Section 1

MANE 6313

Subsection 1

Week 11, Module D

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

Describe sequential analysis and full fold-over design.

Sequential Experimentation

- This is particularly applicable for resolution III experiments
- Conduct another fraction based upon information gained in the initial model
- A fold-over design can be either a full fold-over or fold-over on an effect

Example Problem

- Taken from Devore, Change and Sutherland
- 2^{7-4} resolution III design
- Generators are 4=12, 5=13, 6=23 and 7=123 (DCS uses numbers instead of letters)

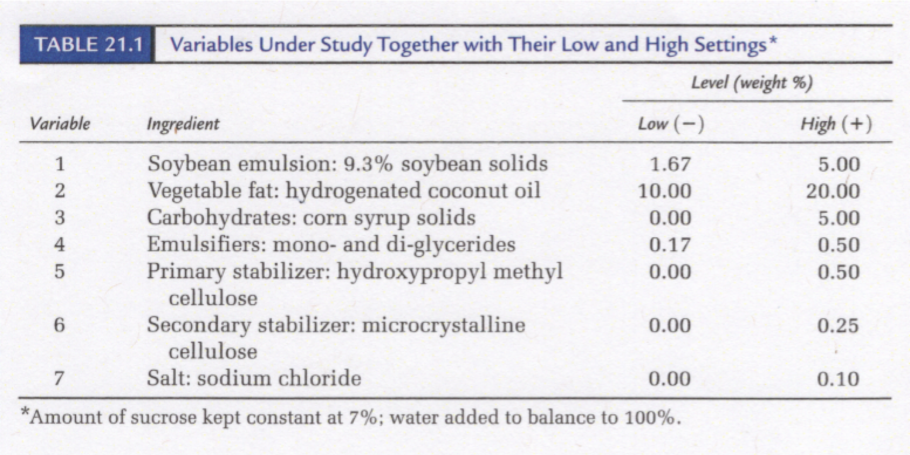


TABLE 21.1 Variables Under Study Together with Their Low and High Settings*

Variable	Ingredient	Level (weight %) Low (−)	Level (weight %) High (+)
1	Soybean emulsion: 9.3% soybean solids	1.67	5.00
2	Vegetable fat: hydrogenated coconut oil	10.00	20.00
3	Carbohydrates: corn syrup solids	0.00	5.00
4	Emulsifiers: mono- and di-glycerides	0.17	0.50
5	Primary stabilizer: hydroxypropyl methyl cellulose	0.00	0.50
6	Secondary stabilizer: microcrystalline cellulose	0.00	0.25
7	Salt: sodium chloride	0.00	0.10

*Amount of sucrose kept constant at 7%; water added to balance to 100%.

