STAT 5309 –SP 2019

LAB 7

Contents:

- **Blocking in Factorial designs**
- **Fractional Factorial Designs : 2^{K-1} , $2^{K-2} - designs$**

Due: Thurs, Apr 11

A. PRACTICE

A1. Blocking in Factorial designs

An experiment was performed to improve the yield of a chemical process. Four factors were selected, and two replicates of a completely randomized experiment were run. The results are shown in the following table:

Treatment Combination	Replicate I	Replicate II	Treatment Combination	Replicate I	Replicate II
(1)	90	93	*d*	98	95
a	74	78	*ad*	72	76
b	81	85	*bd*	87	83
ab	83	80	*abd*	85	86
c	77	78	*cd*	99	90
ac	81	80	*acd*	79	75
bc	88	82	*bcd*	87	84
abc	73	70	*abcd*	80	80

Note: There are 4 factors, so a Full factorial design needs 16 runs. Here we have 2 replicates, so the total is 32 runs.

Now we need to create a factor named Block. Note: 1,3,5,7 belong to the Block 1, 2,4,6,8,.. belong to Block 2.

```
chem <- read.csv(file.choose(), header=TRUE)
chem

bl <- rep(1:2, times=16)
chem.bl <- data.frame(chem, bl)
```

```
> head(chem.bl)
   A  B  C  D Yield bl
1 -1 -1 -1 -1    90  1
2 -1 -1 -1 -1    93  2
3  1 -1 -1 -1    74  1
4  1 -1 -1 -1    78  2
5 -1  1 -1 -1    81  1
6 -1  1 -1 -1    85  2
```

```
attach(chem.bl)
```

chem.bl.aov <- aov(Yield ~ A*B*C*D + Error(bl))

```
> summary(chem.bl.aov)

Error: bl
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  1  11.28   11.28

Error: Within
          Df Sum Sq Mean Sq F value   Pr(>F)
A          1  657.0   657.0  88.613 1.10e-07 ***
B          1   13.8    13.8   1.859 0.192893
C          1   57.8    57.8   7.793 0.013690 *
D          1  124.0   124.0  16.728 0.000966 ***
A:B        1  132.0   132.0  17.807 0.000743 ***
A:C        1    3.8     3.8   0.510 0.486115
B:C        1    2.5     2.5   0.341 0.567713
A:D        1   38.3    38.3   5.163 0.038219 *
B:D        1    0.3     0.3   0.038 0.848193
C:D        1   22.8    22.8   3.072 0.100035
A:B:C      1  215.3   215.3  29.035 7.53e-05 ***
A:B:D      1  175.8   175.8  23.708 0.000204 ***
A:C:D      1    7.0     7.0   0.948 0.345596
B:C:D      1    7.0     7.0   0.948 0.345596
A:B:C:D    1   47.5    47.5   6.411 0.023008 *
Residuals 15  111.2     7.4
```

install.packages("Rcmdr")
library(Rcmdr)

chem.bl.red <- stepwise(chem.bl.aov)

A2. Fractional designs

A2.1 2^{K-1} (One-Half) Fractional designs

2^(k -1) fractional using FrF2() , package FrF2.

The FrF2 package has several functions for working with fractioned designs. The basic function is FrF2(). We can use it to construct designs using generators that we specify, or it can select a design for us when we specify number of runs and number of factors.
For example, we want it to find a design for 5 factors in sixteen runs. A Full factorial design requires 32 runs. If the generator is left off, FrF2 finds one that is optimal

Create 16-run 2^(5-1), with generator E = ABCD.

```
library(FrF2)
des.1<- FrF2(16,  5, generators ="ABCD", randomize =FALSE)
#5 factors, with 16 runs
```

```
> summary(des.1)
Call:
FrF2(16, 5, generators = "ABCD", randomize = FALSE)

Experimental design of type  FrF2.generators
16  runs

Factor settings (scale ends):
   A  B  C  D  E
1 -1 -1 -1 -1 -1
2  1  1  1  1  1

Design generating information:
$legend
[1] A=A B=B C=C D=D E=E

$generators
[1] E=ABCD


Alias structure:
[[1]]
[1] no aliasing among main effects and 2fis


The design itself:
    A  B  C  D  E
1  -1 -1 -1 -1  1
2   1 -1 -1 -1 -1
3  -1  1 -1 -1 -1
4   1  1 -1 -1  1
5  -1 -1  1 -1 -1
6   1 -1  1 -1  1
7  -1  1  1 -1  1
8   1  1  1 -1 -1
9  -1 -1 -1  1 -1
10  1 -1 -1  1  1
11 -1  1 -1  1  1
12  1  1 -1  1 -1
13 -1 -1  1  1  1
14  1 -1  1  1 -1
15 -1  1  1  1 -1
16  1  1  1  1  1
class=design, type= FrF2.generators
```

A 2^{K-1} is called **One- Half Factorial Design**.

