R Exercises-Session 5

Neil Diamond

March 26, 2018

1. TR 18532:2009 12.1.4.2.3 Pages 154-155

> On Silk screen printing, visual blemishes, termed "trail-marks" are being experienced. Squeegee speed, ink viscosity and dwell time were investigated. Factors and levels are shown in Table 1 below.
>
> Ten blank sheets of polyester material were taken from each run of a standard 2^3 design and a defect count was taken with the aid of a 200 square matrix. The total number of squares affected by trail-marks was counted for each run. This response was then recorded as a percentage. Resuts are shown in Table 2.

Design Factor	Level 1	Level 2
Squeegee speed	45	80
Ink viscocity	700 mPa.s	2,200 mPa.s
Dwell time	Auto	4.5

Table 1: Silk screen printing design factors and levels

Run	Squeegee speed	Ink viscosity	Dwell time	% Trail marks
1	−	−	−	0.00
2	+	−	−	1.05
3	−	+	−	5.40
4	+	+	−	0.05
5	−	−	+	0.20
6	+	−	+	0.05
7	−	+	+	5.85
8	+	+	+	0.00

Table 2: Results from the runs on the silk screen printing experiment

(a) Create a 2^3 factorial design.

```
library(FrF2)
des <- FrF2(nruns=8, randomize=FALSE,
factor.names=c("SqueegeeSpeed","InkViscosity","DwellTime"))
des
```

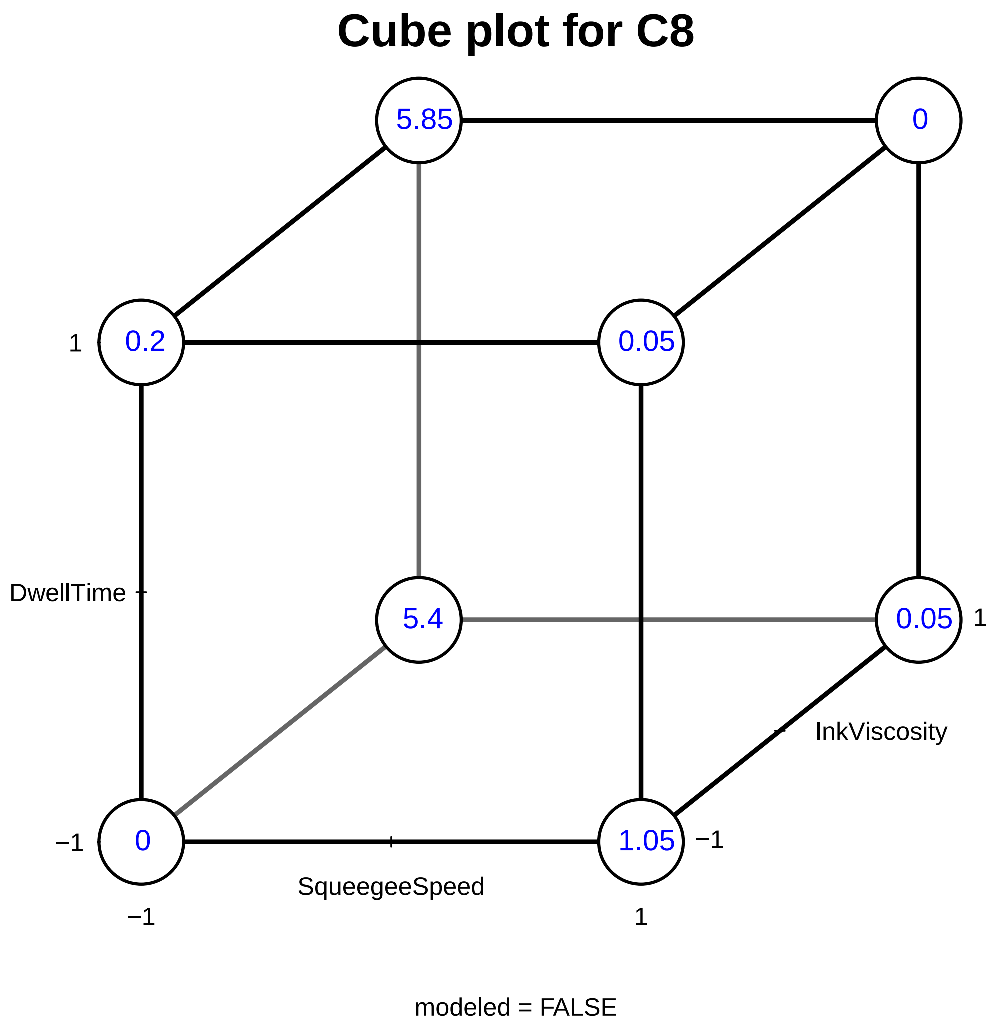
```
##   SqueegeeSpeed InkViscosity DwellTime
## 1            -1           -1        -1
## 2             1           -1        -1
## 3            -1            1        -1
## 4             1            1        -1
## 5            -1           -1         1
## 6             1           -1         1
## 7            -1            1         1
## 8             1            1         1
## class=design, type= full factorial
```

(b) Add results in C8 and make a cube plot.

```
C8 <- c(0.00, 1.05, 5.40, 0.05, 0.20, 0.05, 5.85, 0.00)
des.resp <- add.response(des, response = C8)
des.resp
```

```
##   SqueegeeSpeed InkViscosity DwellTime   C8
## 1            -1           -1        -1 0.00
## 2             1           -1        -1 1.05
## 3            -1            1        -1 5.40
## 4             1            1        -1 0.05
## 5            -1           -1         1 0.20
## 6             1           -1         1 0.05
## 7            -1            1         1 5.85
## 8             1            1         1 0.00
## class=design, type= full factorial
```

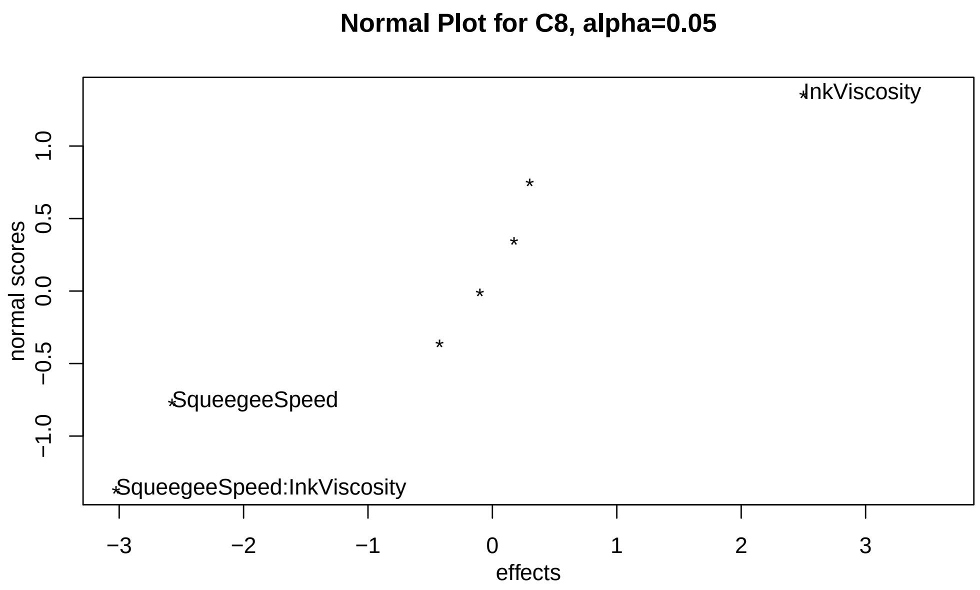
```
cubePlot(lm(des.resp, degree=3),
         "SqueegeeSpeed","InkViscosity","DwellTime", modeled=F)
```