Example Problem - Design Matrix

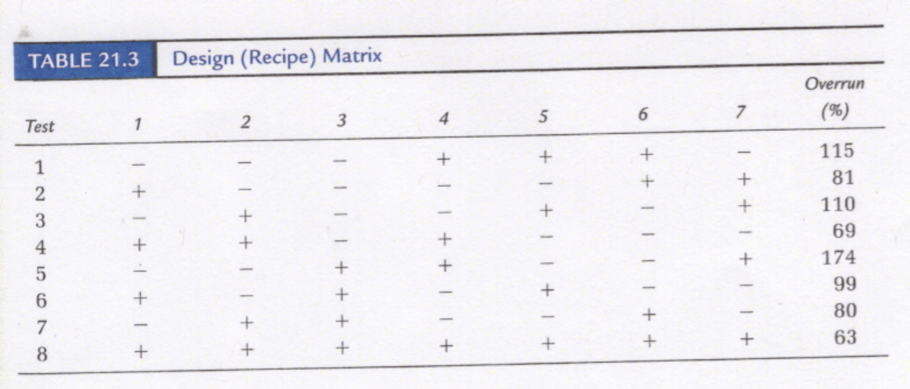


TABLE 21.3 Design (Recipe) Matrix

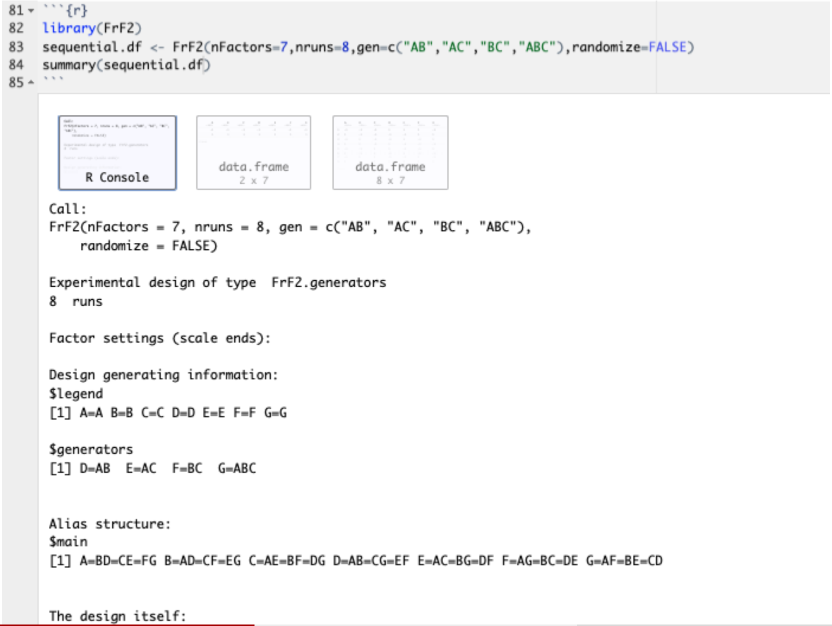
Test	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*Overrun (%)*
1	–	–	–	+	+	+	–	115
2	+	–	–	–	–	+	+	81
3	–	+	–	–	+	–	+	110
4	+	+	–	+	–	–	–	69
5	–	–	+	+	–	–	+	174
6	+	–	+	–	+	–	–	99
7	–	+	+	–	–	+	–	80
8	+	+	+	+	+	+	+	63

Figure 2: Design Matrix for Example Problem

Design using R

~~~
```{r}
library(FrF2)
sequential.df <- FrF2(nFactors=7,nruns=8,gen=c("AB","AC","BC","ABC"),randomize=FALSE)
summary(sequential.df)
~~~
```

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

8 rows

Figure 3: Base Design in R
```

# R Design Details



~~~
```{r}
library(FrF2)
sequential.df <- FrF2(nFactors=7,nruns=8,gen=c("AB","AC","BC","ABC"),randomize=FALSE)
summary(sequential.df)
~~~
```

```
Call:
FrF2(nFactors = 7, nruns = 8, gen = c("AB", "AC", "BC", "ABC"),
    randomize = FALSE)

Experimental design of type  FrF2.generators
8  runs

Factor settings (scale ends):

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

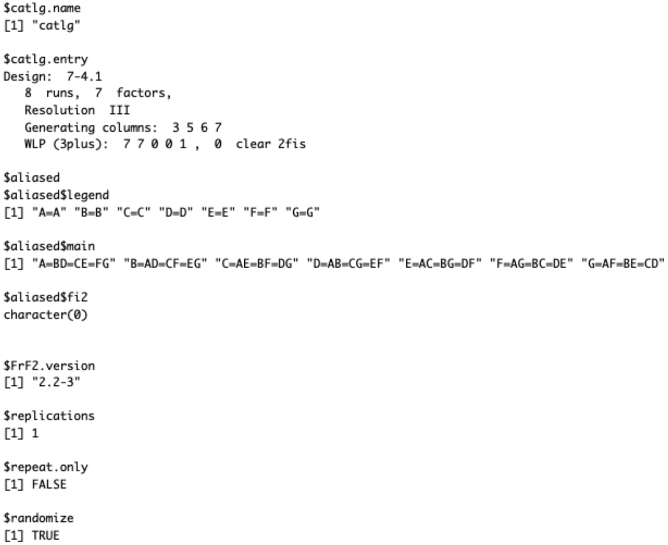
$generators
[1] D=AB   E=AC   F=BC   G=ABC