We can supply **factor names** if you don't like A, B, C, etc.
Moe Larry Curly

FrF2(4, 3, factor.names=c("Moe","Larry","Curly"))

```
  Moe Larry Curly
1  -1     1    -1
2   1     1     1
3   1    -1    -1
4  -1    -1     1
class=design, type= FrF2
```

We can also name **the levels** by using a list of factor names, each of which contains the level names.

FrF2(4,3,factor.names=list(speed=c("fast","slow"),temp=c("hot","cold"), time=c("long","short")))

```
  speed temp  time
1  slow  hot  long
2  fast  hot short
3  slow cold short
4  fast cold  long
class=design, type= FrF2
```

We can extract the design information

design.info(des.1)

```
$generators
[1] "E=ABCD"

$aliased
$aliased$legend
[1] "A=A" "B=B" "C=C" "D=D" "E=E"

$aliased[[2]]
[1] "no aliasing among main effects and 2fis"
```

Note: Most of these designs come from a catalog of designs. Some interesting bits include the resolution (here V), and the generating columns.

Note: A 2^{n-k} design is set up as Full factorial design in the first (*n – k)* factors. Each of the k additional factors is aliased to an interaction of the first (*n – k)* factors. The "**Generating columns" information tells us which columns are the generator**; These columns are the +1/–1 columns for the various interactions, and they are numbered in standard order beginning with A.

Note: ABCD interaction, so column E is generated by ABCD, E= ABCD. Multiplied both sides by E, with E^2= I, yielding the overall generator I=ABCDE, giving us resolution V.

A2.2 2^{K-2} (One-Fourth) Fractional Factorial Designs

Suppose we have 5 factors. Full Factorial needs 32 runs. Fractional Factorial 2^{K-1} requires 16 runs. Now we build a Fractional 2^{K-2} - Factorial requires only 8 runs.

```
des.2 <- FrF2(8,5)
des.2
```

```
   A  B  C  D  E
1 -1 -1  1  1 -1
2  1  1 -1  1 -1
3 -1  1  1 -1 -1
4 -1  1 -1 -1  1
5  1 -1 -1 -1 -1
6  1  1  1  1  1
7 -1 -1 -1  1  1
8  1 -1  1 -1  1
class=design, type= FrF2
```

Note: design that will have two generators.

```
design.info(des.2)$catlg.entry
```

```
Design:  5-2.1
   8  runs,  5  factors,
   Resolution  III
   Generating columns:  3 5
   WLP (3plus):  2 1 0 0 0 ,  0  clear 2fis
```

The generating columns are (AB) and (AC), so the aliasing is based on I = ABD = ACE = BCDE. In this case, **all two factor interactions are aliased to a main effect**, Just above in des.1, all 2-factor interactions were aliased with 3-factor interactions.

One-Fourth Fraction design, with 5 factors, and 2 generators

> FrF2(8, 5, generators=c("AC","BC"), randomize=FALSE)

```
   A  B  C  D  E
1 -1 -1 -1  1  1
2  1 -1 -1 -1  1
3 -1  1 -1  1 -1
4  1  1 -1 -1 -1
5 -1 -1  1 -1 -1
6  1 -1  1  1 -1
7 -1  1  1 -1  1
8  1  1  1  1  1
class=design, type= FrF2.generators
```

(Here we ask for a design with generators ACD and BCE, so ABDE is also aliased.)

