Section 1

MANE 6313

Subsection 1

Week 11, Module E

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 fold over on a factor.

Fold-over on a Factor

- To separate an interesting effect, e.g. C, go to the approppriate column in the design matrix and change the sign for all rows in that column
- Folding over on a factor will result in a combined design that:
  - Main effect of factor will not be aliased with other two-factor interactions,
  - In general, all two-factor interactions involving the factor used for the fold-over will not be aliased with other two-factor interactions

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)

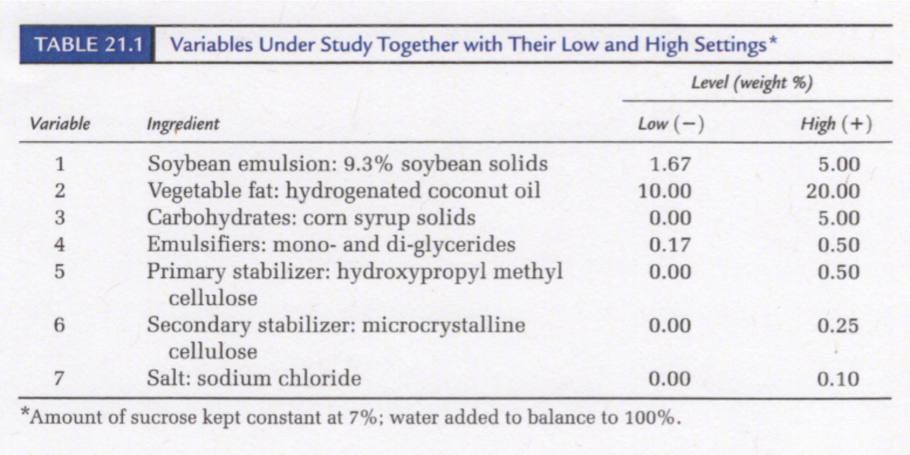


TABLE 21.1 Variables Under Study Together with Their Low and High Settings*

		Level (weight %)	
Variable	*Ingredient*	*Low* (−)	*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

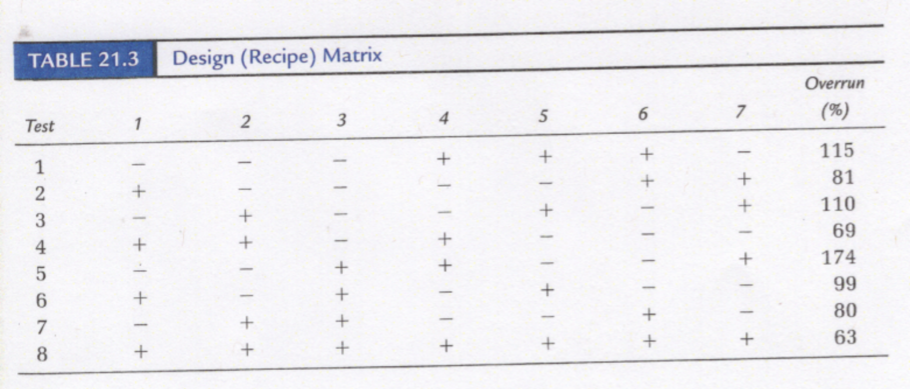
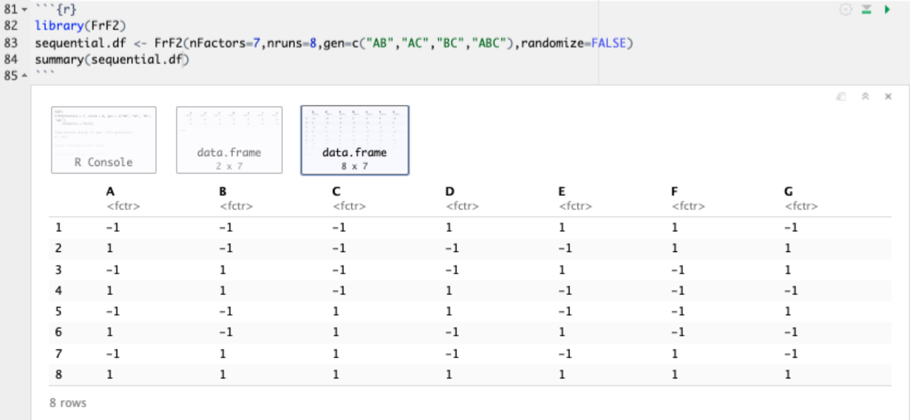