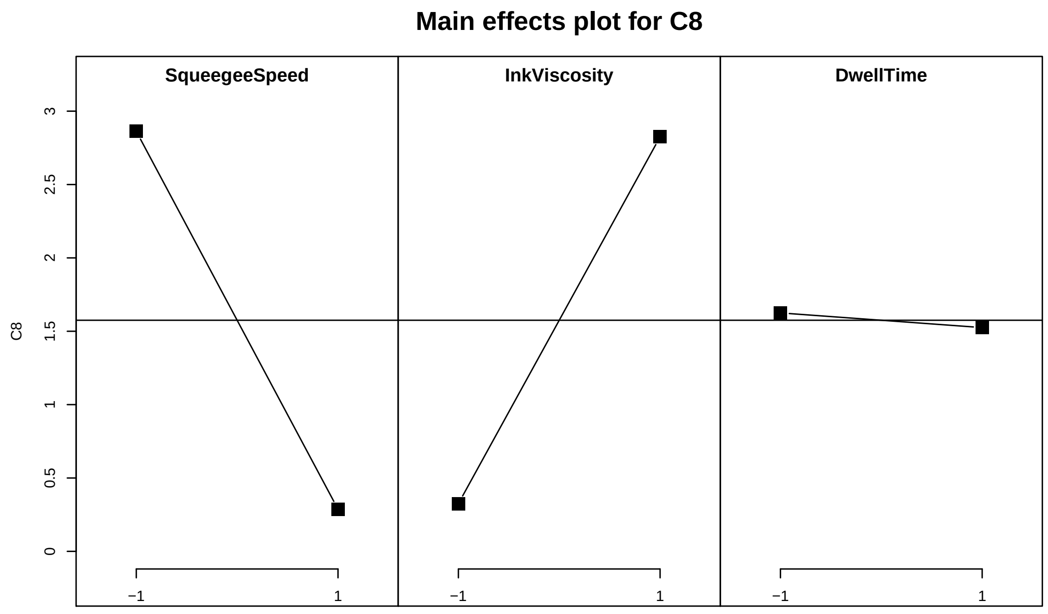


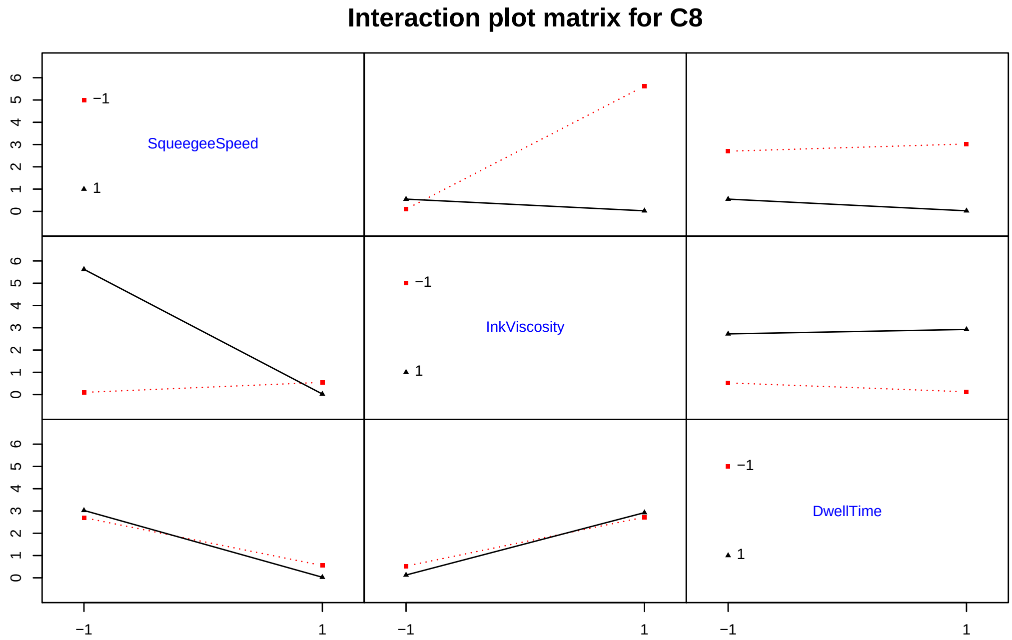
(c) Analyse the experiment.

```r
coef(lm(des.resp, degree=3))

##                         (Intercept)
##                              1.5750
##                     SqueegeeSpeed1
##                             -1.2875
##                       InkViscosity1
##                              1.2500
##                          DwellTime1
##                             -0.0500
##        SqueegeeSpeed1:InkViscosity1
##                             -1.5125
##           SqueegeeSpeed1:DwellTime1
##                             -0.2125
##            InkViscosity1:DwellTime1
##                              0.1500
## SqueegeeSpeed1:InkViscosity1:DwellTime1
##                              0.0875
```

```
DanielPlot(des.resp)
MEPlot(des.resp)
IAPlot(des.resp)
```







There is an interaction between A and B. At low B there is little A effect while at high B there is a large positive A effect.

2. TR 18532:2009 12.1.4.2.2 Pages 150-154

> This example shows an application of experimental design 2 of Table 32. It has two roles; one, as a design development tool to determine the suitability of a sintered part for a particular application and two, as a development tool in the sense of searching for preferred operating conditions. Four design factors were investigated each at two levels as indicated in Table 3.
>
> The experimental layout uses columns 1, 2, 4, and 7 of a standard $L8$ array. Strength of fit, in kN, at minimum interference conditions was recorded for each part subjected to each experimental combination.
>
> Three parts were used for each run in order to separate means from variation in order to permit a search for design factors that would enhance mean strength (signal factors) and those that would reduce variation (control factors). Variation is expressed in terms of standard deviation. The results are given in Table 4.

Design Factor	Level 1	Level 2
A: Surface finish	Fine turned	Microlled
B: Lubrication	Yes–number 2 oil	No
C: Speed	Low	High
D: Density	6.5	6.8

Table 3: Sintered part design factors and their levels

(a) Create a 2^{4-1} design.

ESQUANT
Statistical Consulting

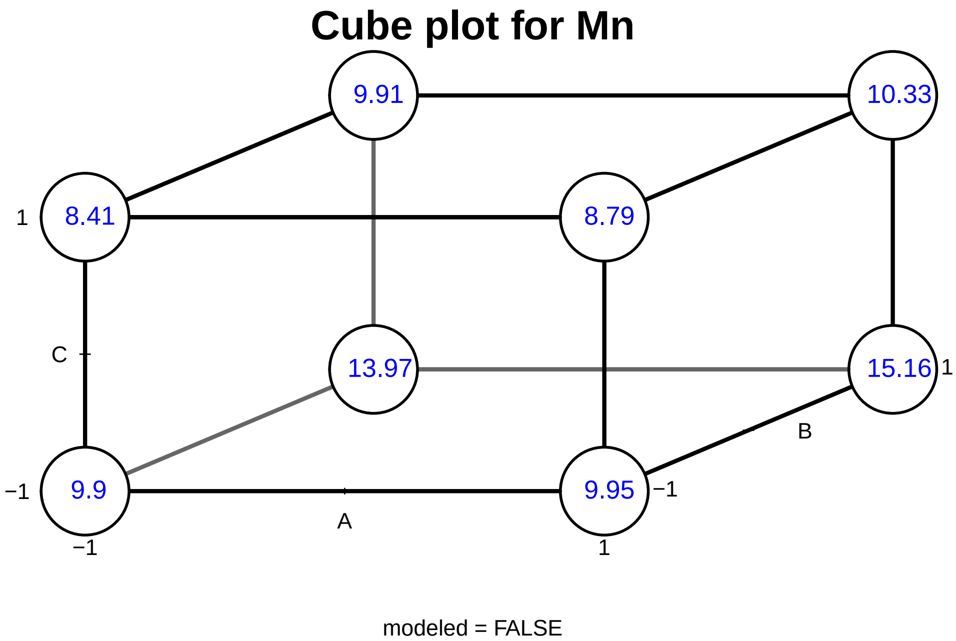
Run	A	B	C	D	Result1	Result2	Result3	Mean	StdDev
1	-1	-1	-1	-1	12.7	7.27	9.74	9.9	2.72
2	1	-1	-1	1	10.36	10.45	9.05	9.95	0.78
3	-1	1	-1	1	12.61	15.19	14.11	13.97	1.3
4	1	1	-1	-1	16.8	14.76	13.92	15.16	1.48
5	-1	-1	1	1	9.41	8.52	7.29	8.41	1.06
6	1	-1	1	-1	7.45	8.9	10.02	8.79	1.29
7	-1	1	1	-1	13.99	7.65	8.1	9.91	3.54
8	1	1	1	1	11.52	13.92	10.33	10.33	0.74

