Displays for Statistics 5303

Lecture 38

December 6, 2002

Christopher Bingham, Instructor

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

Class Web Page

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

Analysis of a confounded factorial

Here is a partial analysis of a 2^{7-4} design, 2^7 = 128 factorial combinations arranged in 2^4 = 16 blocks of size 2^{7-4} = 8.

```
Cmd> data <- read("","exmpl15.6")
exmpl15.6          128    9
) A data set from Oehlert (2000) \emph{A First Course in Design
) and Analysis of Experiments}, New York: W. H. Freeman.
)
) Table 15.6, p. 398
) Data for a 2^7 in standard order.  Factors are size of image,
) shape of image, color of image, orientation of image, duration
) of image vertical location of image, and horizontal location
) of image.
) ABCD, ACEG, BCE, BCFG, ACF, CDEF, ABG, BDEG, ADE, ADFG, BDF,
) EFG, CDG, ABEF, and ABCDEFG are confounded with blocks.
) Columns are block, A, B, C, D, E, F, G, and response
Read from file "TP1:Stat5303:Data:OeCh15.dat"
```

There seven 3-way, seven 4-way and one 7-way interactions confounded, 15 = 2^4-1 in all. A design equivalent to this one is found by **choosedef2(7,4,all:T)**.

There is r = 1 replicate.

```
Cmd> makecols(data,block,a,b,c,d,e,f,g,y,factors:run(8))
Column 1 saved as factor block  with 16 levels
Column 2 saved as factor a  with 2 levels
Column 3 saved as factor b  with 2 levels
Column 4 saved as factor c  with 2 levels
Column 5 saved as factor d  with 2 levels
Column 6 saved as factor e  with 2 levels
Column 7 saved as factor f  with 2 levels
Column 8 saved as factor g  with 2 levels
Column 9 saved as vector y
```

Here is an ANOVA, including main effects plus 2-way, 3-way and 4-way interactions. This is an *intrablock* analysis.

```
Cmd> anova("y=block + (a+b+c+d+e+f+g)^4")
Model used is y=block + (a+b+c+d+e+f+g)^4
WARNING: summaries are sequential
                DF          SS          MS
CONSTANT         1       30.44       30.44
block           15      1.3384    0.089225
a                1     0.02645     0.02645
b                1    0.014028    0.014028
c                1   0.0091125   0.0091125
d                1      0.0162      0.0162
e                1      1.9602      1.9602
f                1   0.0038281   0.0038281
g                1   0.0011281   0.0011281
a.b              1       0.005       0.005
a.c              1   0.0022781   0.0022781
a.d              1    0.029403    0.029403
a.e              1   0.0034031   0.0034031
a.f              1    0.017112    0.017112
a.g              1     0.01125     0.01125
b.c              1    0.021012    0.021012
b.d              1      0.0392      0.0392
b.e              1       5e-05       5e-05
b.f              1   0.0057781   0.0057781
b.g              1   0.0038281   0.0038281
c.d              1   0.0063281   0.0063281
c.e              1  0.00015312  0.00015312
c.f              1      0.0032      0.0032
c.g              1      0.0072      0.0072
d.e              1   0.0022781   0.0022781
d.f              1     0.00045     0.00045
d.g              1        0.02        0.02
e.f              1           0           0
e.g              1     0.00845     0.00845
f.g              1  0.00070312  0.00070312
a.b.c            1   0.0052531   0.0052531
a.b.d            1   0.0034031   0.0034031
a.b.e            1   0.0011281   0.0011281
a.b.f            1     0.00125     0.00125
a.b.g            0           0   undefined
a.c.d            1      0.0392      0.0392
```

```
a.c.e            1      0.0242      0.0242
a.c.f            0           0   undefined
a.c.g            1   0.0011281   0.0011281
a.d.e            0           0   undefined
a.d.f            1   3.125e-06   3.125e-06
a.d.g            1   0.0087781   0.0087781
a.e.f            1    0.045753    0.045753
a.e.g            1   0.0087781   0.0087781
a.f.g            1   0.0006125   0.0006125
b.c.d            1    0.024753    0.024753
b.c.e            0           0   undefined
b.c.f            1    0.037812    0.037812
b.c.g            1      0.0098      0.0098
b.d.e            1   0.0019531   0.0019531
b.d.f            0           0   undefined
b.d.g            1      0.0338      0.0338
b.e.f            1   0.0021125   0.0021125
b.e.g            1     0.00045     0.00045
b.f.g            1    0.025878    0.025878
c.d.e            1     0.01445     0.01445
c.d.f            1  0.00070312  0.00070312
c.d.g            0           0   undefined
c.e.f            1  0.00037812  0.00037812
c.e.g            1    0.067528    0.067528
c.f.g            1    0.027612    0.027612
d.e.f            1    0.015753    0.015753
d.e.g            1   0.0034031   0.0034031
d.f.g            1      0.0072      0.0072
e.f.g            0           0   undefined
a.b.c.d          0           0   undefined
a.b.c.e          1      0.0162      0.0162
a.b.c.f          1    0.015753    0.015753
a.b.c.g          1  0.00090313  0.00090313
a.b.d.e          1   0.0091125   0.0091125
a.b.d.f          1   0.0087781   0.0087781
a.b.d.g          1   0.0047531   0.0047531
a.b.e.f          0           0   undefined
a.b.e.g          1    0.038503    0.038503
a.b.f.g          1     0.00845     0.00845
a.c.d.e          1   0.0047531   0.0047531
a.c.d.f          1     0.00045     0.00045
a.c.d.g          1   0.0006125   0.0006125
a.c.e.f          1      0.0128      0.0128
a.c.e.g          0           0   undefined
a.c.f.g          1   0.0022781   0.0022781
a.d.e.f          1     0.00605     0.00605
a.d.e.g          1    0.012012    0.012012
a.d.f.g          0           0   undefined
```

```
a.e.f.g           1    0.011628    0.011628
b.c.d.e           1        0.02        0.02
b.c.d.f           1  0.00052812  0.00052812
b.c.d.g           1    0.014878    0.014878
b.c.e.f           1   0.0094531   0.0094531
b.c.e.g           1   0.0052531   0.0052531
b.c.f.g           0           0   undefined
b.d.e.f           1  0.00025313  0.00025313
b.d.e.g           0           0   undefined
b.d.f.g           1   0.0055125   0.0055125
b.e.f.g           1    0.032512    0.032512
c.d.e.f           0           0   undefined
c.d.e.g           1   0.0003125   0.0003125
c.d.f.g           1      0.1164      0.1164
c.e.f.g           1    0.023653    0.023653
d.e.f.g           1   0.0019531   0.0019531
ERROR1           28     0.59048    0.021088
```