TABLE 21.3 Design (Recipe) Matrix

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: Example Design Matrix

Design using R



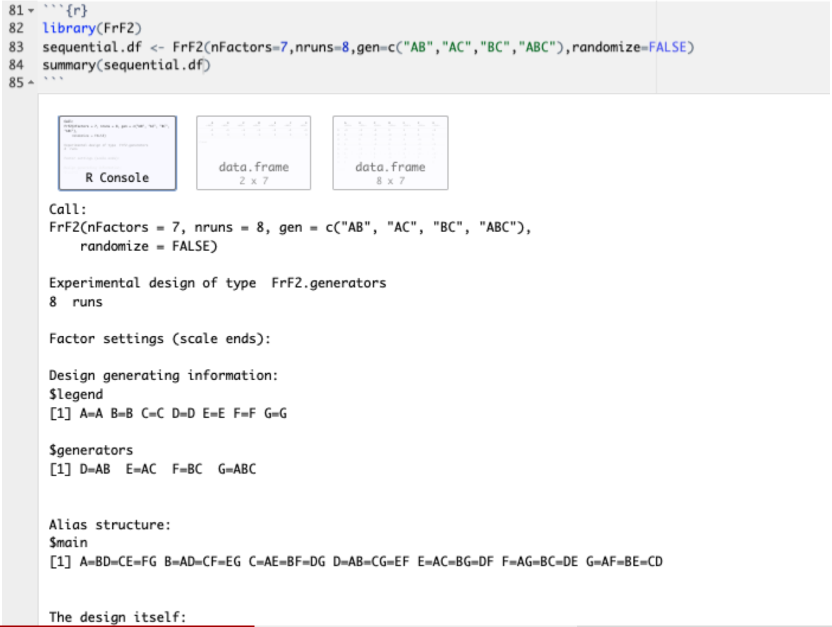
```{r}
library(FrF2)
sequential.df <- FrF2(nFactors=7,nruns=8,gen=c("AB","AC","BC","ABC"),randomize=FALSE)
summary(sequential.df)
```

	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: Initial Design in R

R Design Details



~~~r
```{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):

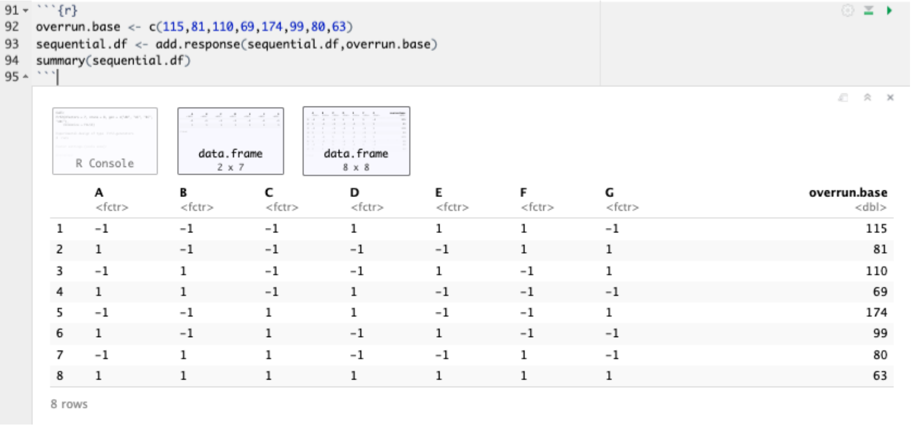
Design generating information:
$legend
[1] A=A B=B C=C D=D E=E F=F G=G

