65276  0.00050731
ERROR1           8       30607      3825.9

Cmd> fitted <- sned15_9 - RESIDUALS  # muij_hat
```

fitted contains the additive fit.

```
Cmd> muhat <- coefs(CONSTANT)

Cmd> z <-  (fitted - muhat)^2/2

Cmd> anova("sned15_9=period + trap + z", fstat:T)
Model used is sned15_9=period + trap + z
WARNING: summaries are sequential
                DF          SS          MS           F     P-value
CONSTANT         1  2.8965e+05  2.8965e+05   322.45184  4.1032e-07
period           4       52066       13016    14.49064   0.0016932
trap             2  1.7333e+05       86667    96.48180  8.0263e-06
z                1       24319       24319    27.07330   0.0012486
ERROR1           7      6287.9      898.27

Cmd> phat <- 1 - muhat*coefs(z);phat # suggests a log transform
(1)     0.11653
```

```
Cmd> main_effects <- coefs()[vector(2,3)]

Cmd> main_effects
component: period        alpha_hats
(1)     -74.893      60.007       78.207       -35.16       -28.16
component: trap          beta_hats
(1)     -116.56      -26.24       142.8

Cmd> alpha_hat <- main_effects$period

Cmd> beta_hat <- main_effects$trap

Cmd> z_tilde <- alpha_hat[period]*beta_hat[trap]

Cmd> anova("sned15_9=period + trap + z_tilde", fstat:T)
Model used is sned15_9=period + trap + z_tilde
WARNING: summaries are sequential
              DF          SS          MS           F     P-value
CONSTANT       1  2.8965e+05  2.8965e+05   322.45184  4.1032e-07
period         4       52066       13016    14.49064   0.0016932
trap           2  1.7333e+05       86667    96.48180  8.0263e-06
z_tilde        1       24319       24319    27.07330   0.0012486
ERROR1         7      6287.9      898.27

Cmd> 1 - muhat*coefs(z_tilde) # 1 - muhat*gammahat
(1)     0.11653
```

ANOVA and $\hat{p}$ are the same for z and $\tilde{z}$.

Now do it Oehlert's way.

```
Cmd> z_star <- z/muhat # Oehlert's recipe

Cmd> anova("sned15_9=period + trap + z_star", fstat:T)
Model used is sned15_9=period + trap + z_star
WARNING: summaries are sequential
              DF          SS          MS           F     P-value
CONSTANT       1  2.8965e+05  2.8965e+05   322.45184  4.1032e-07
period         4       52066       13016    14.49064   0.0016932
trap           2  1.7333e+05       86667    96.48180  8.0263e-06
z_star         1       24319       24319    27.07330   0.0012486
ERROR1         7      6287.9      898.27

Cmd> 1 - coefs(z_star)  # 1 - gammahat_start
(1)     0.11653
```

A four factor example

```
Cmd> data8_8 <- read("","exmpl8.8")
exmpl8.8     54     5
) A data set from Oehlert (2000) \emph{A First Course in Design
) and Analysis of Experiments}, New York: W. H. Freeman.
)
) Data originally from an example (homework problem?) of Barry.
) Margolin
) Table 8.7, p. 187
) Columns are paging algorithm, initialization sequence, program
) size (small, medium, or large), RAM allocation (large, medium,
) or small), and number of page faults.
Read from file "TP1:Stat5303:Data:OeCh08.dat"

Cmd> makecols(data8_8,algo,seq,size,ram,faults)

Cmd> algo <- factor(algo); seq <- factor(seq)

Cmd> size <- factor(size); ram <- factor(ram)

