Chapter 4

Two-Level Fractional Factorial Designs

4.1 Generate a 2^{5-2}_{III} design

Original design is a 2^{5-2}III fractional factorial design.

Below, I specify generators I=ABD=ACE. A, B, and C in notes are x_3, x_4, and x_5, respectively. E=AC and below $x_1 = -x_3x_5$, and D=AB and below $x_2 = -x_1x_4$.

```
#### Generate design
library(rsm)

# help for cube, see examples
?cube

# create the first block
block1 <- cube( basis = ~ x1 + x2 + x3 + x4 + x5
              , n0 = 0
              , blockgen = ~ c(x1 * x2 * x4, x1 * x3 * x5)
              , randomize = FALSE
              , bid = 1)
block1
```

```
##    run.order std.order x1.as.is x2.as.is x3.as.is x4.as.is x5.as.is
## 1          1         1       -1       -1       -1       -1       -1
## 8          2         2        1        1        1       -1       -1
## 11         3         3       -1        1       -1        1       -1
## 14         4         4        1       -1        1        1       -1
## 20         5         5        1        1       -1       -1        1
## 21         6         6       -1       -1        1       -1        1
## 26         7         7        1       -1       -1        1        1
## 31         8         8       -1        1        1        1        1
##
## Data are stored in coded form using these coding formulas ...
## x1 ~ x1.as.is
## x2 ~ x2.as.is
## x3 ~ x3.as.is
## x4 ~ x4.as.is
## x5 ~ x5.as.is
```

To put this in the same order as in the notes, change the order of the basis.

```
# create the first block
block1 <- cube( basis = ~ x4 + x5 + x1 + x2 + x3
              , n0 = 0
              , blockgen = ~ c(x1 * x2 * x4, x1 * x3 * x5)
              , randomize = FALSE
              , bid = 1)
block1
```

```
##    run.order std.order x4.as.is x5.as.is x1.as.is x2.as.is x3.as.is
## 1          1         1       -1       -1       -1       -1       -1
## 8          2         2        1        1        1       -1       -1
## 10         3         3        1       -1       -1        1       -1
## 15         4         4       -1        1        1        1       -1
```

```
## 19         5         5       -1        1       -1       -1        1
## 22         6         6        1       -1        1       -1        1
## 28         7         7        1        1       -1        1        1
## 29         8         8       -1       -1        1        1        1
##
## Data are stored in coded form using these coding formulas ...
## x4 ~ x4.as.is
## x5 ~ x5.as.is
## x1 ~ x1.as.is
## x2 ~ x2.as.is
## x3 ~ x3.as.is
```

4.2 Example 4.5, Table 4.15, p. 161

Original design is a 2^{7-4}III fractional factorial design.

```
#### 4.5a
fn.data <- "http://statacumen.com/teach/RSM/data/RSM_EX_04-05a.txt"
df.4.5a <- read.table(fn.data, header=TRUE)
str(df.4.5a)
```

```
## 'data.frame': 8 obs. of  4 variables:
##  $ a   : int  -1 1 -1 1 -1 1 -1 1
##  $ b   : int  -1 -1 1 1 -1 -1 1 1
##  $ c   : int  -1 -1 -1 -1 1 1 1 1
##  $ time: num  85.5 75.1 93.2 145.4 83.7 ...
```

```
df.4.5a
```

```
##    a  b  c  time
## 1 -1 -1 -1  85.5
## 2  1 -1 -1  75.1
## 3 -1  1 -1  93.2
## 4  1  1 -1 145.4
## 5 -1 -1  1  83.7
## 6  1 -1  1  77.6
## 7 -1  1  1  95.0
## 8  1  1  1 141.8
```

Fit first-order with three-way interaction linear model.

```
lm.4.5a.time.3WIabc <- lm(time ~ (a + b + c)^3, data = df.4.5a)
## externally Studentized residuals
#lm.4.5a.time.3WIabc£studres <- rstudent(lm.4.5a.time.3WIabc)
summary(lm.4.5a.time.3WIabc)
```