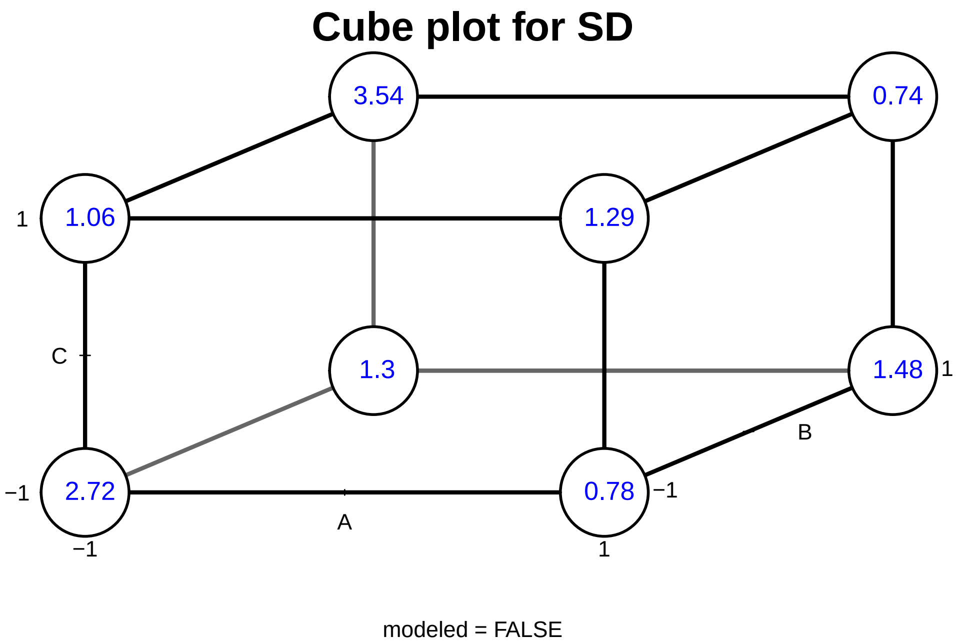
Table 4: Results from the runs on sintered part experiment

```
##    A  B  C  D    Mn
## 1 -1 -1 -1 -1  9.90
## 2  1 -1 -1  1  9.95
## 3 -1  1 -1  1 13.97
## 4  1  1 -1 -1 15.16
## 5 -1 -1  1  1  8.41
## 6  1 -1  1 -1  8.79
## 7 -1  1  1 -1  9.91
## 8  1  1  1  1 10.33
## class=design, type= FrF2
##    A  B  C  D   SD
## 1 -1 -1 -1 -1 2.72
## 2  1 -1 -1  1 0.78
## 3 -1  1 -1  1 1.30
## 4  1  1 -1 -1 1.48
## 5 -1 -1  1  1 1.06
## 6  1 -1  1 -1 1.29
## 7 -1  1  1 -1 3.54
## 8  1  1  1  1 0.74
## class=design, type= FrF2
```

(b) Make cube plots of the data.

```
cubePlot(lm(SinteredPart.Mn, degree=3),
         "A","B","C",modeled=F)
cubePlot(lm(SinteredPart.SD, degree=3),
         "A","B","C",modeled=F)
```

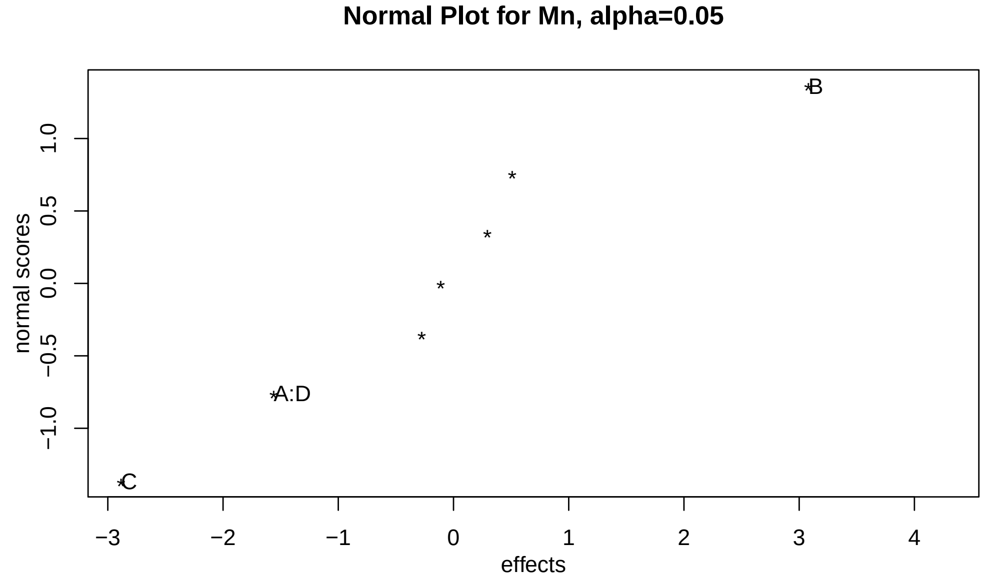


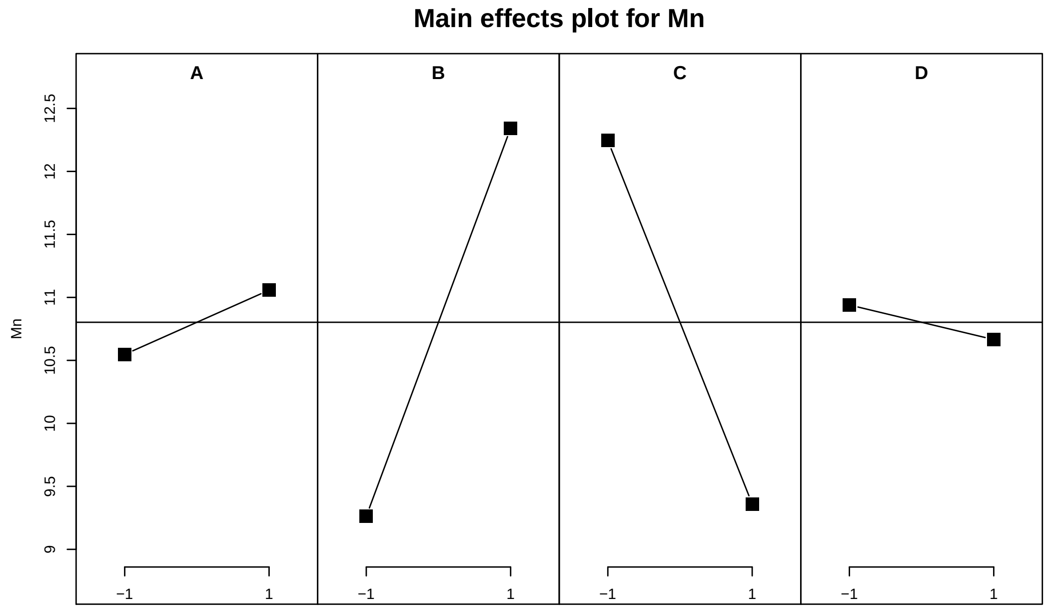


(c) Analyse the experiment.