The error term consists of all the pooled 5-, 6- and 7-way interaction SS.

Note the following:

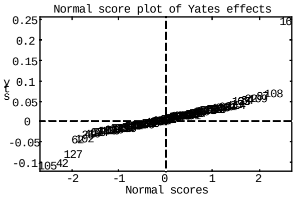
- By far the largest effect mean square is for **e**.
- Terms **a.b.g**, **a.c.f**, **a.d.e**, **b.c.e**, **b.d.f**, **c.d.g**, **e.f.g**, **a.b.c.d**, **a.b.e.f**, **a.c.e.g**, **a.d.f.g**, **b.c.f.g**, **b.d.e.g** and **c.d.e.f** (underlined) all have 0 degrees of freedom. These are 14 of the 15 confounded effects. The 15th, **a.b.c.d.e.f.g** would have been there if the model was **y=(a+b+c+d+e+f+g)^7**.

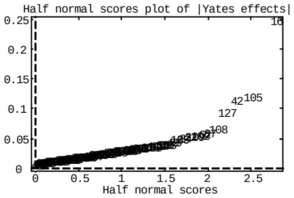
Just as with non-confounded designs you can look at Yates effects. However, 15 are confounded with blocks and may be large because of large block differences, not because of treatment differences.

```
Cmd> yts <- yates(y)

Cmd> chplot(rankits(yts),yts,xlab:"Normal scores",\
     title:"Normal score plot of Yates effects")

Cmd> chplot(halfnorm(abs(yts)),abs(yts),\
     xlab:"Half normal scores",\
     title:"Normal score plot of Yates effects")
```





Effects 16, 105, 42 and 127 appear to be outliers. How do you identify the effects?

It's not hard to check that main effects A, B, C, D, E, F, and G are effects 1, 2, 4, 8, 16, 32 and 64 (2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6), so the biggest effect (16) is clearly the unconfounded E main effect.

In binary notation these are

	GFEDCBA	Effect
16 =	0010000b	E
105 =	1101001b	ADFG
42 =	0101010b	BDF
127 =	1111111b	ABCDEFG

When the right most digit is 1, A is in the effect. When the 2nd from the right is 1, B is in the effect and so on.

Note that 105, 42 and 127 correspond to confounded contrast and represent differences between blocks, not treatment effects.

Here I compute in a very arcane way the indices of *all* the confounded effects.

```
Cmd> confounded <- vector("abg", "acf", "ade", "bce", "bdf",
"cdg", "efg", "abcd", "abef", "aceg", "adfg", "bcfg", "bdeg",
"cdef","abcdefg"); conf <- rep(0,15)

Cmd> for(i,1,15){
tmp <- match(vector("*a*","*b*","*c*","*d*","*e*","*f*","*g"),\
confounded[i],0,exact:F)
tmp[tmp != 0] <- 1
conf[i] <- sum(tmp*2^run(0,6));;}

Cmd> print(format:"3.0f",conf) # confounded terms
conf:
(1)  67  37  25  22  42  76 112  15  51  85 105 102  90  60 127
```

They include 42, 105 and 127.