One-Fourth Fraction design, with 5 factors, and 2 generators AC, -BC

FrF2(8,generators=c("AC","-BC"),randomize=FALSE)

```
   A  B  C  D  E
1 -1 -1 -1  1 -1
2  1 -1 -1 -1 -1
3 -1  1 -1  1  1
4  1  1 -1 -1  1
5 -1 -1  1 -1  1
6  1 -1  1  1  1
7 -1  1  1 -1 -1
8  1  1  1  1 -1
class=design, type= FrF2.generators
```

2^{K-4} *Fractional Design*

des.3 <- FrF2(16,8,randomize=FALSE);des3
Here is a more interesting design, 2^{K-4} , which is K =8 factors in 16 runs.

design.info(des.3)$catlg.entry

```
Design:  8-4.1
   16  runs,  8  factors,
   Resolution  IV
   Generating columns:  7 11 13 14
   WLP (3plus):  0 14 0 0 0 ,  0  clear 2fis
```

```
    A  B  C  D  E  F  G  H
1  -1 -1 -1 -1 -1 -1 -1 -1
2   1 -1 -1 -1  1  1  1 -1
3  -1  1 -1 -1  1  1 -1  1
4   1  1 -1 -1 -1 -1  1  1
5  -1 -1  1 -1  1 -1  1  1
6   1 -1  1 -1 -1  1 -1  1
7  -1  1  1 -1 -1  1  1 -1
8   1  1  1 -1  1 -1 -1 -1
9  -1 -1 -1  1 -1  1  1  1
10  1 -1 -1  1  1 -1 -1  1
11 -1  1 -1  1  1 -1  1 -1
12  1  1 -1  1 -1  1 -1 -1
13 -1 -1  1  1  1  1 -1 -1
14  1 -1  1  1 -1 -1  1 -1
15 -1  1  1  1 -1 -1 -1  1
16  1  1  1  1  1  1  1  1
class=design, type= FrF2
```

data.2 <- data.frame(des.2, y<-rnorm(8))
data.2.lm <- lm(y ~ A*B*C*D, data=data.2)

The FrF2 function **alias()** print the alias structure previously constructed. This function requires a response vector.

library(FrF2)
des.4 <- FrF2(16, 5)   # 5 factors, Half Fractional design
y <- runif(16, 0,1)   # and a vector y, a vector of random uniform numbers.
des.4 <- data.frame(des.4, y)
aliases(lm(y~(.)^5, data = des.4))   #give the alias relationships

```
A = B:C:D:E
 B = A:C:D:E
 C = A:B:D:E
 D = A:B:C:E
 E = A:B:C:D
 A:B = C:D:E
 A:C = B:D:E
 A:D = B:C:E
 A:E = B:C:D
 B:C = A:D:E
 B:D = A:C:E
 B:E = A:C:D
 C:D = A:B:E
 C:E = A:B:D
 D:E = A:B:C
```

Note: main effects aliased with 4-way interactions; 2-way interactions aliased with 3-way interactions

##----**add.response() :** function from the DoE.base package to include the response.

#After adding the response, the model(mod1) was fit to the data using R function lm()

library(FrF2)
soup <- FrF2(16,5, generators = "ABCD", factor.names= list(Ports =c(1,3), Temp=c("Cool", "Ambient"), MixTime=c(60,80),BatchWt=c(1500,2000), delay=c(7,1)) ,randomize=FALSE)

a design with 5 factors, in 16 runs (Half Factorial)

```
   Ports    Temp MixTime BatchWt delay    y
1      1    Cool      60    1500     1 1.13
2      3    Cool      60    1500     7 1.25
3      1 Ambient      60    1500     7 0.97
4      3 Ambient      60    1500     1 1.70
5      1    Cool      80    1500     7 1.47
6      3    Cool      80    1500     1 1.28
7      1 Ambient      80    1500     1 1.18
8      3 Ambient      80    1500     7 0.98
9      1    Cool      60    2000     7 0.78
10     3    Cool      60    2000     1 1.36
11     1 Ambient      60    2000     1 1.85
12     3 Ambient      60    2000     7 0.62
13     1    Cool      80    2000     1 1.09
14     3    Cool      80    2000     7 1.10
15     1 Ambient      80    2000     7 0.76
16     3 Ambient      80    2000     1 2.10
class=design, type= FrF2.generators
```

y <- c(1.13,1.25,.97,1.70,1.47,1.28,1.18,.98,.78,1.36,1.85,.62,1.09,1.1,.76,2.10)
soup <-add.response(soup, y) # add response column to the design
library(DoE.base)