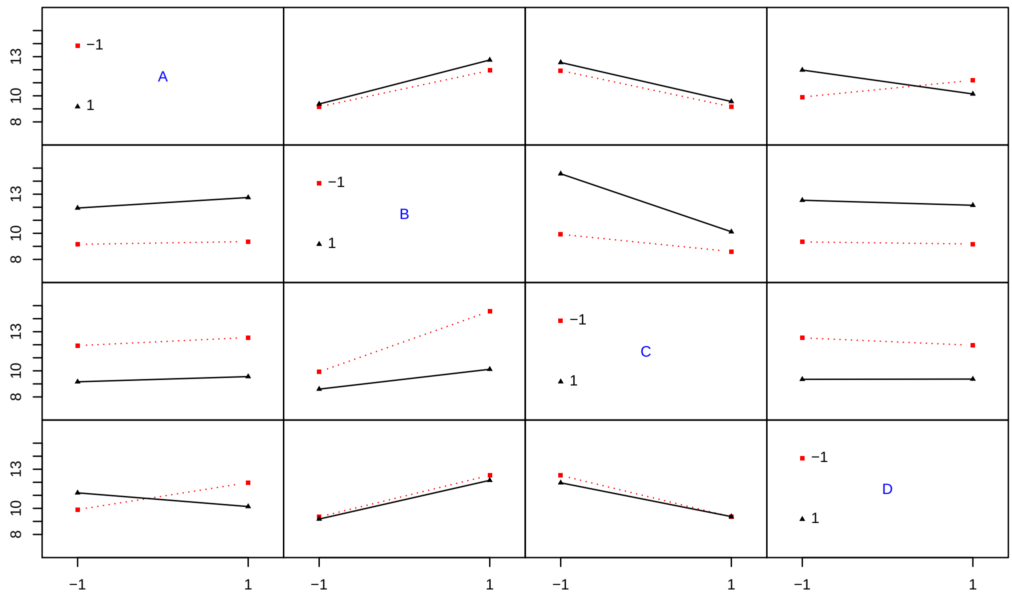
```
coef(lm(SinteredPart.Mn, degree=2))

## (Intercept)          A1          B1          C1          D1       A1:B1
##     10.8025      0.2550      1.5400     -1.4425     -0.1375      0.1475
##       A1:C1       A1:D1
##     -0.0550     -0.7800

DanielPlot(SinteredPart.Mn)
MEPlot(SinteredPart.Mn)
IAPlot(SinteredPart.Mn)
coef(lm(SinteredPart.SD, degree=2))
```

```
## (Intercept)          A1          B1          C1          D1       A1:B1
##     1.61375    -0.54125     0.15125     0.04375    -0.64375    -0.11375
##       A1:C1       A1:D1
##    -0.10125     0.33125
DanielPlot(SinteredPart.SD)
MEPlot(SinteredPart.SD)
IAPlot(SinteredPart.SD)
```

