Displays for Statistics 5303

Lecture 41

December 13, 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 Fractional Factorial Designs

An important buzzword (vocabulary) is the **resolution** of a fractional factorial design.

The higher the resolution of a design the deeper in the well of interactions you can go without finding effects that are aliased with main effects of low order interactions.

Recall

A 2^{k-p} fractional factorial design is defined by a set of p *defining relations* of the form typified by

$$I = ABCF, I = -BCDG$$

or, more compactly

$$I = ABCF = -BCDG$$

These pick out the quarter replicate of treatment combinations with the ABCF contrast having +1 and the BCDG contrast having -1.

These p defining relationships are the *generators* of the design, because from them you can find additional $2^p - 1 - p$ defining relationships, yielding $2^p - 1$ in all.

The additional defining relations are derived from the generators by finding all generalized products of pairs, triples, quadruples, etc. of the generators.

Small example:

2^{4-2} with generators

	1	2
	I = -AB = CD:	
1×2	I = -ABCD	

Example: 2^{8-4}, A 1/16 fraction of a 2^8 with defining relations generated by

	1	2	3	4
I =	BCDE	= ACDF	= ABDG	= ABCH

1×2	BCDE×ACDF = ABEF
1×3	BCDE×ABDG = ACEG
1×4	BCDE×ABCH = ADEH
2×3	ACDF×ABDG = BCFG
2×4	ACDF×ABCH = BDFH
3×4	ABDG×ABCH = CDGH
1×2×3	ABEF×ABDG = DEFG
1×2×4	ABEF×ABCH = CEFH
1×3×4	ACEG×ABCH = BEGH
2×3×4	BCFG×ABCH = AFGH
1×2×3×4	ABEF×CDGH = ABCDEFGH

These are summarized by

I = BCDE = ACDF = ABDG = ABCH= ABEF = ACEG = ADEH = BCFG = BDFH = CDGH = DEFG = CEFH = BEGH = AFGH = ABCDEFGH, $2^4-1 = 15$ sets of letters in all.

If any generators have minus signs, the signs propagate as with normal products.

In MacAnova, you specify a set of generators by a basis matrix with p rows and k columns of 0's and ±1's (all that matters is whether there are an even or odd number of minus signs).

```
Cmd> basis <- matrix(vector(0,1,1,1,1,0,0,0, 1,0,1,1,0,1,0,0,\
       1,1,0,1,0,0,1,0, 1,1,1,0,0,0,0,1),8)'

Cmd> print(basis,format:"3.0f")
basis:
(1,1)   0   1   1   1   1   0   0   0     BCDE
(2,1)   1   0   1   1   0   1   0   0     ACDF
(3,1)   1   1   0   1   0   0   1   0     ABDG
(4,1)   1   1   1   0   0   0   0   1     ABCH
```

`aliases2(basis)` finds all defining contrasts.

```
Cmd> aliases2(basis)
 (1) "I"
 (2) "BCDE"
 (3) "ACDF"
 (4) "ABEF"
 (5) "ABDG"
 (6) "ACEG"
 (7) "BCFG"
 (8) "DEFG"
 (9) "ABCH"
(10) "ADEH"
(11) "BDFH"
(12) "CEFH"
(13) "CDGH"
(14) "BEGH"
(15) "AFGH"
(16) "ABCDEFGH"
```


Set one +1 to -1 in each of rows 2 and 3:

```
Cmd> basis1 <- basis; basis1[2,1] <- basis1[3,1] <- -1

Cmd> print(basis1,format:"3.0f")
basis1:
(1,1)   0   1   1   1   1   0   0   0     BCDE
(2,1)  -1   0   1   1   0   1   0   0    -ACDF
(3,1)  -1   1   0   1   0   0   1   0    -ABDG
(4,1)   1   1   1   0   0   0   0   1     ABCH
```

```
Cmd> aliases2(basis1)
 (1) "I"
 (2) "BCDE"
 (3) "-ACDF"
 (4) "-ABEF"
 (5) "-ABDG"
 (6) "-ACEG"
 (7) "BCFG"
 (8) "DEFG"
 (9) "ABCH"
(10) "ADEH"
(11) "-BDFH"
(12) "-CEFH"
(13) "-CDGH"
(14) "-BEGH"
(15) "AFGH"
(16) "ABCDEFGH"
```

You can get all $2^p - 1$ aliases of any effect by generalized multiplication of that effect by the complete set of defining relations.