Cmd> anova("faults=(algo+seq+size+ram)^3",pvals:T)
Model used is faults=(algo+seq+size+ram)^3
                  DF          SS          MS     P-value
CONSTANT           1  3.4326e+08  3.4326e+08   2.034e-11
algo               1  1.1672e+07  1.1672e+07  1.1402e-05
seq                2  5.9566e+07  2.9783e+07  7.7509e-08
size               2  2.1688e+08  1.0844e+08  4.6303e-10
ram                2  2.6155e+08  1.3077e+08  2.1962e-10
algo.seq           2  1.6333e+06  8.1666e+05    0.021354
algo.size          2  7.7345e+06  3.8672e+06  0.00017844
algo.ram           2  6.0014e+06  3.0007e+06  0.00043168
seq.size           4   3.955e+07  9.8875e+06  1.8896e-06
seq.ram            4  2.2268e+07  5.5671e+06  1.7152e-05
size.ram           4  1.5805e+08  3.9514e+07  8.1102e-09
algo.seq.size      4  1.4408e+06   3.602e+05    0.096821
algo.seq.ram       4  1.6047e+06  4.0119e+05    0.077039
algo.size.ram      4  3.8685e+06  9.6712e+05   0.0076816
seq.size.ram       8  1.2241e+07  1.5301e+06  0.00098119
ERROR1             8  1.0107e+06  1.2634e+05
```

There is plenty of interaction. Could some be reduced by transforming the data?

The 1-dofna model is more complicated for models with more than two factors, but the analysis is the same, including how you choose a transformation.

$$y_{ijk\ell} = \mu + \alpha_i + \beta_j + \gamma_k + \delta_\ell$$
$$+ \gamma(\alpha_i\beta_j + \alpha_i\gamma_k + \alpha_i\delta_\ell + \beta_j\gamma_k + \beta_j\delta_\ell + \gamma_k\delta_\ell) + \varepsilon_{ijk\ell}$$

First fit the additive model

```
Cmd> Model used is faults=algo+seq+size+ram
               DF          SS          MS     P-value
CONSTANT        1  3.4326e+08  3.4326e+08  4.7161e-10
algo            1  1.1672e+07  1.1672e+07     0.15388
seq             2  5.9566e+07  2.9783e+07   0.0080552
size            2  2.1688e+08  1.0844e+08  7.2377e-07
ram             2  2.6155e+08  1.3077e+08  9.0568e-08
ERROR1         46  2.5541e+08  5.5523e+06
```

Get all the main effect coefficients ($\hat{\alpha}_i$, $\hat{\beta}_j$, $\hat{\gamma}_k$, $\hat{\delta}_\ell$) (terms 2, 3, 4 and 5)

```
Cmd> muhat <- coefs(CONSTANT)

Cmd> maineffects <- coefs()[run(2,5)]

Cmd> maineffects
component: algo
(1)     -464.91      464.91               alphahat
component: seq
(1)     -1262.6      -46.13      1308.8   betahat
component: size
(1)     -2010.5      -724.8      2735.3   gammahat
component: ram
(1)     -2253.5     -732.46      2985.9   deltahat
Cmd> alphahat <- maineffects$algo; betahat <- maineffects$seq

Cmd> gammahat <- maineffects$size; deltahat <- maineffects$ram
```

Here I compute terms consisting of

$$z_{1,ij} = \hat{\alpha}_i\hat{\beta}_j,\ z_{2,ik} = \hat{\alpha}_i\hat{\gamma}_k,\ z_{3,i\ell} = \hat{\alpha}_i\hat{\delta}_\ell,$$
$$z_{4,jk} = \hat{\beta}_j\hat{\gamma}_k,\ z_{5,j\ell} = \hat{\beta}_j\hat{\delta}_\ell,\ z_{6,k\ell} = \hat{\gamma}_k\hat{\delta}_\ell,$$

```
Cmd> z1 <- alphahat[algo]*betahat[seq]

Cmd> z2 <- alphahat[algo]*gammahat[size]

Cmd> z3 <- alphahat[algo]*deltahat[ram]

Cmd> z4 <- betahat[seq]*gammahat[size]

Cmd> z5 <- betahat[seq]*deltahat[ram]

Cmd> z6 <- gammahat[size]*deltahat[ram]
```

The same way as for the two-factor:

```
Cmd> fitted <- faults - RESIDUALS

Cmd> z <- (fitted - muhat)^2/2

Cmd> anova("faults=algo+seq+size+ram + z",pvals:T)
Model used is faults=algo+seq+size+ram + z
WARNING: summaries are sequential
               DF          SS          MS     P-value
CONSTANT        1  3.4326e+08  3.4326e+08  1.0711e-23
algo            1  1.1672e+07  1.1672e+07  0.00074319
seq             2  5.9566e+07  2.9783e+07  1.2581e-09
size            2  2.1688e+08  1.0844e+08  6.9611e-19
ram             2  2.6155e+08  1.3077e+08  1.8906e-20
z               1  2.1533e+08  2.1533e+08  1.0233e-19
ERROR1         45  4.0074e+07  8.9054e+05