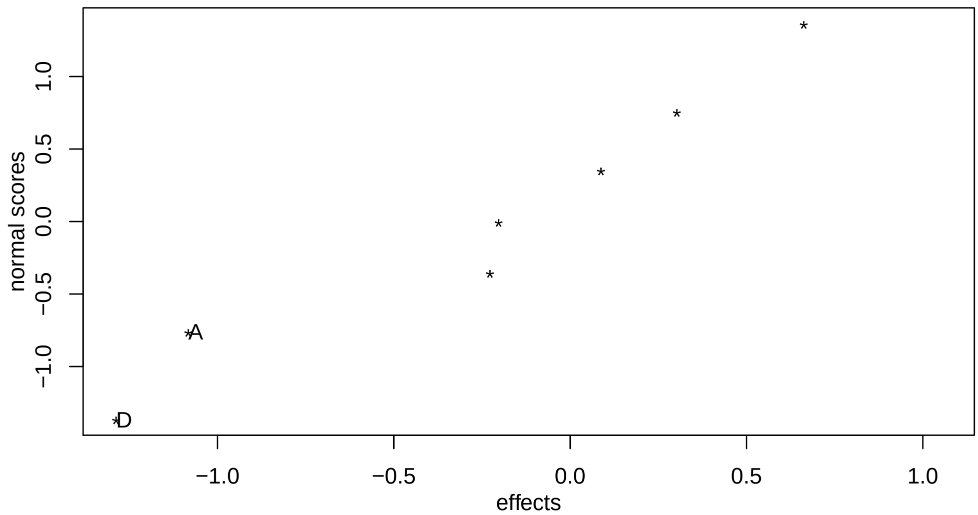


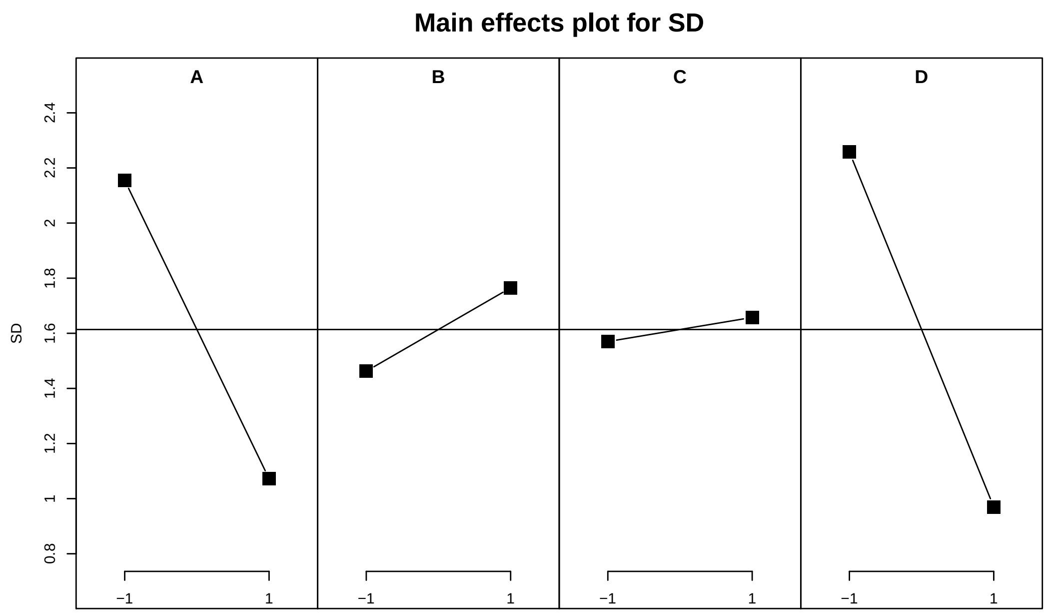


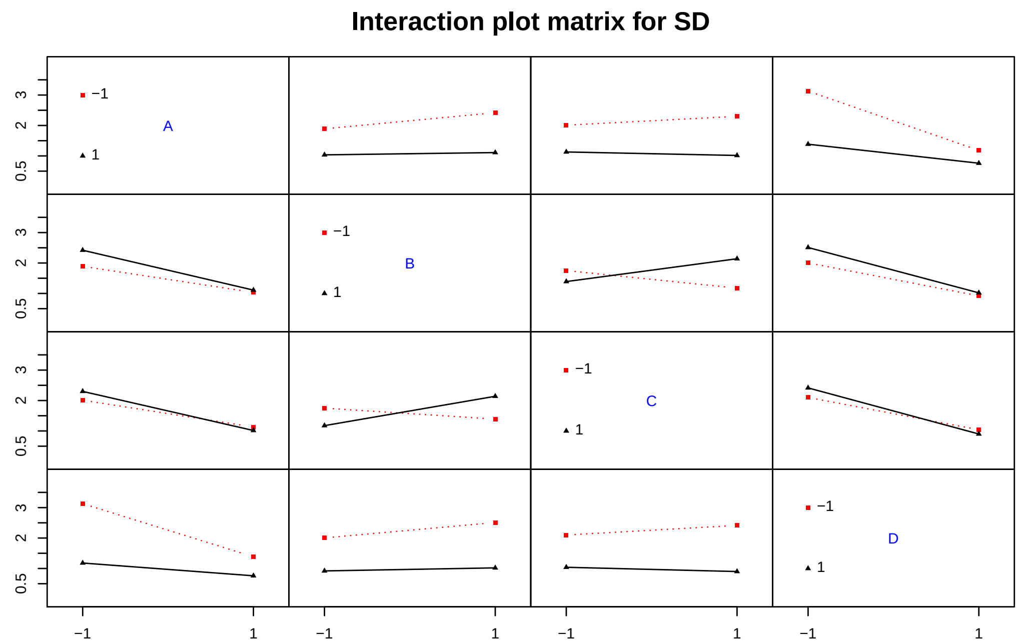
Interaction plot matrix for Mn



Normal Plot for SD, alpha=0.05







For means, factor B has a large positive effect while factor C has a large negative effect. There may be an interaction between B and C (which is aliased with AD). There do not appear to be any noticeable effects for standard deviations.

3. The Bush Experiment

> In a VU student project conducted by Peter Kostaridis and Nick Condilis, an experiment was carried out to improve the rubber composition of the bush, an important part of the suspension system, used in a locally manufactured car. The levels of six components were varied and the resulting compound was tested to determine the Loss Angle and Dynamic Stiffness. The factors have

been coded as 301 (A), 302 (B), 303 (C), 304 (D), 308 (E), and 309 (F) for confidentiality reasons and the levels given as $-$ and $+$. The design and results are given in Table 5.

Run	A	B	C	D	E	F	Loss Angle	Dynamic Stiffness
1	$-$	$-$	$-$	$-$	$-$	$-$	5.45	520
2	$+$	$-$	$-$	$+$	$+$	$+$	4.55	501
3	$-$	$+$	$-$	$+$	$+$	$-$	7.23	864
4	$+$	$+$	$-$	$-$	$-$	$+$	6.11	999
5	$-$	$-$	$+$	$-$	$+$	$+$	4.93	573
6	$+$	$-$	$+$	$+$	$-$	$-$	6.37	523
7	$-$	$+$	$+$	$+$	$-$	$+$	5.59	946
8	$+$	$+$	$+$	$-$	$+$	$-$	7.72	686

Table 5: Design and results for the Bush Experiment

(a) Read the design and results into R and set up the Experimental Design.

```
Bush <- read.csv("Bush1.csv")
lm(D ~ A*B*C, data=Bush)

##
## Call:
## lm.default(formula = D ~ A * B * C, data = Bush)
##
## Coefficients:
## (Intercept)            A            B            C          A:B
##   2.790e-48    1.743e-32    1.743e-32   -2.369e-32   -1.000e+00
##         A:C          B:C        A:B:C
##   1.006e-16    1.006e-16   -7.200e-17

lm(E ~ A*B*C, data=Bush)

##
## Call:
## lm.default(formula = E ~ A * B * C, data = Bush)
##
## Coefficients:
## (Intercept)            A            B            C          A:B
##           0            0            0            0            0
##         A:C          B:C        A:B:C
##           0            0            1

lm(FF ~ A*B*C, data=Bush)
```

```
##
## Call:
## lm.default(formula = FF ~ A * B * C, data = Bush)
##
## Coefficients:
## (Intercept)            A            B            C          A:B
##   0.000e+00   -7.865e-33    8.217e-33    0.000e+00    8.716e-33
##         A:C          B:C        A:B:C
##  -1.000e+00   -2.004e-16   -9.569e-17
```

```
BushDesign <- FrF2(8, 6, randomize=FALSE,
     generators=c("-AB", "ABC", "-AC"))

BushDesign.LA <- add.response(BushDesign, Bush$LossAngle)
BushDesign.LA
```

```
##    A  B  C  D  E  F Bush.LossAngle
## 1 -1 -1 -1 -1 -1 -1           5.45
## 2  1 -1 -1  1  1  1           4.55
## 3 -1  1 -1  1  1 -1           7.23
## 4  1  1 -1 -1 -1  1           6.11
## 5 -1 -1  1 -1  1  1           4.93
## 6  1 -1  1  1 -1 -1           6.37
## 7 -1  1  1  1 -1  1           5.56
## 8  1  1  1 -1  1 -1           7.72
## class=design, type= FrF2.generators
```

```
BushDesign.DS <- add.response(BushDesign, Bush$DynamicStiffness)
BushDesign.DS
```

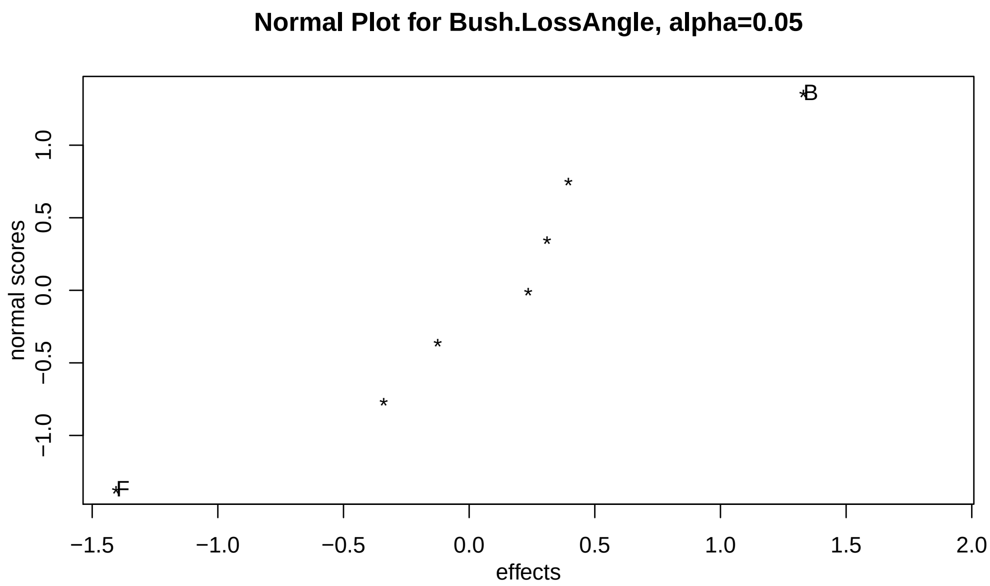
```
##    A  B  C  D  E  F Bush.DynamicStiffness
## 1 -1 -1 -1 -1 -1 -1                   520
## 2  1 -1 -1  1  1  1                   501
## 3 -1  1 -1  1  1 -1                   864
## 4  1  1 -1 -1 -1  1                   999
## 5 -1 -1  1 -1  1  1                   573
## 6  1 -1  1  1 -1 -1                   523
## 7 -1  1  1  1 -1  1                   946
## 8  1  1  1 -1  1 -1                   686
## class=design, type= FrF2.generators
```