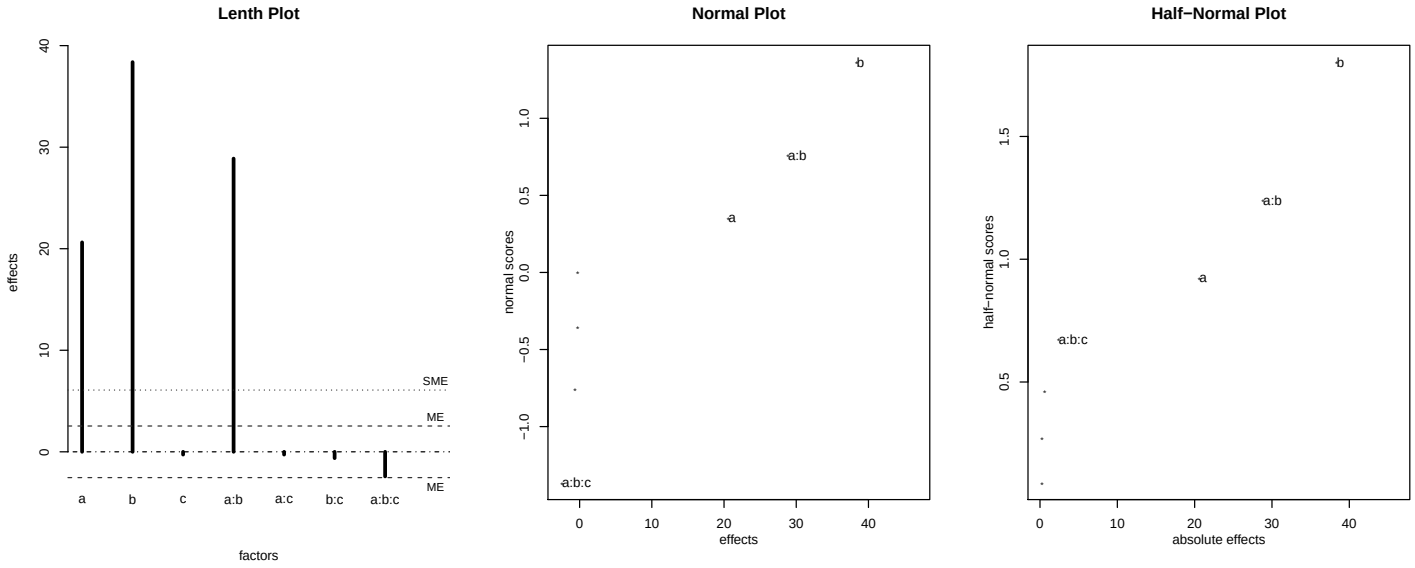
```
##
## Call:
## lm.default(formula = time ~ (a + b + c)^3, data = df.4.5a)
##
## Residuals:
## ALL 8 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   99.662         NA      NA       NA
## a             10.312         NA      NA       NA
## b             19.188         NA      NA       NA
## c             -0.137         NA      NA       NA
## a:b           14.438         NA      NA       NA
## a:c           -0.138         NA      NA       NA
## b:c           -0.313         NA      NA       NA
## a:b:c         -1.212         NA      NA       NA
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:     1,Adjusted R-squared:   NaN
## F-statistic:  NaN on 7 and 0 DF,  p-value: NA
```

The Lenth plot below indicates three large effects worth investigating (a, b, ab).

```
# BsMD package has unreplicated factorial tests (Daniel plots (aka normal), and Lenth)
library(BsMD)
par(mfrow=c(1,3))
LenthPlot(lm.4.5a.time.3WIabc, alpha = 0.05, main = "Lenth Plot") # , adj = 0.2
```

```
## alpha   PSE    ME   SME
## 0.050 0.675 2.541 6.081
```

```
DanielPlot(lm.4.5a.time.3WIabc, main = "Normal Plot")
DanielPlot(lm.4.5a.time.3WIabc, half = TRUE, main = "Half-Normal Plot")
```