soup.lm <- lm(y ~ (.)^2,data=soup)   #linear model with only 2-way interactions
summary(soup.lm)

```
Call:
lm.default(formula = y ~ (.)^2, data = soup)

Residuals:
ALL 16 residuals are 0: no residual degrees of freedom!

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         1.22625         NA      NA       NA
Ports1              0.07250         NA      NA       NA
Temp1               0.04375         NA      NA       NA
MixTime1            0.01875         NA      NA       NA
BatchWt1           -0.01875         NA      NA       NA
delay1              0.23500         NA      NA       NA
Ports1:Temp1        0.00750         NA      NA       NA
Ports1:MixTime1     0.04750         NA      NA       NA
Ports1:BatchWt1     0.01500         NA      NA       NA
Ports1:delay1       0.07625         NA      NA       NA
Temp1:MixTime1     -0.03375         NA      NA       NA
Temp1:BatchWt1      0.08125         NA      NA       NA
Temp1:delay1        0.20250         NA      NA       NA
MixTime1:BatchWt1   0.03625         NA      NA       NA
MixTime1:delay1    -0.06750         NA      NA       NA
BatchWt1:delay1     0.15750         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,    Adjusted R-squared:    NaN
F-statistic:   NaN on 15 and 0 DF,  p-value: NA
```

Note: P-values are not calculated

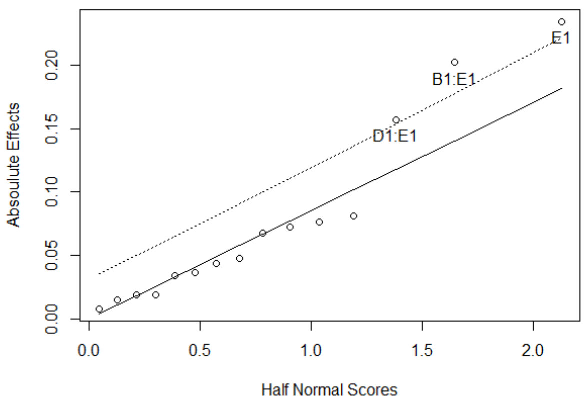
-----------PLOTTING: NORMAL PLOT OF EFFECTS

design <- FrF2(16, 5, generators="ABCD", randomize=FALSE)
soup <- add.response(design, y)

```
    A  B  C  D  E    y
1  -1 -1 -1 -1  1 1.13
2   1 -1 -1 -1 -1 1.25
3  -1  1 -1 -1 -1 0.97
4   1  1 -1 -1  1 1.70
5  -1 -1  1 -1 -1 1.47
6   1 -1  1 -1  1 1.28
7  -1  1  1 -1  1 1.18
8   1  1  1 -1 -1 0.98
```

```
9   -1 -1 -1   1 -1 0.78
10   1 -1 -1   1  1 1.36
11 -1   1 -1   1  1 1.85
12   1   1 -1   1 -1 0.62
13 -1 -1   1   1  1 1.09
14   1 -1   1   1 -1 1.10
15 -1   1   1   1 -1 0.76
16   1   1   1   1  1 2.10
class=design, type= FrF2.generators
```

```
soup.lm <- lm( y ~ (.)^2, data=soup)
library(daewr)
LGB(coef(soup.lm)[-1],  rpt = FALSE)
```



##----------More on Fractional Factorial: ----------------------

#8runs, 6 factors.

```
des.5<- FrF2(8,6, generators=c("AB", "AC", "BC"))    # generators mentioned
des.5
```

```
> summary(des.5)
Call:
FrF2(8, 6, generators = c("AB", "AC", "BC"))

Experimental design of type  FrF2.generators
8  runs

Factor settings (scale ends):
   A  B  C  D  E  F
1 -1 -1 -1 -1 -1 -1
2  1  1  1  1  1  1
```

```
Design generating information:
$legend
[1] A=A B=B C=C D=D E=E F=F

$generators
[1] D=AB E=AC F=BC


Alias structure:
$main
[1] A=BD=CE B=AD=CF C=AE=BF D=AB=EF E=AC=DF F=BC=DE

$fi2
[1] AF=BE=CD
```

#----------FrF2() automatic selects the generators that results in a minimum design

des.6 <- FrF2(16, 8)   # 8 factors 2-level; 16runs.   # generators not mentioned

```
    A  B  C  D  E  F  G  H
1   1  1 -1 -1 -1 -1  1  1
2   1 -1  1 -1 -1  1 -1  1
3  -1  1  1  1 -1 -1 -1  1
4  -1  1  1 -1 -1  1  1 -1
5   1  1  1  1  1  1  1  1
6   1 -1  1  1 -1 -1  1 -1
7   1  1 -1  1 -1  1 -1 -1
8   1 -1 -1  1  1 -1 -1  1
9   1 -1 -1 -1  1  1  1 -1
10 -1 -1  1 -1  1 -1  1  1
11 -1  1 -1  1  1 -1  1 -1
12 -1 -1 -1  1 -1  1  1  1
13 -1  1 -1 -1  1  1 -1  1
14 -1 -1  1  1  1  1 -1 -1
15  1  1  1 -1  1 -1 -1 -1
16 -1 -1 -1 -1 -1 -1 -1 -1
class=design, type= FrF2
```