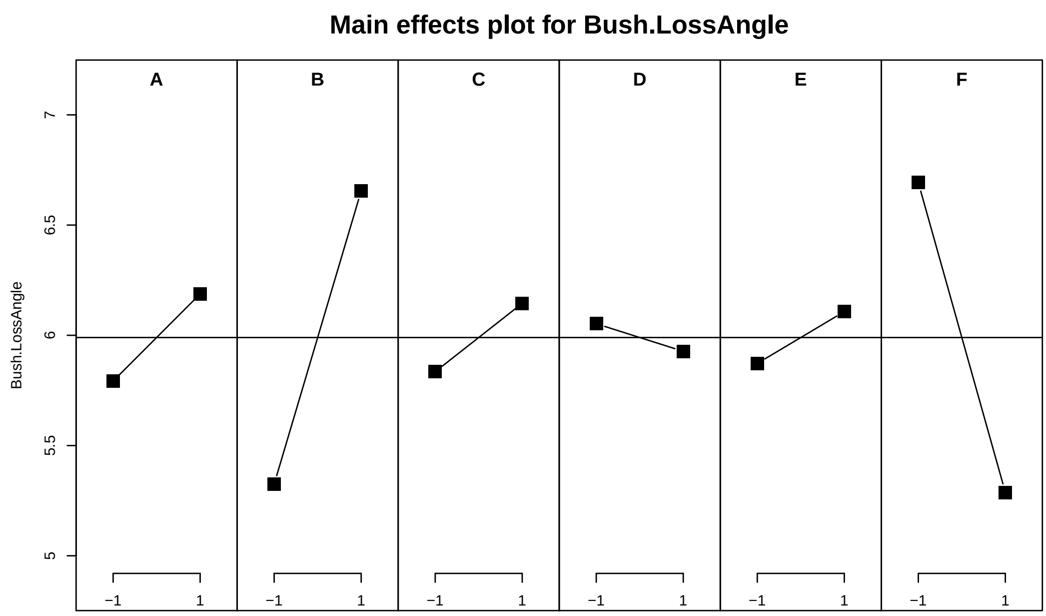
(b) Analyse the experiment.

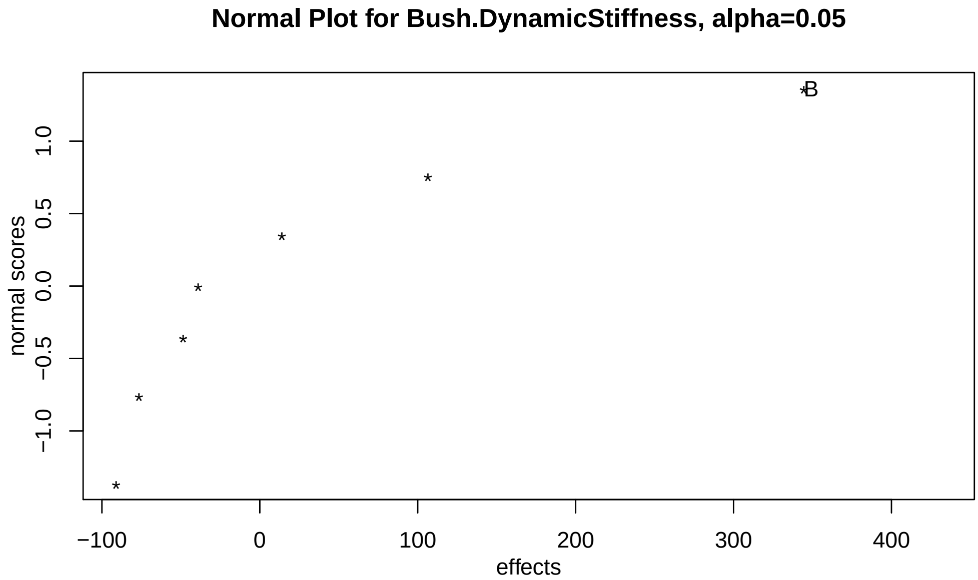
```
coef(lm(BushDesign.LA, degree=2))
```

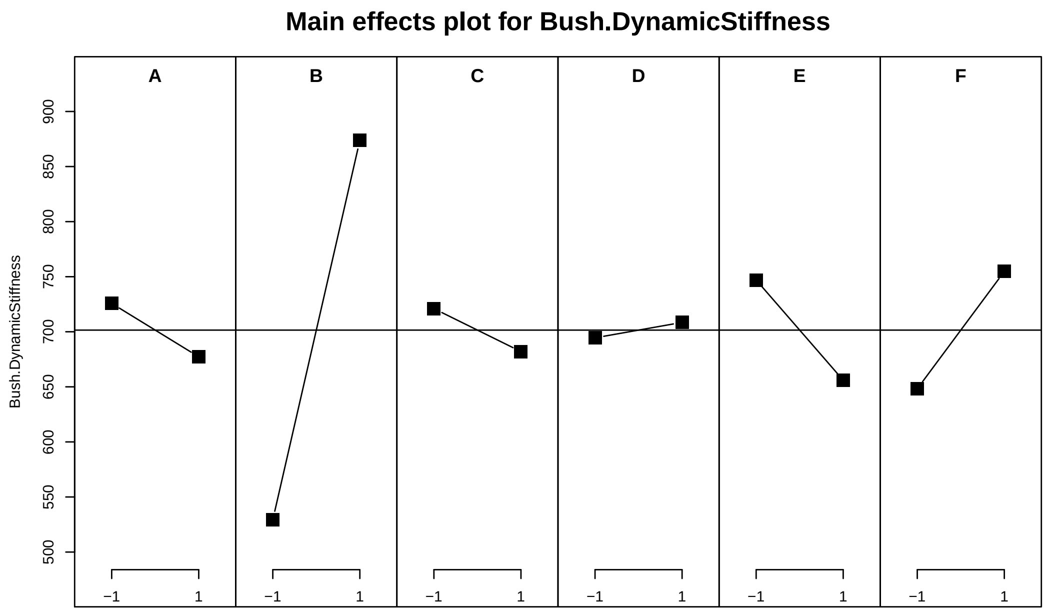
```
## (Intercept)          A1          B1          C1          D1          E1
##      5.9900      0.1975      0.6650      0.1550     -0.0625      0.1175
##          F1       A1:E1
##     -0.7025     -0.1700
```

```r
DanielPlot(BushDesign.LA)
MEPlot(BushDesign.LA)

coef(lm(BushDesign.DS, degree=2))

## (Intercept)          A1          B1          C1          D1          E1
##      701.50      -24.25      172.25      -19.50        7.00      -45.50
##          F1       A1:E1
##       53.25      -38.25

DanielPlot(BushDesign.DS)
MEPlot(BushDesign.DS)
```









For Loss Angle, factor B has a large positive effect while factor B has a large negative effect. For Dynamic Stiffness, factor B has a large negative effect.

4. Explosives Development

> In the development of a new explosive, a set of experiments were run involving five factors with A, Age of Gum; B, Age of Thiourea; C, pH controlled; D, Aluminium level; and E, Crosslinker Level. The response was the Gel Strength after 10 minutes. The design and results are given in Table 6.

(a) Read the design and results into R and set up the Experimental Design.