Cmd> 1 - coefs(z)*muhat # power
(1)     0.10122      p is small, suggesting a log transform
```

It's more understandable on the basis of the 1-dofna model to use

$$\tilde{z} = \hat{\alpha}_i\hat{\beta}_j + \hat{\alpha}_i\hat{\gamma}_k + \hat{\alpha}_i\hat{\delta}_\ell + \hat{\beta}_j\hat{\gamma}_k + \hat{\beta}_j\hat{\delta}_\ell + \hat{\gamma}_k\hat{\delta}_\ell$$

$$= z_1 + z_2 + z_3 + z_4 + z_5 + z_6$$

```
Cmd> anova("faults=algo+seq+size+ram+{z1+z2+z3+z4+z5+z6}",\
    pvals:T)
Model used is faults=algo+seq+size+ram+{z1+z2+z3+z4+z5+z6}
WARNING: summaries are sequential
                          DF          SS          MS      P-value
CONSTANT                   1  3.4326e+08  3.4326e+08   1.0711e-23
algo                       1  1.1672e+07  1.1672e+07   0.00074319
seq                        2  5.9566e+07  2.9783e+07   1.2581e-09
size                       2  2.1688e+08  1.0844e+08   6.9611e-19
ram                        2  2.6155e+08  1.3077e+08   1.8906e-20
{z1+z2+z3+z4+z5+z6}        1  2.1533e+08  2.1533e+08   1.0233e-19
ERROR1                    45  4.0074e+07  8.9054e+05

Cmd> 1 - coefs(6)*muhat
(1)     0.10122
```

This gives identical results as with z.

Let's see how much interaction is left by adding interaction terms *after* {z1+z2+z3+z4+z5+z6}

```
Cmd> anova("faults=algo+seq+size+ram+{z1+z2+z3+z4+z5+z6}+
(algo+seq+size+ram)^3",pvals:T)
Model used is
faults=algo+seq+size+ram+{z1+z2+z3+z4+z5+z6}+(algo+seq+size+ram)
^3
WARNING: summaries are sequential
                          DF          SS          MS      P-value
CONSTANT                   1  3.4326e+08  3.4326e+08    2.034e-11
algo                       1  1.1672e+07  1.1672e+07   1.1402e-05
seq                        2  5.9566e+07  2.9783e+07   7.7509e-08
size                       2  2.1688e+08  1.0844e+08   4.6303e-10
ram                        2  2.6155e+08  1.3077e+08   2.1962e-10
{z1+z2+z3+z4+z5+z6}        1  2.1533e+08  2.1533e+08   1.3052e-10
algo.seq                   2   1.002e+05       50100      0.68517
algo.size                  2  1.1915e+05       59575      0.64035
algo.ram                   2  3.0598e+05  1.5299e+05       0.3472
seq.size                   4  7.6407e+05  1.9102e+05      0.28632
seq.ram                    4  1.8546e+07  4.6365e+06   3.4199e-05
size.ram                   3       73061       24354      0.89842
algo.seq.size              4  1.4408e+06   3.602e+05     0.096821
algo.seq.ram               4  1.6047e+06  4.0119e+05     0.077039
algo.size.ram              4  3.8685e+06  9.6712e+05    0.0076816
seq.size.ram               8  1.2241e+07  1.5301e+06   0.00098119
ERROR1                     8  1.0107e+06  1.2634e+05
```

This one degree of freedom has accounted for all the two-way interaction except for **seq.ram**.

Could the **seq.ram** be SS be explained by $z_5 = \hat{\beta}_j\delta_\ell$, removing z_5 from the sum of z's?