4.3 Example 4.5, Table 4.16, p. 163

Full fold-over design. Note that the column for a, b, and c are -1 times the same columns in the original design.

```
#### 4.5b
fn.data <- "http://statacumen.com/teach/RSM/data/RSM_EX_04-05b.txt"
df.4.5b <- read.table(fn.data, header=TRUE)
str(df.4.5b)
```

```
## 'data.frame': 8 obs. of  4 variables:
##  $ a   : int  1 -1 1 -1 1 -1 1 -1
##  $ b   : int  1 1 -1 -1 1 1 -1 -1
##  $ c   : int  1 1 1 1 -1 -1 -1 -1
##  $ time: num  91.3 136.7 82.4 73.4 94.1 ...
```

```
df.4.5b
```

```
##    a  b  c  time
## 1  1  1  1  91.3
## 2 -1  1  1 136.7
## 3  1 -1  1  82.4
## 4 -1 -1  1  73.4
## 5  1  1 -1  94.1
## 6 -1  1 -1 143.8
## 7  1 -1 -1  87.3
## 8 -1 -1 -1  71.9
```

Fit first-order with three-way interaction linear model.

```
lm.4.5b.time.3WIabc <- lm(time ~ (a + b + c)^3, data = df.4.5b)
## externally Studentized residuals
#lm.4.5b.time.3WIabc£studres <- rstudent(lm.4.5b.time.3WIabc)
summary(lm.4.5b.time.3WIabc)
```

```
##
## Call:
## lm.default(formula = time ~ (a + b + c)^3, data = df.4.5b)
##
## Residuals:
## ALL 8 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   97.612         NA      NA       NA
## a             -8.838         NA      NA       NA
## b             18.862         NA      NA       NA
## c             -1.663         NA      NA       NA
## a:b          -14.938         NA      NA       NA
## a:c           -0.263         NA      NA       NA
## b:c           -0.813         NA      NA       NA
## a:b:c          1.337         NA      NA       NA
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:     1,Adjusted R-squared:   NaN
```

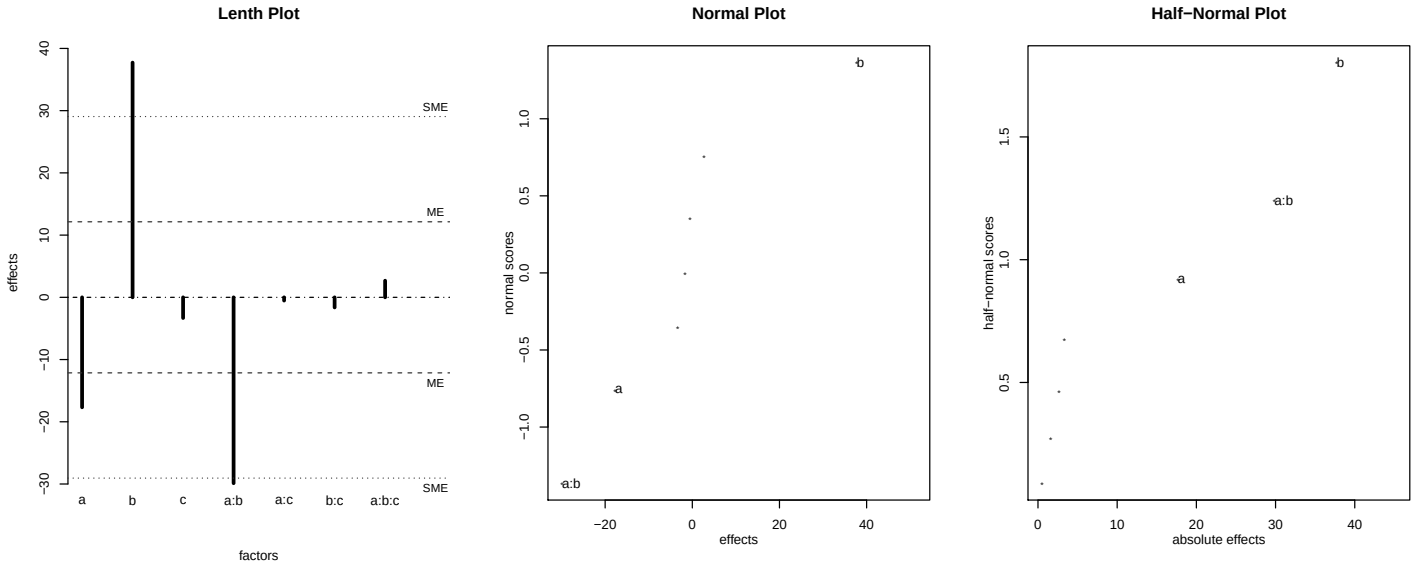
```
## F-statistic:  NaN on 7 and 0 DF,  p-value: NA
```