Alias structure:
$main
[1] A=BD=CE=FG B=AD=CF=EG C=AE=BF=DG D=AB=CG=EF E=AC=BG=DF F=AG=BC=DE G=AF=BE=CD

The design itself:
```
```

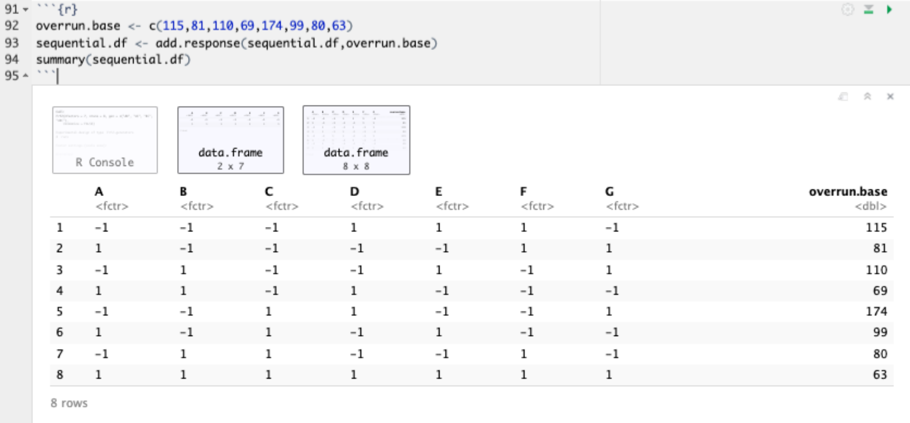
FrF2 Catalog Information

- Design is 7-4.1
- Design.info is shown below



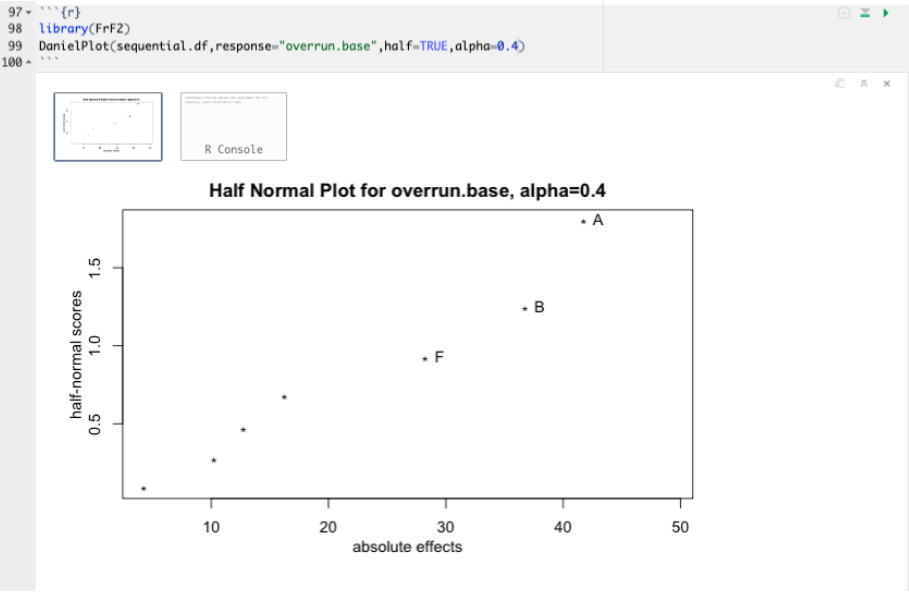
```
$catlg.name
[1] "catlg"

$catlg.entry
Design:  7-4.1
   8  runs,  7  factors,
   Resolution  III
   Generating columns:  3 5 6 7
   WLP (3plus):  7 7 0 0 1 ,  0  clear 2fis

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

$aliased$main
[1] "A=BD=CE=FG" "B=AD=CF=EG" "C=AE=BF=DG" "D=AB=CG=EF" "E=AC=BG=DF" "F=AG=BC=DE" "G=AF=BE=CD"

$aliased$fi2
character(0)


$FrF2.version
[1] "2.2-3"

$replications
[1] 1

$repeat.only
[1] FALSE

$randomize
[1] TRUE
```