For example, multiplying the generalized relations by C

I = BCDE = ACDF = ABDG = ABCH = ABEF = ACEG = ADEH = BCFG = BDFH = CDGH = DEFG = CEFH = BEGH = AFGH = ABCDEFGH

you get the aliases of C:

C = BDE = ADF = ABCDG = ABH = ABCEF = AEG = ACDEH = BFG = BCDFH = DGH = CDEFG = EFH = CBEGH = ACFGH = ABDEFGH

You can use **aliases2()** to accomplish this in MacAnova using keyword **effect**:

```
Cmd> aliases2(basis,effect:vector(0,0,1,0,0,0,0,0))#C's aliases
 (1) "C"
 (2) "BDE"
 (3) "ADF"
 (4) "ABCEF"
 (5) "ABCDG"
 (6) "AEG"
 (7) "BFG"
 (8) "CDEFG"
 (9) "ABH"
(10) "ACDEH"
(11) "BCDFH"
(12) "EFH"
(13) "DGH"
(14) "BCEGH"
(15) "ACFGH"
(16) "ABDEFGH"
```

If C was in a defining relation, it drops out of the alias; otherwise it gets added.

So in this case, because all the defining relations except I = ABCDEFG involve 4 letters, C is aliased with 3 letter effects (if C in 4 letter defining relation), 5 letter effects (if C *not* in 4 letter defining relation) or 7 letter effects (because C not in 8 letter defining relation).

The generalized product of a 2 letter effect such as AB and R letter effect can be R-2, R or R+2 depending on whether both letters are in the effect (ABEF), just 1 letter (ACDF) or 0 letters (CDEF).

So the product of a 2 letter effect and a R = 4 letter effect can be have 2, 4 or 6 letters.

```
Cmd> aliases2(basis,effect:vector(1,0,1,0,0,0,0,0))
 (1) "AC"
 (2) "ABDE"       ACxBCDE
 (3) "DF"         ACxACDF
 (4) "BCEF"       ACxABEF
 (5) "BCDG"       ACxABDG
 (6) "EG"         ACxACEG
 (7) "ABFG"       ACxBCFG
 (8) "ACDEFG"     ACxDEFG
 (9) "BH"         ACxABCH
(10) "CDEH"       ACxADEH
(11) "ABCDFH"     ACxBDFH
(12) "AEFH"       ACxCEFH
(13) "ADGH"       ACxCDGH
(14) "ABCEGH"     ACxBEGH
(15) "CFGH"       ACxAFGH
(16) "BDEFGH"     ACxABCDEFGH
```

In general, the product of a J letter effect and a R letter effect, with $R \geq J$ can have R - J, R - J + 2, ..., R + J letters.

The **Resolution** R of a design is the minimum number of letters in any defining relation.

The preceding 2^{8-4} design with 14 defining relations with 4 letters and one with 8 is of resolution 4 or in Roman numerals, IV. It is a 2_{IV}^{8-4} design.

With R = II (2):

- Main effects (J = 1) are aliased with main effects R - J = 1. *Never* useful.

With R = III (3),

- Main effects (J = 1) are aliased with interactions involving R-J = 2 or more letters but not with other main effects

With R = IV (4),

- Main effects (J = 1) are aliased with R-1 = 3-way interactions but not with main effects or 2-way interactions
- 2-way interactions (J = 2) are aliased with other R-J = 2-way interactions.

With R = V (5),

- Main effects (J = 1) are aliased with R-1 = 4-way interactions but not with main effects or 2 or 3-way interactions
- 2-way interactions (J = 2) are aliased with R-J = 3-way interactions but not with main effects or other 2-way interactions.

Thus, when any 2-way interaction may be non-negligible, you need at least a resolution V design.

However, if you believe only specific 2-way interactions, say AB, AC and BC, are non-negligible, you can use a resolution IV design in which treatments are assigned to letters in such a way that these are not aliased with each other.

What are the aliases of AB?

```
Cmd> aliases2(basis,effect:vector(1,1,0,0,0,0,0,0))
 (1) "AB"
 (2) "ACDE"
 (3) "BCDF"
 (4) "EF"
 (5) "DG"
 (6) "BCEG"
 (7) "ACFG"
 (8) "ABDEFG"
 (9) "CH"
(10) "BDEH"
(11) "ADFH"
(12) "ABCEFH"
(13) "ABCDGH"
(14) "AEGH"
(15) "BFGH"
(16) "CDEFGH"
```

AB is not aliased with AC or BC and we saw previously that AC was not aliased with BC, so this resolution IV design would be adequate if the only two-way interactions of interest were AB, AC and BC.

Ex. 18.4: Design a 2^{7-2} resolution IV factorial and show how you would block the principal fraction (fraction containing (1)) into two blocks of size 16.

First, look just at determining a 2^{7-2}_{IV} design (1/4 replicate, 32 treatments).