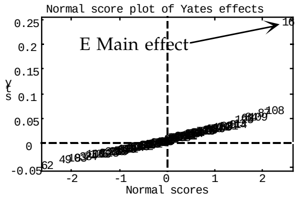
To be meaningful, a normal or half normal plots of Yates effects must omit the confounded ones. One way to do it is to replace them by **MISSING**. These will be ignored in the plot, except for warning messages.

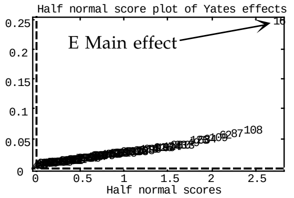
```
Cmd> yts[terms] <- ? # set con

Cmd> yts[grade(abs(yts),down:T)][run(10)]
WARNING: missing values in argument(s) to abs()
WARNING:  MISSING values in argument to grade()
(1)      0.2475    0.060313    0.054375    -0.05125    0.048125
(6)    0.045937    0.043438    0.042187   -0.037813       0.035

Cmd> chplot(rankits(yts),yts,xlab:"Normal scores",\
     title:"Normal score plot of Yates effects")
WARNING:  MISSING values in argument to rankits()

Cmd> chplot(halfnorm(abs(yts)),abs(yts),\
     xlab:"Half normal scores",\
     title:"Half normal score plot of Yates effects")
WARNING: missing values in argument(s) to abs()
WARNING:  MISSING values in argument to halfnorm()
WARNING: missing values in argument(s) to abs()
```





Only the E main effect stands out.

Split Plot designs

A split plot design can be thought of as a type of incomplete block design, where at least one main effects and sometimes interactions are confounded with blocks.

- *All* inference about the confounded effects effectively comes from an *interblock* analysis.
- All inference about the unconfounded effects comes from an *intrablock* analysis

The basic treatment structure is usually a complete factorial.

In the simplest case:

- You have just two factors, A and B with a and b levels, respectively.
- You have n×a "blocks" of size b, N=nab EU's in all. These are *whole plots*.

The experimental units within a block are called *split plots* or sometimes *sub-plots.*

One of the factors, say A, is the *whole plot factor*.

Treatment assignment is in two steps.

- You randomly assign levels to the whole plots and all the split plots in the whole plot get the same level of A.
- *Within* each whole plot, you randomly assign levels of B.

The result is a design that is balanced for treatments.

A classic example is an experiment to compare a = 3 levels of tillage (plowing) and b = 4 fertilizers on the yield of a crop in n = 3 replicates.

It's not practical to vary tillage between too small pieces of a field. So the field is divided into 9 homogeneous strips to which tillage methods A_1, A_2, and A_3 are assigned randomly. Then B_1, B_2, B_3 and B_4 are randomly assigned to the EU's (split or sub plots) in a block = whole plot.

1	2	3	4	5	6	7	8	9
A_2B_4	A_1B_3	A_2B_1	A_2B_3	A_3B_2	A_1B_3	A_1B_4	A_3B_4	A_3B_1
A_2B_3	A_1B_1	A_2B_3	A_2B_4	A_3B_3	A_1B_4	A_1B_1	A_3B_2	A_3B_2
A_2B_2	A_1B_4	A_2B_4	A_2B_1	A_3B_1	A_1B_2	A_1B_2	A_3B_3	A_3B_4
A_2B_1	A_1B_2	A_2B_2	A_2B_2	A_3B_4	A_1B_1	A_1B_3	A_3B	A_3B_3

Note there is one level of A and a complete set of B levels in each of the nine whole plots (blocks).

Another agricultural example:

The response is yields of oats.

The whole-plot and split-plot factors are

- A: 4 lots of oats, 1 infected with a fungus, 1 not, and 2 resistant.
- B: 4 protectants

The information about the 4 lots of oats might guide choosing contrasts. For example you might want to compare resistant and non resistant using contrast weights (-1,-1,1,1).

Seed lots were assigned to 16 contiguous areas in a field, the whole plots, 4 whole plots per seed lot.

Each whole plot was subdivided into four compact areas to which the four protectant treatments were randomly assigned.

If you didn't have the protectant or it had 0 effect, you would have a model like

$$y_{ijk} = \mu + \alpha_i + \eta_{k(i)} + \varepsilon_{jk(i)}.$$

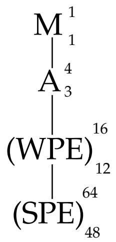
- α_i = seed type effect
- $\eta_{k(i)}$ = random block effect for the k^th block with seed type i. $\eta_{k(i)}$ is the *whole plot error* with variance σ_η^2
- $\varepsilon_{jk(i)}$ = random split plot or subplot error with variance σ^2. These errors are different in each split plot and is the split plot.