R: Response Variable



~~~
```{r}
overrun.base <- c(115,81,110,69,174,99,80,63)
sequential.df <- add.response(sequential.df,overrun.base)
summary(sequential.df)
~~~
```

	A <fctr>	B <fctr>	C <fctr>	D <fctr>	E <fctr>	F <fctr>	G <fctr>	overrun.base <dbl>
1	-1	-1	-1	1	1	1	-1	115
2	1	-1	-1	-1	-1	1	1	81
3	-1	1	-1	-1	1	-1	1	110
4	1	1	-1	1	-1	-1	-1	69
5	-1	-1	1	1	-1	-1	1	174
6	1	-1	1	-1	1	-1	-1	99
7	-1	1	1	-1	-1	1	-1	80
8	1	1	1	1	1	1	1	63

8 rows

Figure 6: Response Variable for Example Problem
```

# Half-normal Plot



```{r}
library(FrF2)
DanielPlot(sequential.df,response="overrun.base",half=TRUE,alpha=0.4)
```

Interpreting Half-Normal Plot Results

- Factors A, B, and F alone are responsible for large effect estimates
- From aliasing scheme of A, BD and/or CE and/or FG could be responsible for large effect estimate for A (with or without A being large),
- From aliasing scheme of B, AD and/or DF and/or EG could be responsible for large effect estimate for B (with or without B being large),
- From aliasing scheme of F, AG and/or BC and/or DE could be respnosible for large effect estimate for F (with or without F being large)
- **No clear pattern**

Full Fold-over

- Full fold-over occurs when you change all the signs in the design matrix; run another fraction
- A full fold-over will break the alias links between the main effects and two-factor interactions
    - Two-factor interactions may (still) be aliased with each other

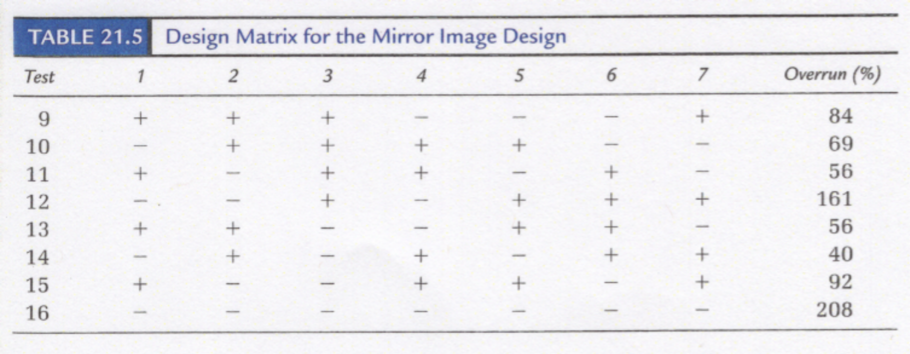
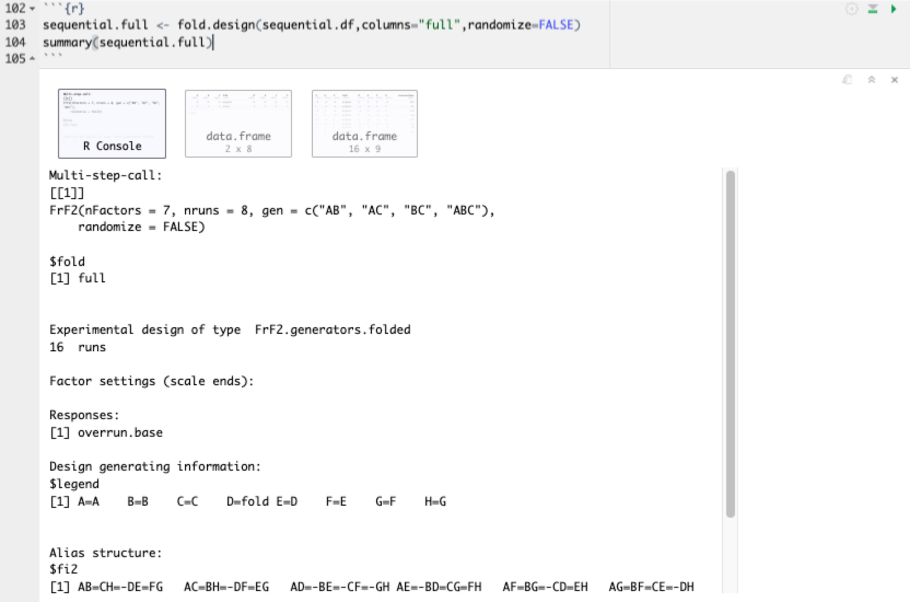


TABLE 21.5 Design Matrix for the Mirror Image Design

Test	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*Overrun (%)*
9	+	+	+	−	−	−	+	84
10	−	+	+	+	+	−	−	69
11	+	−	+	+	−	+	−	56
12	−	−	+	−	+	+	+	161
13	+	+	−	−	+	+	−	56
14	−	+	−	+	−	+	+	40
15	+	−	−	+	+	−	+	92
16	−	−	−	−	−	−	−	208

Figure 8: DCS Full Fold-Over

Full Fold-Over in R