The Lenth plot below indicates the same three large effects are worth investigating (a, b, ab), but some effects are in different directions.

```
# BsMD package has unreplicated factorial tests (Daniel plots (aka normal), and Lenth)
library(BsMD)
par(mfrow=c(1,3))
LenthPlot(lm.4.5b.time.3WIabc, alpha = 0.05, main = "Lenth Plot") # , adj = 0.2
```

```
##   alpha    PSE     ME    SME
##   0.050  3.225 12.139 29.052
```

```
DanielPlot(lm.4.5b.time.3WIabc, main = "Normal Plot")
DanielPlot(lm.4.5b.time.3WIabc, half = TRUE, main = "Half-Normal Plot")
```



4.4 Combining the data

Combine the two datasets to see what the full factorial suggests.

```
# combine two data sets
df.4.5 <- rbind(df.4.5a, df.4.5b)
str(df.4.5)
```

```
## 'data.frame': 16 obs. of  4 variables:
##  $ a   : int  -1 1 -1 1 -1 1 -1 1 1 -1 ...
##  $ b   : int  -1 -1 1 1 -1 -1 1 1 1 1 ...
##  $ c   : int  -1 -1 -1 -1 1 1 1 1 1 1 ...
##  $ time: num  85.5 75.1 93.2 145.4 83.7 ...
```

```
df.4.5
```

```
##     a  b  c  time
## 1  -1 -1 -1  85.5
## 2   1 -1 -1  75.1
## 3  -1  1 -1  93.2
## 4   1  1 -1 145.4
## 5  -1 -1  1  83.7
## 6   1 -1  1  77.6
## 7  -1  1  1  95.0
## 8   1  1  1 141.8
## 9   1  1  1  91.3
## 10 -1  1  1 136.7
## 11  1 -1  1  82.4
## 12 -1 -1  1  73.4
## 13  1  1 -1  94.1
## 14 -1  1 -1 143.8
## 15  1 -1 -1  87.3
## 16 -1 -1 -1  71.9
```

Fit first-order with three-way interaction linear model.

```
lm.4.5.time.3WIabc <- lm(time ~ (a + b + c)^3, data = df.4.5)
## externally Studentized residuals
#lm.4.5.time.3WIabc£studres <- rstudent(lm.4.5.time.3WIabc)
summary(lm.4.5.time.3WIabc)
```

```
##
## Call:
## lm.default(formula = time ~ (a + b + c)^3, data = df.4.5)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -25.6  -10.3    0.0   10.3   25.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  98.6375     6.2325   15.83  2.5e-07 ***
## a             0.7375     6.2325    0.12    0.909
## b            19.0250     6.2325    3.05    0.016 *
## c            -0.9000     6.2325   -0.14    0.889
## a:b          -0.2500     6.2325   -0.04    0.969
```