$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:
```
```

# R: Response Variable



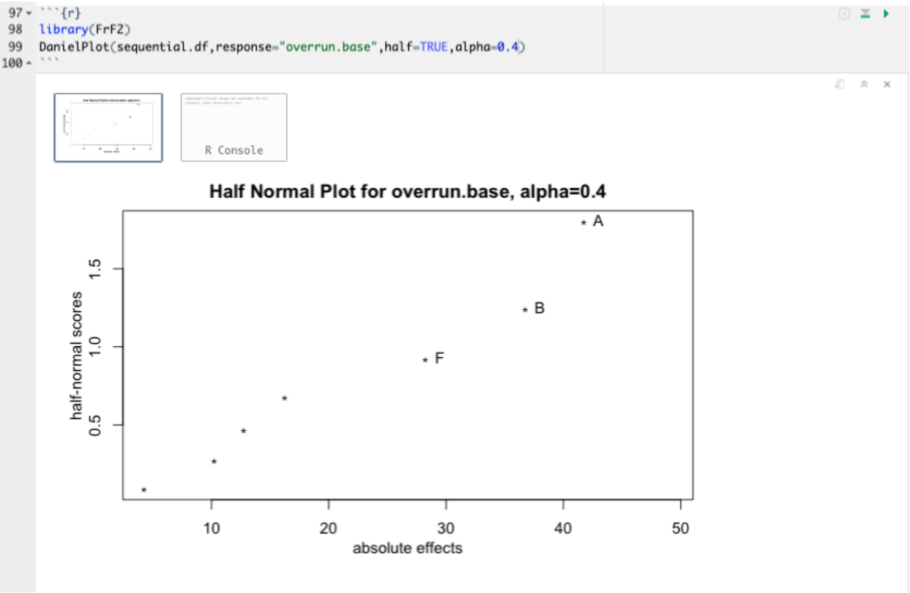
~~~
```{r}
overrun.base <- c(115,81,110,69,174,99,80,63)
sequential.df <- add.response(sequential.df,overrun.base)
summary(sequential.df)
```
~~~

	A	B	C	D	E	F	G	overrun.base
	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<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 5: Adding Response Variable in R

Half-normal Plot



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

Fold-over on A

- Change the sign for all rows of design column matrix for variable A; run another fraction

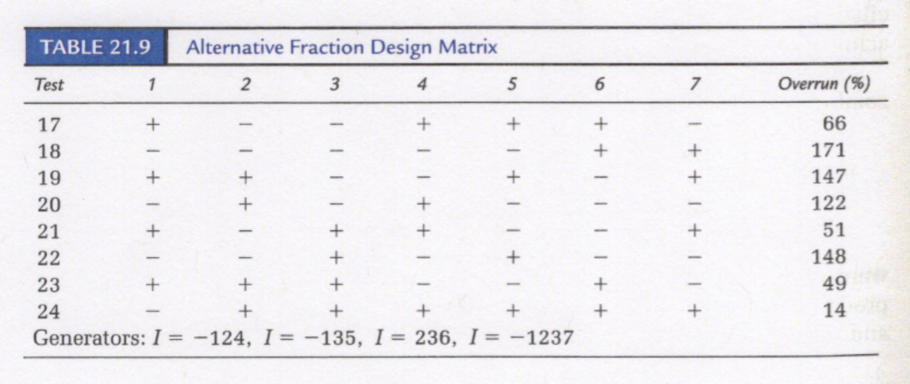


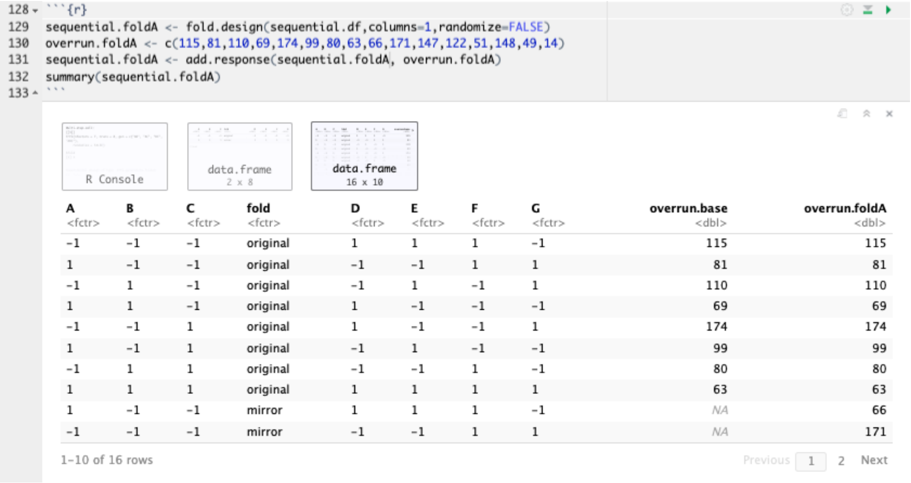
TABLE 21.9 Alternative Fraction Design Matrix