A	B	C	D	E	Gel Strength (10 Mins)
−	−	−	−	+	469
+	−	−	−	−	330
−	+	−	−	−	266
+	+	−	−	+	351
−	−	+	−	−	316
+	−	+	−	+	522
−	+	+	−	+	357
+	+	+	−	−	430
−	−	−	+	−	293
+	−	−	+	+	708
−	+	−	+	+	267
+	+	−	+	−	341
−	−	+	+	+	502
+	−	+	+	−	453
−	+	+	+	−	197
+	+	+	+	+	568

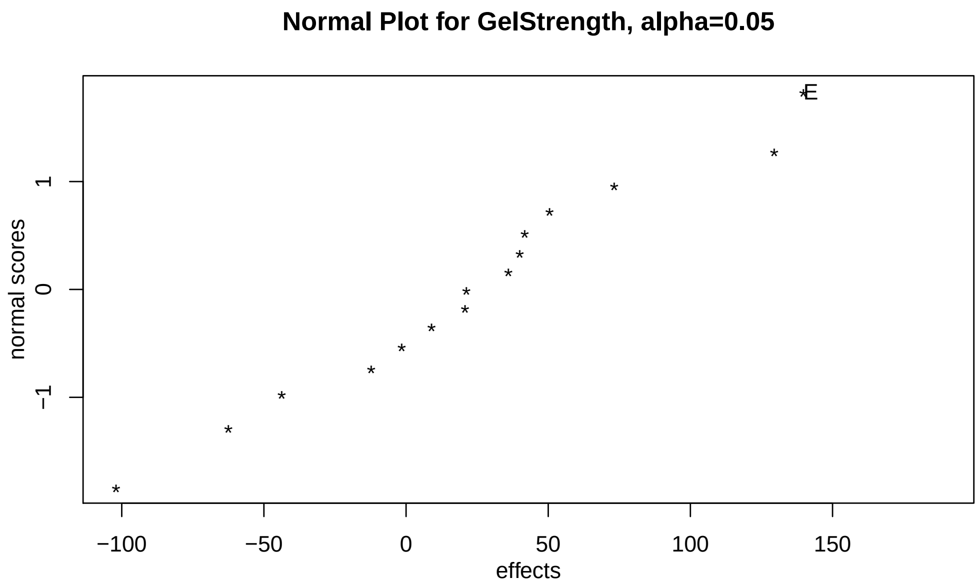
Table 6: Design and results for the Explosives Development Experiment

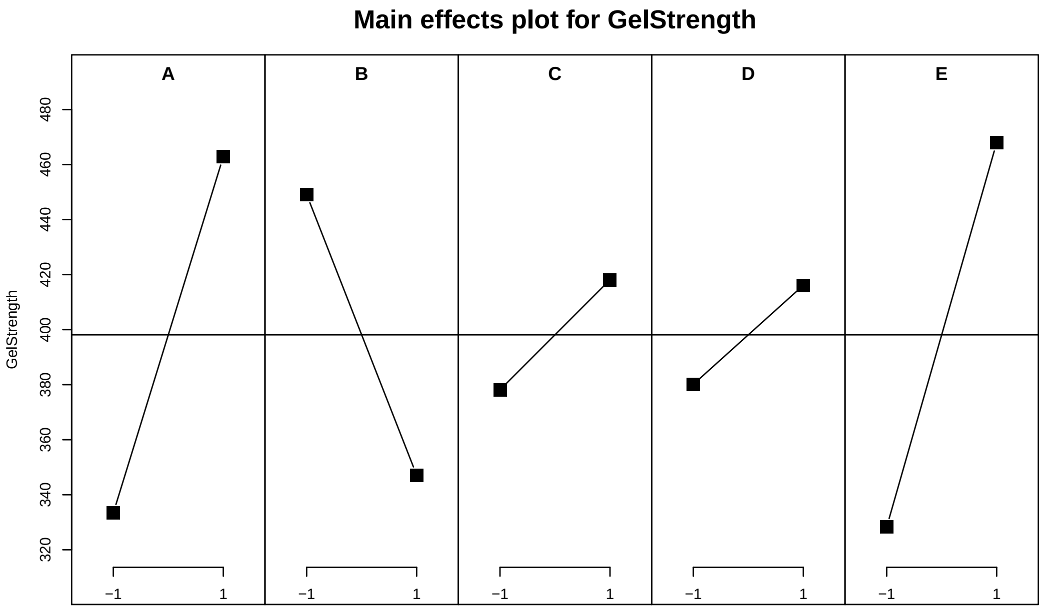
```
GelStrength <- c(469, 330, 266, 351,
                 316, 522, 357, 430,
                 293, 708, 267, 341,
                 502, 453, 197, 568)
Explosives <- FrF2(16, 5, randomize = F)
Explosives.resp <- add.response(Explosives, response=GelStrength)
Explosives.resp

##     A  B  C  D  E GelStrength
## 1  -1 -1 -1 -1  1         469
## 2   1 -1 -1 -1 -1         330
## 3  -1  1 -1 -1 -1         266
## 4   1  1 -1 -1  1         351
## 5  -1 -1  1 -1 -1         316
## 6   1 -1  1 -1  1         522
## 7  -1  1  1 -1  1         357
## 8   1  1  1 -1 -1         430
## 9  -1 -1 -1  1 -1         293
## 10  1 -1 -1  1  1         708
## 11 -1  1 -1  1  1         267
## 12  1  1 -1  1 -1         341
## 13 -1 -1  1  1  1         502
## 14  1 -1  1  1 -1         453
## 15 -1  1  1  1 -1         197
## 16  1  1  1  1  1         568
## class=design, type= FrF2
```

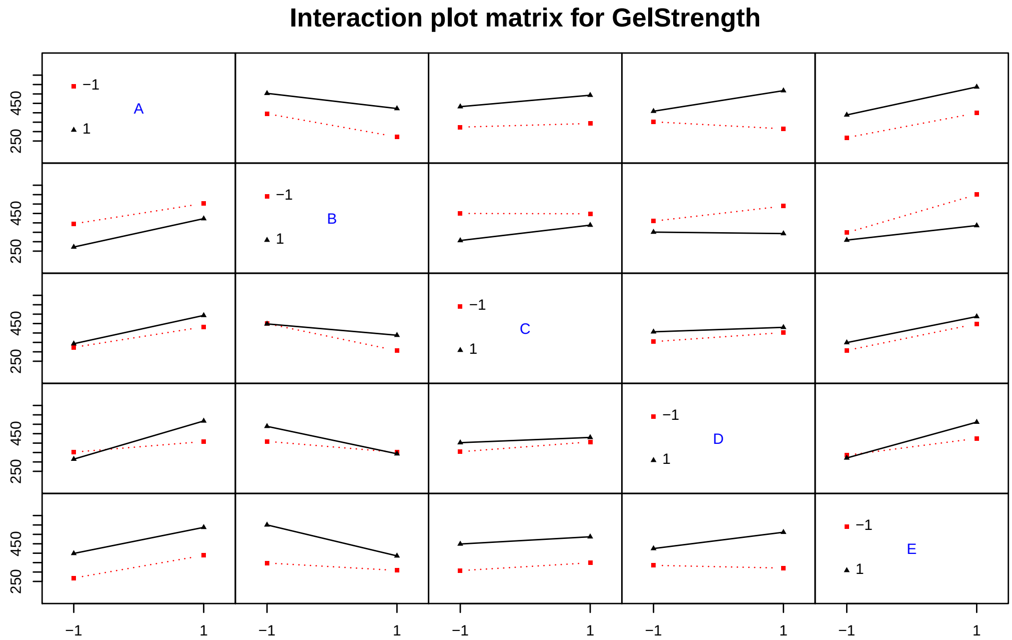
(b) Analyse the experiment.

```
coef(lm(Explosives.resp, degree=2))

## (Intercept)          A1          B1          C1          D1          E1
```

```
##      398.125     64.750     -51.000     20.000     18.000     69.875
##        A1:B1      A1:C1       A1:D1      A1:E1      B1:C1      B1:D1
##       10.625     10.375      36.625      4.500     20.875    -21.875
##        B1:E1      C1:D1       C1:E1      D1:E1
##      -31.250     -6.125      -0.750     25.250

DanielPlot(Explosives.resp)
MEPlot(Explosives.resp)
IAPlot(Explosives.resp)
```

