Printout

Friday, March 17, 2023 1:47 PM

Section 1

MANE 6313

Subsection 1

Week 10, Module B

Student Learning Outcome

- Select an appropriate experimental design with one or more factors,
- Select an appropriate model with one or more factors,
- Evaluate statistical analyses of experimental designs,
- Assess the model adequacy of any experimental design, and
- Interpret model results.

Module Learning Outcome

Analyze a one-half fraction factorial design using R.

Example Problem

20.17 A 2^{4-1} fractional factorial design was conducted on a chemical process by assigning variable 4 to the 123 interaction column.

Variable	Low Level	High Level
1. Feedrate (liters/min)	5	20
2. Catalyst (%)	A	B
3. Temperature (°C)	200	220
4. Concentration (%)	5	7

The second table summarizes the eight tests that were run, including the levels of each of the four variables and the yield (% reacted) for each test.

Test	1	2	3	4	y (% reacted)
1	−	−	−	−	33
2	+	−	−	+	51
3	−	+	−	+	44
4	+	+	−	−	40
5	−	−	+	+	35
6	+	−	+	−	82
7	−	+	+	−	46
8	+	+	+	+	69

(a) Write down all of the linear combinations of effects that can be estimated from this experiment (assume that third- and higher-order interactions are negligible).

(b) Calculate numerical values for the effect estimates and determine which effects are significant using a normal probability plot of the effect estimates.

Design

```r
## Module B
```{r}
# create design
library(FrF2)
p20_17 <- FrF2(8,4,randomize=FALSE)
summary(p20_17)
```
```

R Console

data.frame
2 x 4

data.frame
8 x 4

	A <fctr>	B <fctr>	C <fctr>	D <fctr>
1	-1	-1	-1	-1
2	1	-1	-1	1
3	-1	1	-1	1
4	1	1	-1	-1
5	-1	-1	1	1
6	1	-1	1	-1
7	-1	1	1	-1
8	1	1	1	1

8 rows

Figure 2: Experimental Design

Reviewing Design

```
12 ## Module B
13 ```{r}
14 # create design
15 library(FrF2)
16 p20_17 <- FrF2(8,4,randomize=FALSE)
17 summary(p20_17)
18 ```
```

R Console | data.frame 2 x 4 | data.frame 8 x 4

lengths

```
Call:
FrF2(8, 4, randomize = FALSE)

Experimental design of type  FrF2
8  runs

Factor settings (scale ends):

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

$generators
[1] D=ABC

Alias structure:
$fi2
[1] AB=CD AC=BD AD=BC
```

defining Relation → I = ABCD

A: A = BCD

B: AB = CD

$(4) = \frac{4!}{2!(4-2)!} = \frac{4 \cdot 3}{2} = 6$

Adding Response Variable

```r
# create and add response
y <- c(33,51,44,40,35,82,46,69)
library(DoE.base)
p20_17 <- add.response(p20_17,y)
print(p20_17)
```

data.frame 8 x 5 | R Console

	A <fctr>	B <fctr>	C <fctr>	D <fctr>	y <dbl>
1	-1	-1	-1	-1	33
2	1	-1	-1	1	51
3	-1	1	-1	1	44
4	1	1	-1	-1	40
5	-1	-1	1	1	35
6	1	-1	1	-1	82
7	-1	1	1	-1	46
8	1	1	1	1	69