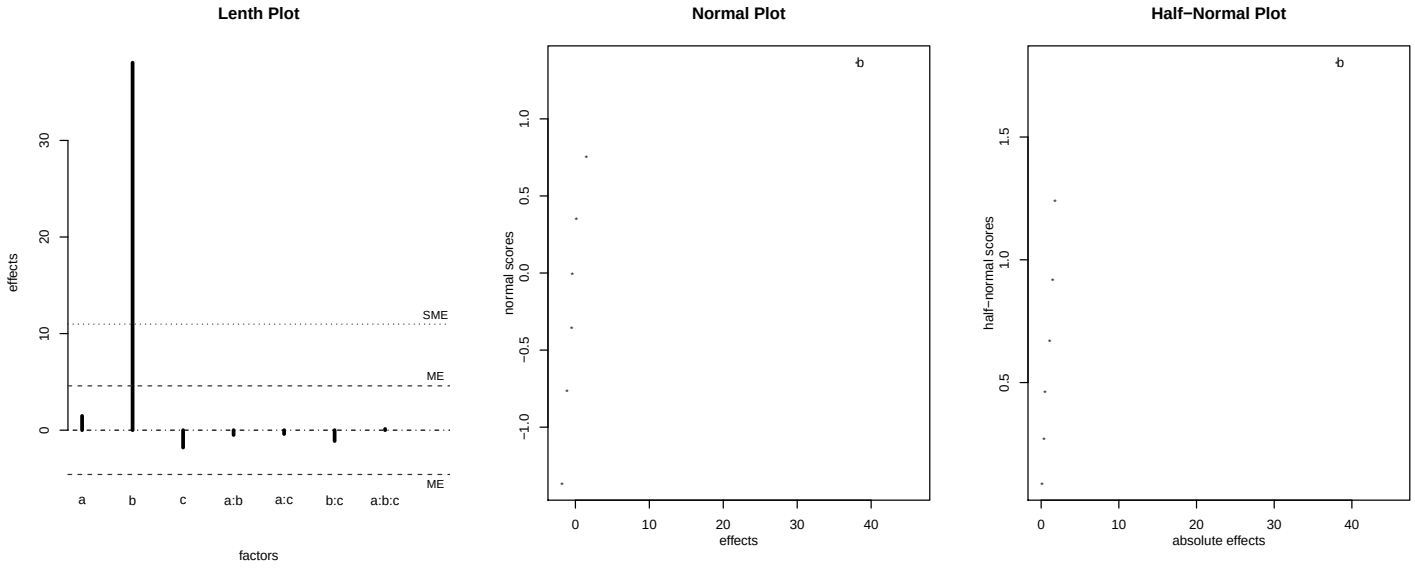
```
## a:c          -0.2000     6.2325   -0.03    0.975
## b:c          -0.5625     6.2325   -0.09    0.930
## a:b:c         0.0625     6.2325    0.01    0.992
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.9 on 8 degrees of freedom
## Multiple R-squared:  0.539,Adjusted R-squared:  0.136
## F-statistic: 1.34 on 7 and 8 DF,  p-value: 0.344
```

With more evidence, only effect b is important.

```
# BsMD package has unreplicated factorial tests (Daniel plots (aka normal), and Lenth)
library(BsMD)
par(mfrow=c(1,3))
LenthPlot(lm.4.5.time.3WIabc, alpha = 0.05, main = "Lenth Plot") # , adj = 0.2
```

```
##  alpha    PSE     ME    SME
##  0.050  1.219  4.588 10.979
```

```
DanielPlot(lm.4.5.time.3WIabc, main = "Normal Plot")
DanielPlot(lm.4.5.time.3WIabc, half = TRUE, main = "Half-Normal Plot")
```



4.5 Plackett-Berman design

Here are some examples of Plackett-Berman designs.

```
library(FrF2)
pb(nruns=8, randomize=FALSE)
```

```
## Warning:  Plackett-Burman designs in 8 runs coincide with regular fractional factorials.
##           For screening more than four factors, you may want to consider increasing the
number of runs to 12.
##           Make sure to take the alias structure into account for interpretation!
##    A  B  C  D  E  F  G
## 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
## class=design, type= pb
```

```
pb(nruns=12, randomize=FALSE)
```

```
##     A  B  C  D  E  F  G  H  J  K  L
## 1   1  1 -1  1  1  1 -1 -1 -1  1 -1
## 2  -1  1  1 -1  1  1  1 -1 -1 -1  1
## 3   1 -1  1  1 -1  1  1  1 -1 -1 -1
## 4  -1  1 -1  1  1 -1  1  1  1 -1 -1
## 5  -1 -1  1 -1  1  1 -1  1  1  1 -1
## 6  -1 -1 -1  1 -1  1  1 -1  1  1  1
## 7   1 -1 -1 -1  1 -1  1  1 -1  1  1
## 8   1  1 -1 -1 -1  1 -1  1  1 -1  1
## 9   1  1  1 -1 -1 -1  1 -1  1  1 -1
## 10 -1  1  1  1 -1 -1 -1  1 -1  1  1
## 11  1 -1  1  1  1 -1 -1 -1  1 -1  1
## 12 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
## class=design, type= pb
```