Printout

Friday, March 17, 2023 3:28 PM

Section 1

MANE 6313

Subsection 1

Week 10, Module C

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 using advanced FrF2 features.

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

```
## Module C - Fancy
# create design
```{r}
library(FrF2)
fancy.design <- FrF2(8,4,factor.names=list(FeedRate=c(5,20),Catalyst=c("A","B"),Temperature=c(200,220),Concentration=c(5,7)),randomize=FALSE)
summary(fancy.design)
```
```

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

	FeedRate <fctr>	Catalyst <fctr>	Temperature <fctr>	Concentration <fctr>
1	5	A	200	5
2	20	A	200	7
3	5	B	200	7
4	20	B	200	5
5	5	A	220	7
6	20	A	220	5
7	5	B	220	5
8	20	B	220	7

8 rows

Figure 2: Design

Reviewing Design

```
## Module C - Fancy
# create design
```{r}
library(FrF2)
fancy.design <- FrF2(8,4,factor.names=list(FeedRate=c(5,20),Catalyst=c("A","B"),Temperature=c(200,220),Concentration=c(5,7)),randomize=FALSE)
summary(fancy.design)
```
```

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

```
Call:
FrF2(8, 4, factor.names = list(FeedRate = c(5, 20), Catalyst = c("A",
    "B"), Temperature = c(200, 220), Concentration = c(5, 7)),
    randomize = FALSE)

Experimental design of type  FrF2
8  runs

Factor settings (scale ends):

Design generating information:
$legend
[1] A=FeedRate      B=Catalyst      C=Temperature   D=Concentration

$generators
[1] D=ABC


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


The design itself:
class=design, type= FrF2
```

Figure 3: Design Details

Adding Response Variable

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

data.frame
8 x 5

R Console

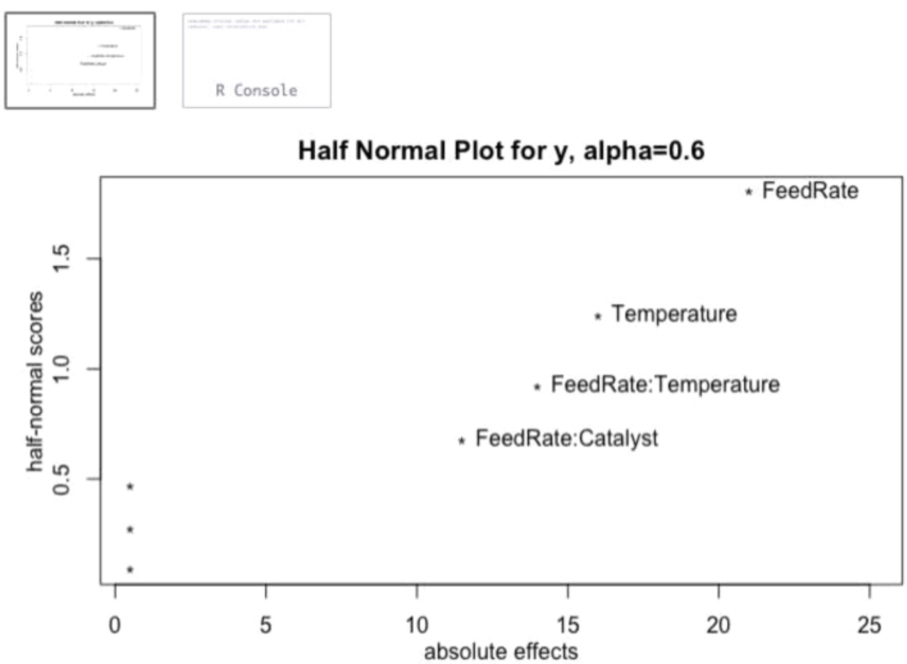
	FeedRate <fctr>	Catalyst <fctr>	Temperature <fctr>	Concentration <fctr>	y <dbl>
1	5	A	200	5	33
2	20	A	200	7	51
3	5	B	200	7	44
4	20	B	200	5	40
5	5	A	220	7	35
6	20	A	220	5	82
7	5	B	220	5	46
8	20	B	220	7	69

8 rows

Figure 4: Adding Response Variable

Half-Normal Plot

```
```{r}
# Daniel Plot
DanielPlot(fancy.design,half=TRUE,response='y',alpha=0.6)
```
```



Model

```
```{r}
fancy.model1=aov(y~FeedRate+Temperature+Catalyst+FeedRate:Temperature+FeedRate:Catalyst,data=fancy.design)
summary(fancy.model1)
```
```

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

Figure 6: Model

Interaction Plot

```{r}
```{r}
# Interaction plot
IAPlot(fancy.design,show.alias=TRUE)
```
```

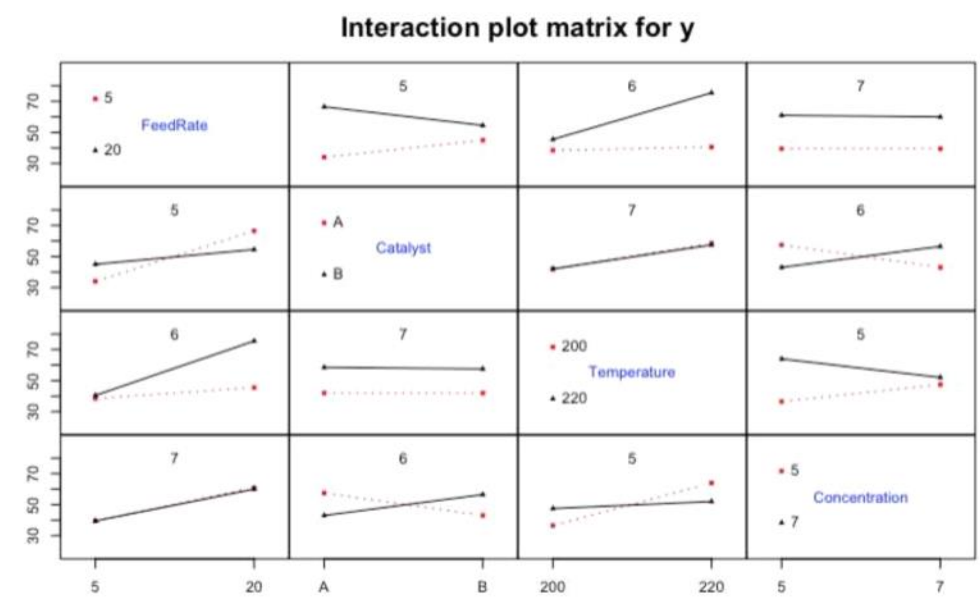


Figure 7: Interaction Plot

Homework/Project Guidelines

- Always use advanced features to avoid variable names A, B, C, etc.