```
Cmd> anova("faults=algo+seq+size+ram+{z1+z2+z3+z4+z6}+z5+
(algo+seq+size+ram)^3",pvals:T)
Model used is faults=algo+seq+size+ram+{z1+z2+z3+z4+z6}+z5+
(algo+seq+size+ram)^3
WARNING: summaries are sequential
                     DF           SS           MS      P-value
CONSTANT              1   3.4326e+08   3.4326e+08    2.034e-11
algo                  1   1.1672e+07   1.1672e+07   1.1402e-05
seq                   2   5.9566e+07   2.9783e+07   7.7509e-08
size                  2   2.1688e+08   1.0844e+08   4.6303e-10
ram                   2   2.6155e+08   1.3077e+08   2.1962e-10
{z1+z2+z3+z4+z6}      1   2.1212e+08   2.1212e+08   1.3857e-10
z5                    1   1.1636e+07   1.1636e+07   1.1533e-05
algo.seq              2   1.2277e+05        61385       0.6322
algo.size             2        15056       7528.2      0.94257
algo.ram              2    5.103e+05   2.5515e+05      0.19498
seq.size              4   1.2993e+05        32483      0.89736
seq.ram               3   1.0633e+07   3.5443e+06    0.0001348
size.ram              3        73061        24354      0.89842
algo.seq.size         4   1.4408e+06    3.602e+05     0.096821
algo.seq.ram          4   1.6047e+06   4.0119e+05     0.077039
algo.size.ram         4   3.8685e+06   9.6712e+05    0.0076816
seq.size.ram          8   1.2241e+07   1.5301e+06   0.00098119
ERROR1                8   1.0107e+06   1.2634e+05
```

$SS_{seq.ram}$ has been reduced from 1.85×10^7 in the previous analysis to 1.06×10^7 but it is still significant.

Unbalanced data

So far we have dealt only with **balanced data**.

Balance is hard to define precisely since it is not a property solely of the data but depends also on the model being fitted.

The simple cases of balance are:

- Non-factorial (or single factor) design with $n_1 = n_2 = \ldots = n_g$
- Complete factorial with all "cell sizes" $n_{i\ldots\ell}$ the same

Reminder: A **complete factorial** has all combinations of factor levels.

These are balanced for any factorial model, no matter what main effects and interactions are included in it.

If there is any inequality of cell sizes, the data are **unbalanced** for some or all factorial models.

For a **non-complete** three-factor model model with *only* main effects

$$y_{ijk\ell} = \mu + \alpha_i + \beta_j + \gamma_k + \varepsilon_{ijk\ell},$$

the data are balanced when

$n_{1++} = n_{2++} = . . = n_{a++}$

$n_{+1+} = n_{+2+} = . . = n_{+b+}$

$n_{++1} = n_{++2} = . . = n_{++c}$

All a×b n_{ij+}'s are equal

All a×c n_{i+k}'s are equal

All b×c n_{+jk}'s are equal

A subscript + indicates a sum over that subscript, so, for example, $n_{+j+} = \sum_i\sum_k n_{ijk}$.

Because it is non-complete, not all combinations of i, j and k are in the design, that is some $n_{ijk} = 0$.

Example: 4^3 factorial with treatments combinations assigned following a **Latin square**.

	B_1	B_2	B_3	B_4
A_1	C_1	C_2	C_3	C_4
A_2	C_2	C_3	C_4	C_1
A_3	C_3	C_4	C_1	C_2
A_4	C_4	C_1	C_2	C_3

This is incomplete because only 16 out of 4^3 = 64 combinations are actually used.

It is balanced for $\mu + \alpha_i + \beta_j + \gamma_k$ because

$n_{1++} = n_{2++} = . . = n_{4++} = 4$ Levels of A

$n_{+1+} = n_{+2+} = . . = n_{+4+} = 4$ Levels of B

$n_{++1} = n_{++2} = . . = n_{++4} = 4$ Levels of C

$n_{ij+} = n_{i+k} = n_{+jk} = 1$, all i,j,k

This data is *not* balanced for any model that includes an interaction.

Why is balance important?

The short answer is this.

When data are not balanced, then

- Calculation is much harder; you really need a computer program
- The order of terms in the model can make a difference in the SS, at least as computed by MacAnova (type I SS)
- The sums of squares used for testing don't add up to what you might think they should
- Two factors can each appear to be insigificant, but you need them both for a good fit.

An important advantage of balanced data:

Contrasts going with different terms in the model are **orthogonal**.

- Contrasts in different main effects are orthogonal.
- Main effect contrasts are orthogonal to interaction contrasts.
- Interaction contrasts associated with different interactions terms in the model are orthogonal.

It's really this property that results in the order of terms to be irrelevant with balanced data, but very important with unbalanced data.