~~~
```{r}
sequential.full <- fold.design(sequential.df,columns="full",randomize=FALSE)
summary(sequential.full)
~~~
```

```
Multi-step-call:
[[1]]
FrF2(nFactors = 7, nruns = 8, gen = c("AB", "AC", "BC", "ABC"),
    randomize = FALSE)

$fold
[1] full

Experimental design of type FrF2.generators.folded
16 runs

Factor settings (scale ends):

Responses:
[1] overrun.base

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

Alias structure:
$fi2
[1] AB=CH=-DE=FG   AC=BH=-DF=EG   AD=-BE=-CF=-GH AE=-BD=CG=FH   AF=BG=-CD=EH   AG=BF=CE=-DH
```
```

Full Fold-over Design

~~~
```{r}
sequential.full <- fold.design(sequential.df,columns="full",randomize=FALSE)
summary(sequential.full)
```
~~~

A <fctr>	B <fctr>	C <fctr>	fold <fctr>	D <fctr>	E <fctr>	F <fctr>	G <fctr>	overrun.base <dbl>
-1	-1	-1	original	1	1	1	-1	115
1	-1	-1	original	-1	-1	1	1	81
-1	1	-1	original	-1	1	-1	1	110
1	1	-1	original	1	-1	-1	-1	69
-1	-1	1	original	1	-1	-1	1	174
1	-1	1	original	-1	1	-1	-1	99
-1	1	1	original	-1	-1	1	-1	80
1	1	1	original	1	1	1	1	63
1	1	1	mirror	-1	-1	-1	1	NA
-1	1	1	mirror	1	1	-1	-1	NA