You could analyze it using block means:

$$\overline{y}_{ij\bullet} = \mu + \alpha_i + \overline{\eta}_{\bullet(i)} + \overline{\varepsilon}_{j\bullet(i)}$$
$$= \mu + \alpha_i + \tilde{\varepsilon}_{ij}$$

This is a standard one factor ANOVA model with errors $\tilde{\varepsilon}_{ij}$. Because means are derived from block totals, this is really an *interblock* analysis.

What would a Hasse diagram for this design look like?

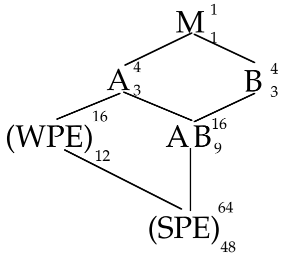


The denominator for testing A would be the whole plot error (WPE) mean square.

But there is a second factor so the model actually is

$$y_{ijk} = \mu + \alpha_i + \eta_{k(i)} + \beta_j + \alpha\beta_{ij} + \varepsilon_{jk(i)}$$

What would a Hasse diagram for this design look like?



The denominator for testing A is again the whole plot error (WPE) mean square because that is the leading eligible random term below A.

Similarly the denominator for testing B and AB is the split plot error (SPE) mean square. Again it is the leading eligible random term below B and AB.

Example

Here is an example based on data from Steele and Torrie for the experiment studying protectants with oats seed sources.

At the whole plot level, what I described is a CRD, with blocks as EU's.

There is no reason you can't have a fancier design. In fact, in this experiment, as is very common, the whole plots themselves were grouped in super blocks of size a = 4, with the whole plot factor levels assigned in a RCB.

```
Cmd> data <- read("","sandt")
sandt    64      4 format
) Split plot data from Steele & Torrie
) Experiment to study effects of 4 protectants on
) oats grown from 4 seed sources.
) Seed source was whole plot factor, arranged in 4 randomized
) blocks (replicates).  Protectant was split plot factor,
) all 4 levels in each whole plot
) Col. 1: Block number (1 - 4)
) Col. 2: Seed lot (1 - 4)
) Col. 3: Protectant (1 - 4)
) Col. 4: Yield (response)
Read from file "TP1:Stat5303:Displays:sandt.dat"

Cmd> makecols(data,block,seed,protectant,y)

Cmd> block <- factor(block);seed <- factor(seed)

Cmd> protectant <- factor(protectant)

Cmd> anova("y=blocks+seed + E(blocks.seed) + protectant +
seed.protectant",fstat:T)
Model used is y=blocks+seed + E(blocks.seed) + protectant +
seed.protectant
                DF         SS         MS          F     P-value
CONSTANT         1  1.7849e+05  1.7849e+05  2598.06040          0
blocks           3      2842.9     947.62    13.79378   0.0010287
seed             3        2848     949.34    13.81877    0.001022
ERROR1           9      618.29     68.699     3.38234   0.0042283
protectant       3      170.54     56.846     2.79874    0.053859
seed.
 protectant      9      586.47     65.163     3.20823   0.0059453
ERROR2          36       731.2     20.311
```

The seed source main effect is highly significant as is the interaction of seed by protectant.

Here is the first use of a very useful feature of MacAnova.

If you "wrap" a term in `E(...)` (here `E(blocks.seed)`), `anova()` treats the term as an error term. It is named `ERROR1` and is the whole plot error. The final error term is named `ERROR2` and is the split plot error term. If you had more than one `E(...)`, terms would be `ERROR1`, `ERROR2`, `ERROR3`, ...

The denominator for each F is the next `ERRORx` mean square.

Here is the same ANOVA without `E()`:

```
Cmd> anova("y=blocks*seed + protectant + seed.protectant",\
     fstat:T)
Model used is y=blocks*seed + protectant + seed.protectant
               DF          SS          MS           F     P-value
CONSTANT        1  1.7849e+05  1.7849e+05  8787.53084  1.2987e-44
blocks          3      2842.9      947.62    46.65531  1.7295e-12
seed            3        2848      949.34    46.73981  1.6855e-12
blocks.seed     9      618.29      68.699     3.38234   0.0042283
protectant      3      170.54      56.846     2.79874    0.053859
seed.protectant 9      586.47      65.163     3.20823   0.0059453
ERROR1         36       731.2      20.311
```

The SS are the same. The F-statistics for `blocks` and `seed` are different (and incorrect), since they use what we know to be the split plot error as denominator.