y<- runif(16,0,1)
library(DoE.base)
generators(des.5)
aliases(lm(y~ (.)^3, data=des.5))

```
> aliases( lm( y~ (.)^3, data=des.5))

 A = B:D = C:E = B:E:F = C:D:F
 B = C:F = A:E:F = C:D:E = A:D
 C = B:F = A:D:F = B:D:E = A:E
 D = E:F = A:C:F = B:C:E = A:B
 E = D:F = A:B:F = B:C:D = A:C
```

```
F = B:C = D:E = A:B:E = A:C:D
A:F = B:E = C:D = A:B:C = A:D:E = B:D:F = C:E:F
```

B. EXERCISE

1. Problem 6-1 , Excel dataset

An engineer is interested in the effects of cutting speed (A), tool geometry (B), and cutting angle (C) on the life (in hours) of a machine tool. Two levels of each factor are chosen, and three replicates of a 2^3 factorial design are run. The results follow:

A	B	C	Treatment Combination	Replicate I	Replicate II	Replicate III
−	−	−	(1)	22	31	25
+	−	−	*a*	32	43	29
−	+	−	*b*	35	34	50
+	+	−	*ab*	55	47	46
−	−	+	*c*	44	45	38
+	−	+	*ac*	40	37	36
−	+	+	*bc*	60	50	54
+	+	+	*abc*	39	41	47

Consider the experiment described in Problem 6-1. Analyze this experiment assuming that each replicate represents a block of a single production shift.

2. Problem 6-5, Excel dataset

A router is used to cut locating notches on a printed circuit board. The vibration level at the surface of the board as it is cut is considered to be a major source of dimensional variation in the notches. Two factors are thought to influence vibration: bit size (A) and cutting speed (B). Two bit sizes ($\frac{1}{16}$ and $\frac{1}{8}$ inch) and two speeds (40 and 90 rpm) are selected, and four boards are cut at each set of conditions shown below. The response variable is vibration measured as the resultant vector of three accelerometers (x, y, and z) on each test circuit board.

			Replicate			
A	*B*	Treatment Combination	I	II	III	IV
−	−	(1)	18.2	18.9	12.9	14.4
+	−	*a*	27.2	24.0	22.4	22.5
−	+	*b*	15.9	14.5	15.1	14.2
+	+	*ab*	41.0	43.9	36.3	39.9

Consider the experiment described in Problem 6-5. Analyze this experiment assuming that each one of the four replicates represents a block.

3. Problem

(a) Use FrF2() to create a design of 16 runs, use 8 factors 2-level: A,B,C,D, E, F, G, H. Use generators: “ABC", “ABD", “ACD" and "BCD".

(b) Add response: y1 <- c(5.75 , 6.7, 11.2, 10.67, 4.92, 5.35, 2.81, 10.83, 6.08, 7.27, 9.68, 4.2, 3.9, 3.78, 11.57, 7.39).
Build a AOV model, use up to 3-way interaction terms.

(c) Find the generators and alias structure.

(d) Plot a (d1) Main effects plot (d2) Effects Interaction plot (d3) Half-Normal Plot of Effects (d4) LenthPlot of Effects
What conclusions can be made about these plots.

4. [Problem 6-15, Excel dataset]

Suppose that in Problem 6-15, only a one-half fraction of the 2^4 design could be run. Construct the design and perform the analysis, using the data from replicate I.

A	B	C	D	Treatment Combination	Replicate I	Replicate II
−	−	−	−	(1)	7.037	6.376
+	−	−	−	a	14.707	15.219
−	+	−	−	b	11.635	12.089
+	+	−	−	ab	17.273	17.815
−	−	+	−	c	10.403	10.151
+	−	+	−	ac	4.368	4.098
−	+	+	−	bc	9.360	9.253
+	+	+	−	abc	13.440	12.923
−	−	−	+	d	8.561	8.951
+	−	−	+	ad	16.867	17.052
−	+	−	+	bd	13.876	13.658
+	+	−	+	abd	19.824	19.639
−	−	+	+	cd	11.846	12.337
+	−	+	+	acd	6.125	5.904
−	+	+	+	bcd	11.190	10.935
+	+	+	+	$abcd$	15.653	15.053