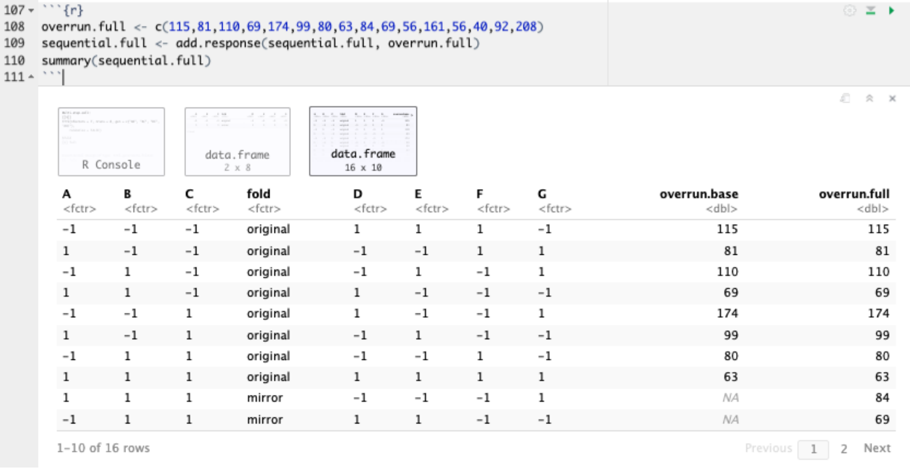
1-10 of 16 rows

Figure 10: Full fold-Over in R

Full Fold-over Design, part 2

~~~
```{r}
sequential.full <- fold.design(sequential.df,columns="full",randomize=FALSE)
summary(sequential.full)
~~~
```

A <fctr>	B <fctr>	C <fctr>	fold <fctr>	D <fctr>	E <fctr>	F <fctr>	G <fctr>	overrun.base <dbl>
1	-1	1	mirror	1	-1	1	-1	NA
-1	-1	1	mirror	-1	1	1	1	NA
1	1	-1	mirror	-1	1	1	-1	NA
-1	1	-1	mirror	1	-1	1	1	NA
1	-1	-1	mirror	1	1	-1	1	NA
-1	-1	-1	mirror	-1	-1	-1	-1	NA

11-16 of 16 rows

Figure 11: Full Fold-Over Design, Part 2
```

# Fold-over Design with Response



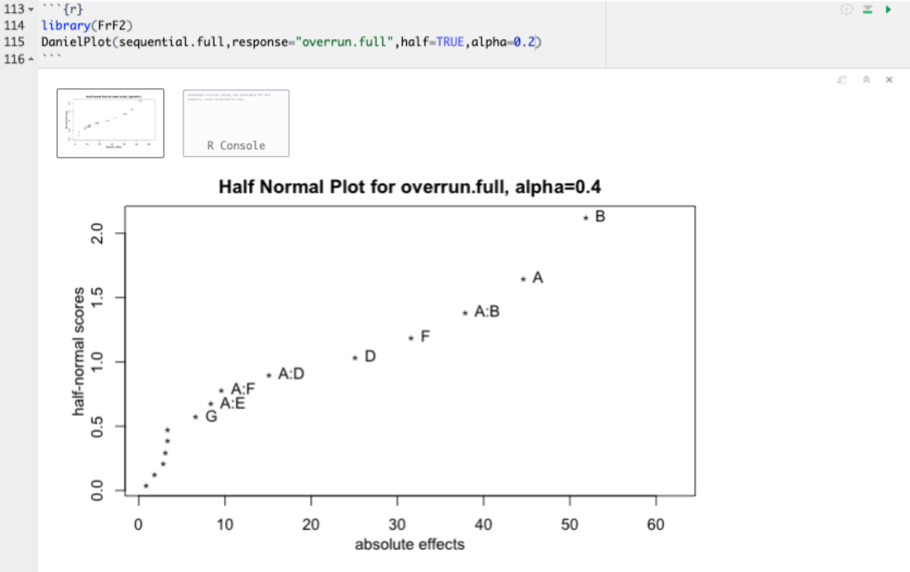
~~~
```{r}
overrun.full <- c(115,81,110,69,174,99,80,63,84,69,56,161,56,40,92,208)
sequential.full <- add.response(sequential.full, overrun.full)
summary(sequential.full)
```
~~~

A <fctr>	B <fctr>	C <fctr>	fold <fctr>	D <fctr>	E <fctr>	F <fctr>	G <fctr>	overrun.base <dbl>	overrun.full <dbl>
-1	-1	-1	original	1	1	1	-1	115	115
1	-1	-1	original	-1	-1	1	1	81	81
-1	1	-1	original	-1	1	-1	1	110	110
1	1	-1	original	1	-1	-1	-1	69	69
-1	-1	1	original	1	-1	-1	1	174	174
1	-1	1	original	-1	1	-1	-1	99	99
-1	1	1	original	-1	-1	1	-1	80	80
1	1	1	original	1	1	1	1	63	63
1	1	1	mirror	-1	-1	-1	1	*NA*	84
-1	1	1	mirror	1	1	-1	-1	*NA*	69

1-10 of 16 rows

Figure 12: Response Variable for Fuill Fold-over

Half-Normal Plot for Fold-over

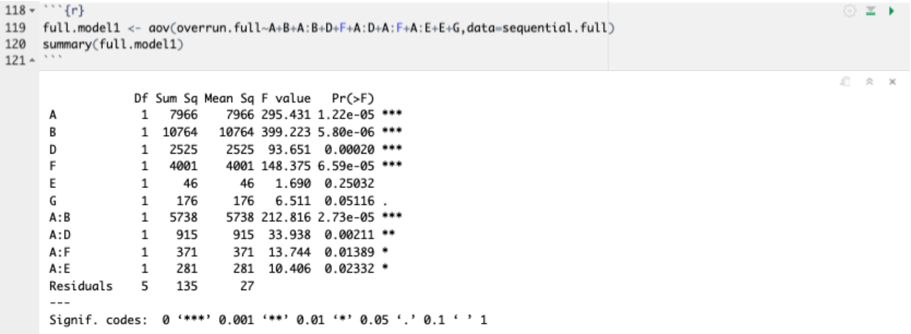