Test	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*Overrun (%)*
17	+	−	−	+	+	+	−	66
18	−	−	−	−	−	+	+	171
19	+	+	−	−	+	−	+	147
20	−	+	−	+	−	−	−	122
21	+	−	+	+	−	−	+	51
22	−	−	+	−	+	−	−	148
23	+	+	+	−	−	+	−	49
24	−	+	+	+	+	+	+	14

Generators: $I = -124$, $I = -135$, $I = 236$, $I = -1237$

Figure 7: Design Matrix for Fold-over on A

Fold-Over on A in R



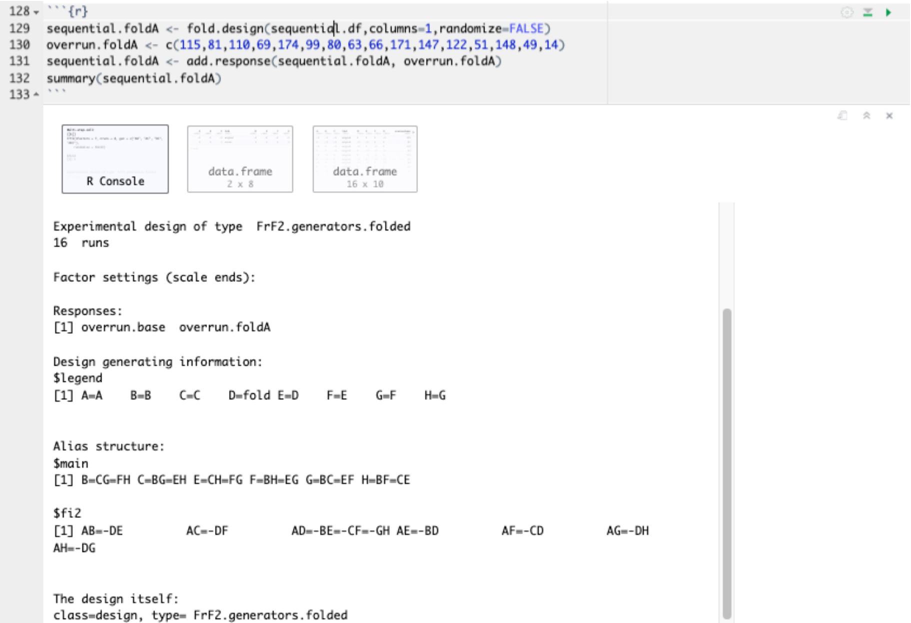
~~~
```{r}
sequential.foldA <- fold.design(sequential.df,columns=1,randomize=FALSE)
overrun.foldA <- c(115,81,110,69,174,99,80,63,66,171,147,122,51,148,49,14)
sequential.foldA <- add.response(sequential.foldA, overrun.foldA)
summary(sequential.foldA)
```
~~~

A <fctr>	B <fctr>	C <fctr>	fold <fctr>	D <fctr>	E <fctr>	F <fctr>	G <fctr>	overrun.base <dbl>	overrun.foldA <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	66
-1	-1	-1	mirror	-1	-1	1	1	NA	171