8 rows

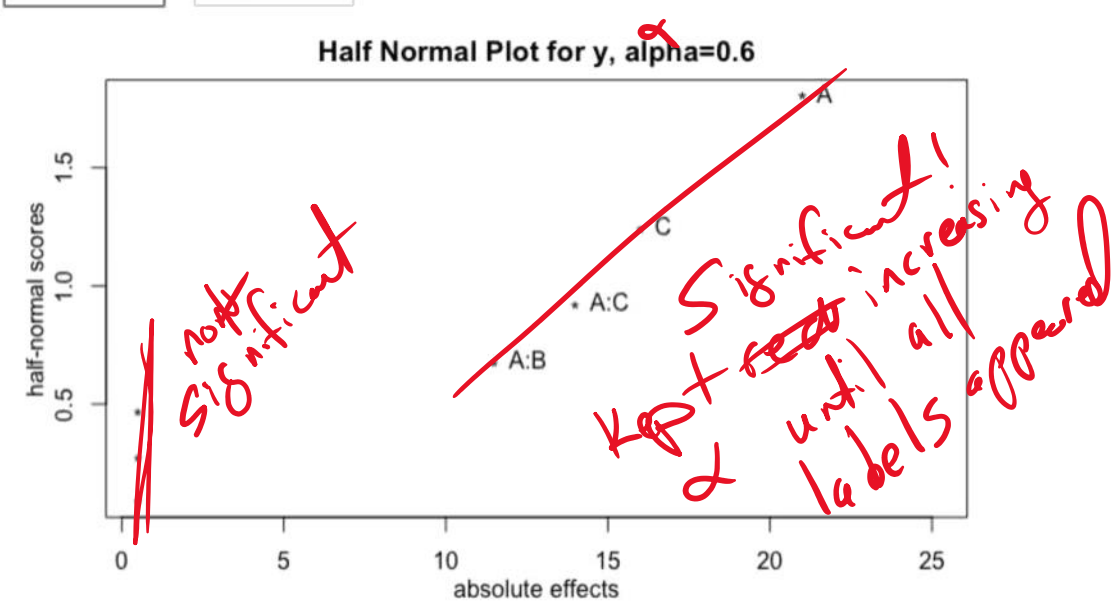
Figure 4: Adding Response Variable

Half-Normal Plot

You don't need a model to generate half-normal plots

```r
# Daniel Plot
DanielPlot(p20_17,half=TRUE,response='y',alpha=0.6)
```

R Console



not significant

Significant! Kept increasing until all labels appeared

Model

A:B but it could be CD interaction due to aliasing

```r
p20_17.model1=aov(y~A+B+C+A:B+A:C,data=p20_17)
summary(p20_17.model1)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)
A            1  882.0   882.0    1764 0.000566 ***
B            1    0.5     0.5       1 0.422650
C            1  512.0   512.0    1024 0.000975 ***
A:B          1  264.5   264.5     529 0.001885 **
A:C          1  392.0   392.0     784 0.001273 **
Residuals    2    1.0     0.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 6: Model Fitting

Interaction Plot

```r
# Interaction plot
IAPlot(p20_17,show.alias=TRUE)
```

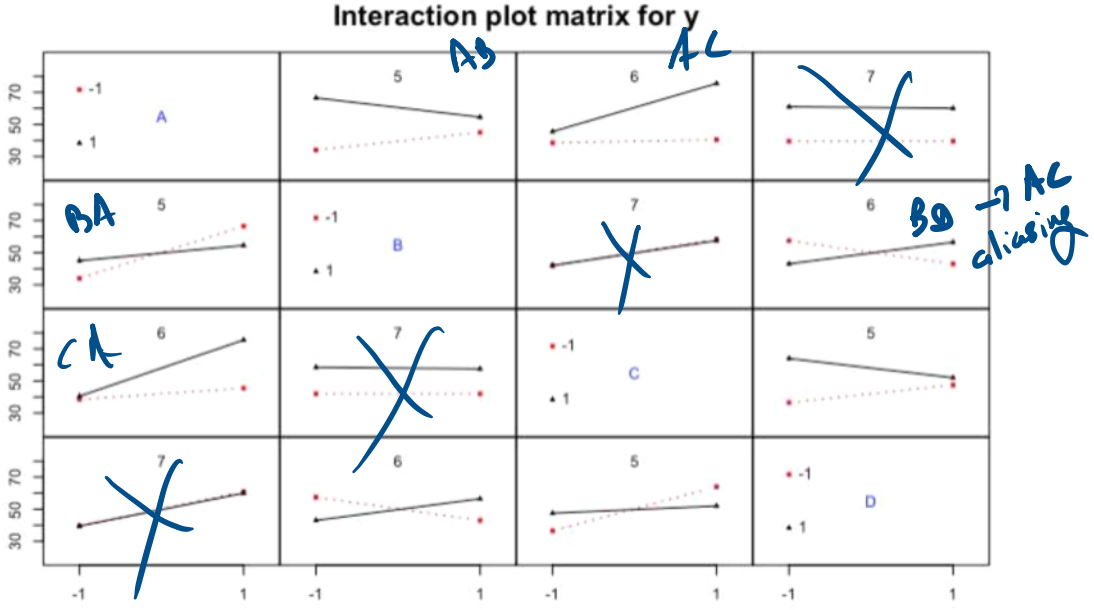


Figure 7: Interaction Plot

R Demonstration