~~~
```{r}
library(FrF2)
DanielPlot(sequential.full,response="overrun.full",half=TRUE,alpha=0.2)
~~~
```

Figure 13: Half Normal Plot for Full fold-over Design
```

Interpretation of Half Normal Plot

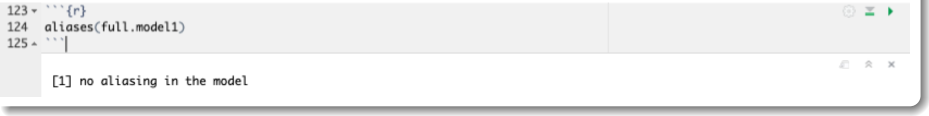
- All main effects are not aliased with other main effects or two factor interactions (A, B, F, D, G)
- From aliasing scheme for AB, CH and/or -DE and/or FG (with or without AB being large)
- From aliasing scheme for AD, -BE and/or -DF and/or -Gh (with or withont AD being large)
- From aliasing scheme for AE, -DB and/or CG and/or FH (with or without AE being large)

Initial Model



~~~
```{r}
full.model1 <- aov(overrun.full~A+B+A:B+D+F+A:D+A:F+A:E+E+G,data=sequential.full)
summary(full.model1)
~~~
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)    
A            1   7966    7966 295.431 1.22e-05 ***
B            1  10764   10764 399.223 5.80e-06 ***
D            1   2525    2525  93.651  0.00020 ***
F            1   4001    4001 148.375 6.59e-05 ***
E            1     46      46   1.690  0.25032    
G            1    176     176   6.511  0.05116 .  
A:B          1   5738    5738 212.816 2.73e-05 ***
A:D          1    915     915  33.938  0.00211 ** 
A:F          1    371     371  13.744  0.01389 *  
A:E          1    281     281  10.406  0.02332 *  
Residuals    5    135      27                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```
```

## Aliasing in Initial Model



~~~
```{r}
aliases(full.model1)
~~~
```

```
[1] no aliasing in the model
```
```