You can use MacAnova macro **choosegen2()**. This is similar to **choosedef2()** for finding good confounded designs, with the additional feature that you can specify a resolution you would be satisfied with.

```
Cmd> usage(choosegen2)
choosegen2(k,p, all:T [,res:r]), positive integers k, p, r.
choosegen2(k,p, tries:m [,res:r]), positive integers k, p, m,
and r.
```

```
Cmd> print(choosegen2(7,2,all:T),format:"3.0f")
STRUCTURE:
component: resolution
(1)   4
component: generators
(1) "BCDEF"
(2) "ACDEG"
component: aberration
(1)   0   0   0   1   2   0   0
component: basis
(1,1)   0   1   1   1   1   1   0
(2,1)   1   0   1   1   1   0   1
```

Defining relations are

I = BCDEF = ACDEG = ABFG

As with **choosedef2()**, for large k this can take some time. Keyword **tries** specifies a random search, with the best design returned. Often it is as good as the best found with **all:T** which examines all possible fractions.

```
Cmd> print(choosegen2(7,2,tries:1000),format:"3.0f")#1000 tries
STRUCTURE:
component: resolution
(1)   4
component: generators
(1) "ABDEF"
(2) "ACEG"
component: aberration
(1)   0   0   0   1   2   0   0
component: basis
(1,1)   1   1   0   1   1   1   0
(2,1)   1   0   1   0   1   0   1
```

These defining relations are

I=ABDEF=ACEG=BCDFG (= ABDEF×ACEG)

This looks different from the first design I = BCDEF = ACDEG = ABFG, but the two designs are really equivalent:

In the first design make the substitutions

A→A, B→C, C→B, D→D, E→G, F→E, G→F

and you get this second design.

The I = BCDEF = ACDEG = ABFG design is the best (minimum aberration) resolution IV design run in *one* block of size 32.

However, it's not the best resolution 4 design to split into *two* blocks of size 16.

To split it you would choose a contrast to confound between the blocks. And any contrast you choose will be aliased with three other contrasts.

You can check that any 4 letter contrast different from ABFG must have at least three letters in common with BCDEF and ACDEG, so at best you would confound a *two* way interaction.

You can do better than that. In fact you can find a resolution IV design split into two blocks with only 4-way interactions confounded with blocks.

First use **choosegen2()** to find a 2^{7-3} design (1/8 replicate with block size 16).

```
Cmd> stuff <- choosegen2(7,3,all:T)

Cmd> print(stuff,format:"3.0f")
STRUCTURE:
component: resolution
(1)   4
component: generators
(1) "BCDE"     First two define a 2^(7-2) design
(2) "ACDF"
(3) "ABDG"
component: aberration
(1)   0   0   0   7   0   0   0
component: basis
(1,1)   0   1   1   1   1   0   0
(2,1)   1   0   1   1   0   1   0
(3,1)   1   1   0   1   0   0   1
```

This is also a resolution 4 design.

It will be half of the 2^{7-2} design with the first two contrast as defining relations (I = BCDE = ACDF = ABEF (= BCDE×ACDF).

Now split the size 32 block of this 2^{7-2} design in blocks of size 16 using contrast ABDG which is aliased to ABDG×BCDE = ACEG, ABDG×ACDF = BCFG, and ABDG×ABEF = DEFG in the 2^{7-2} design, all 4 way interactions. Thus you have a resolution IV design with only 4-way interactions confounded between blocks.

Here are the treatment combinations in the 2^{7-2} design with defining relations I = BCDE = ACDF (use rows 1 and 2 of `stuff$basis`)

```
Cmd> print(paste(ffdesign2(stuff$basis[run(2),])))
(1) af be abef cef ace bcf abc def ade bdf abd cd acdf bcde
abcdef g agf bge abgef cgef acge bcgf abcg dgef adge bdgf abdg
cdg acdgf bcdge abcdgef
```

Here they are for the 2^{7-3} design for I = BCDE = ACDF = ABDG (use all 3 rows of `stuff$basis`):

```
Cmd> print(paste(ffdesign2(stuff$basis)))
(1) afg beg abef cef aceg bcfg abc defg ade bdf abdg cdg acdf
bcde abcdefg
```

These are the treatment combinations in block I. Pick one treatment, say g, in the 2^{7-2} design that is not in the 2^{7-3} design and "multiply" it by the treatments in the 2^{7-3} design. This gives you block II.

I	(1) afg beg abef cef aceg bcfg abc defg ade bdf abdg cdg acdf bcde abcdefg
II	e af be abefg cefg ace bcf abcg def adeg bdfg abd cd acdfg bcdeg abcdef