1-10 of 16 rows

Figure 8: Design Matrix with Responses

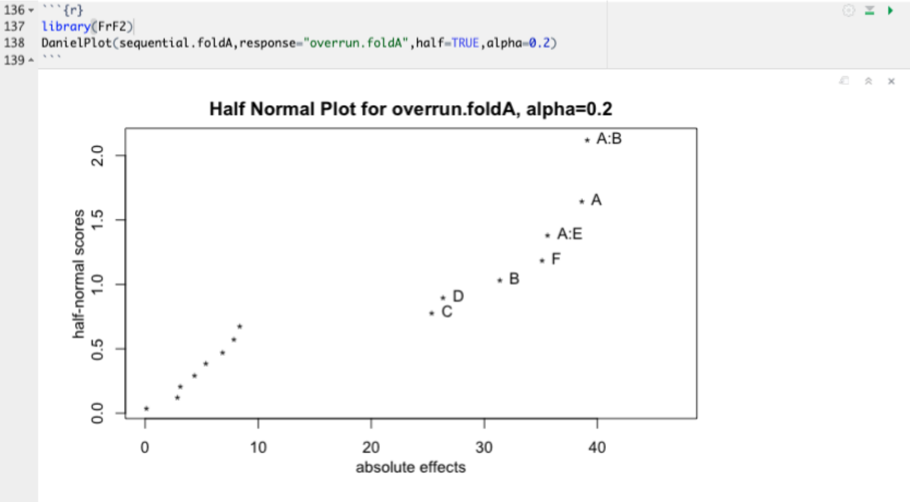
Aliasing Information from Design



```{r}
sequential.foldA <- fold.design(sequential.df,columns=1,randomize=FALSE)
overrun.foldA <- c(115,81,110,69,174,99,80,63,66,171,147,122,51,148,49,14)
sequential.foldA <- add.response(sequential.foldA, overrun.foldA)
summary(sequential.foldA)
```

```
Experimental design of type  FrF2.generators.folded
16  runs

Factor settings (scale ends):

Responses:
[1] overrun.base  overrun.foldA

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


Alias structure:
$main
[1] B=CG=FH C=BG=EH E=CH=FG F=BH=EG G=BC=EF H=BF=CE

$fi2
[1] AB=-DE          AC=-DF          AD=-BE=-CF=-GH AE=-BD          AF=-CD          AG=-DH
AH=-DG


The design itself:
class=design, type= FrF2.generators.folded
```

Half-Normal Plot for Fold-over



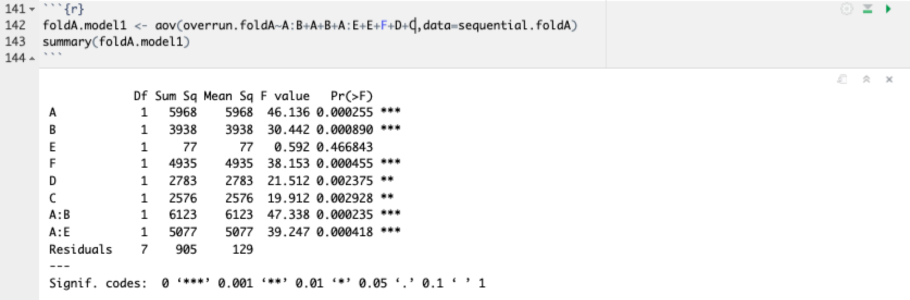
```{r}
library(FrF2)
DanielPlot(sequential.foldA,response="overrun.foldA",half=TRUE,alpha=0.2)
```

Figure 10: Half-Normal Plot for Fold-over Design

Interpretation of Half Normal Plot

- All main effects are not aliased with other main effects or two factor interactions (A, C, D)
- From aliasing scheme for AG, BF and/or CD and/or EH (with or without AG being large)
- From aliasing scheme for AD, -BE and/or -CF and/or -FH (with or withont AD being large)
- From aliasing scheme for AB, CH and/or -DE and/or FG (with or without AB being large)
- From aliasing scheme for AE, -BD and/or CG and/or FH (with or without AE being large)

Initial Model

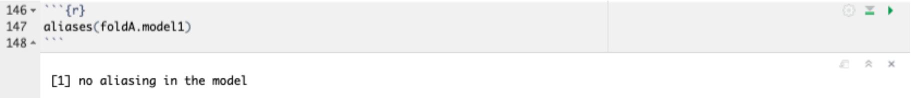


~~~
```{r}
foldA.model1 <- aov(overrun.foldA~A:B+A+B+A:E+E+F+D+C,data=sequential.foldA)
summary(foldA.model1)
```
~~~

```
            Df Sum Sq Mean Sq F value   Pr(>F)    
A            1   5968    5968  46.136 0.000255 ***
B            1   3938    3938  30.442 0.000890 ***
E            1     77      77   0.592 0.466843    
F            1   4935    4935  38.153 0.000455 ***
D            1   2783    2783  21.512 0.002375 ** 
C            1   2576    2576  19.912 0.002928 ** 
A:B          1   6123    6123  47.338 0.000235 ***
A:E          1   5077    5077  39.247 0.000418 ***
Residuals    7    905     129                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 11: Initial Model in R

Aliasing in Folded Model



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

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

Figure 12: Aliasing in Folded Model