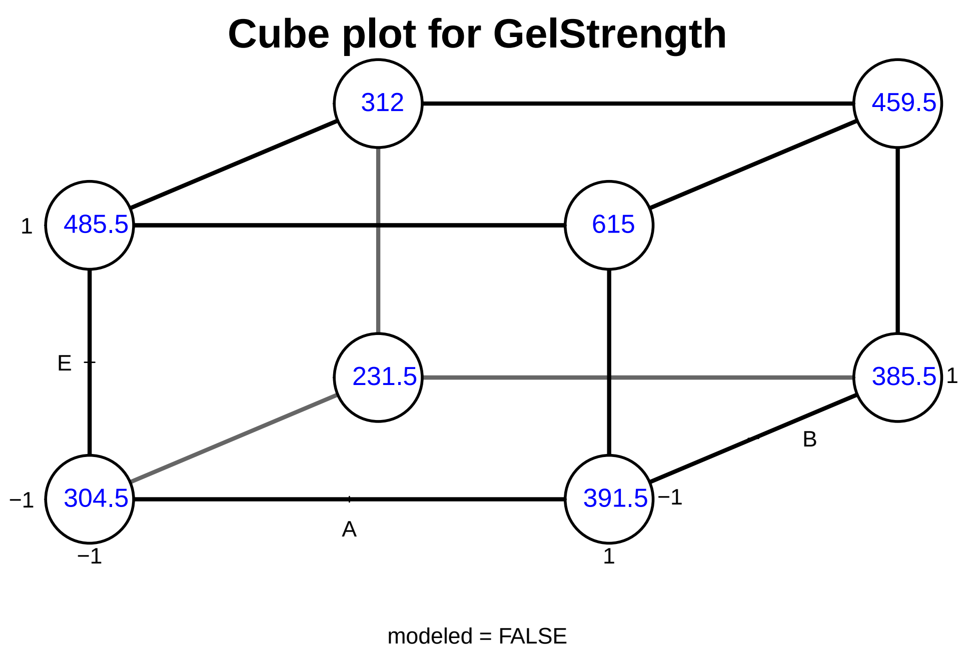






(c) Make a cube plot.

```
cubePlot(lm(Explosives.resp, degree=2),"A","B","E",
        modeled=F)
```



The results are inconclusive. Possiblr factors E, A, and B have significant effects. Follow up experiments should be suggested.

5. Injection Molding Experiment

> In an injection molding experiment eight variables were studied in 16 runs. The response was percentage shrinkage. The variables are given in Table 6 while the

design and results are given in Table 7.

A	Mold Temperature
B	Moisture Content
C	Holding Pressure
D	Cavity Thickness
E	Booster Pressure
F	Cycle Time
G	Gate Size
H	Screw Speed

Table 7: Variables for Injection Molding Experiment

Run	A	B	C	D	E	F	G	H	Shrinkage
1	−	−	−	−	−	−	−	−	20.3
2	+	−	−	−	−	+	+	+	16.8
3	−	+	−	−	+	−	+	+	15.0
4	+	+	−	−	+	+	−	+	15.9
5	−	−	+	−	+	+	+	−	17.5
6	+	−	+	−	+	−	−	+	24.0
7	−	+	+	−	−	+	−	+	27.4
8	+	+	+	−	−	−	+	−	22.3
9	−	−	−	+	+	+	−	−	14.0
10	+	−	−	+	+	−	+	+	16.7
11	−	+	−	+	−	+	+	−	21.9
12	+	+	−	+	−	−	−	+	15.4
13	−	−	+	+	−	−	+	+	27.6
14	+	−	+	+	−	+	−	−	21.5
15	−	+	+	+	+	−	−	−	17.1
16	+	+	+	+	+	+	+	+	22.6

Table 8: Design and results for the Injection Molding Experiment

(a) Read the design and results into R and set up the Experimental Design.

```
InjectionMolding <- read.csv("InjectionMolding.csv")
lm(E ~ (A*B*C*D), data=InjectionMolding)

##
## Call:
## lm.default(formula = E ~ (A * B * C * D), data = InjectionMolding)
##
## Coefficients:
## (Intercept)            A            B            C            D
##           0            0            0            0            0
```

```
##          A:B          A:C          B:C          A:D          B:D
##            0            0            0            0            0
##          C:D        A:B:C        A:B:D        A:C:D        B:C:D
##            0            0            0            0            1
##      A:B:C:D
##            0
```

```r
lm(FF ~ (A*B*C*D), data=InjectionMolding)
```

```
##
## Call:
## lm.default(formula = FF ~ (A * B * C * D), data = InjectionMolding)
##
## Coefficients:
## (Intercept)            A            B            C            D
##   0.000e+00   -6.163e-32    2.465e-32    2.583e-32    3.465e-32
##         A:B          A:C          B:C          A:D          B:D
##  -2.091e-34   -3.698e-32   -9.861e-32   -1.356e-31    6.779e-32
##         C:D        A:B:C        A:B:D        A:C:D        B:C:D
##  -1.479e-31    9.861e-32    3.444e-32    1.000e+00   -1.589e-16
##     A:B:C:D
##  -1.551e-16
```

```r
lm(G ~ (A*B*C*D), data=InjectionMolding)
```

```
##
## Call:
## lm.default(formula = G ~ (A * B * C * D), data = InjectionMolding)
##
## Coefficients:
## (Intercept)            A            B            C            D
##   0.000e+00   -3.081e-32   -1.233e-32    9.861e-32    2.465e-32
##         A:B          A:C          B:C          A:D          B:D
##  -7.396e-32    2.465e-32   -7.396e-32   -6.163e-32    5.345e-33
##         C:D        A:B:C        A:B:D        A:C:D        B:C:D
##  -1.972e-31    1.000e+00   -1.759e-17    1.174e-16   -4.823e-17
##     A:B:C:D
##   2.318e-17
```

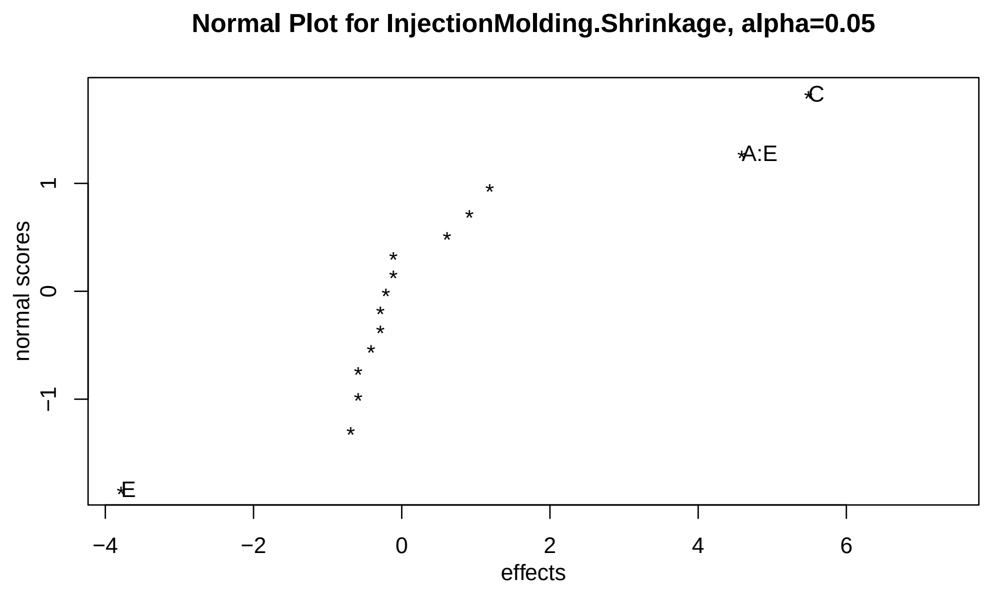
```r
lm(H ~ (A*B*C*D), data=InjectionMolding)
```

```
##
## Call:
## lm.default(formula = H ~ (A * B * C * D), data = InjectionMolding)
##
## Coefficients:
## (Intercept)            A            B            C            D
##   0.000e+00   -2.465e-32   -1.233e-32    3.698e-32   -1.233e-32
##         A:B          A:C          B:C          A:D          B:D
##   2.465e-32    2.465e-32   -2.465e-32   -1.592e-32   -1.064e-63
```

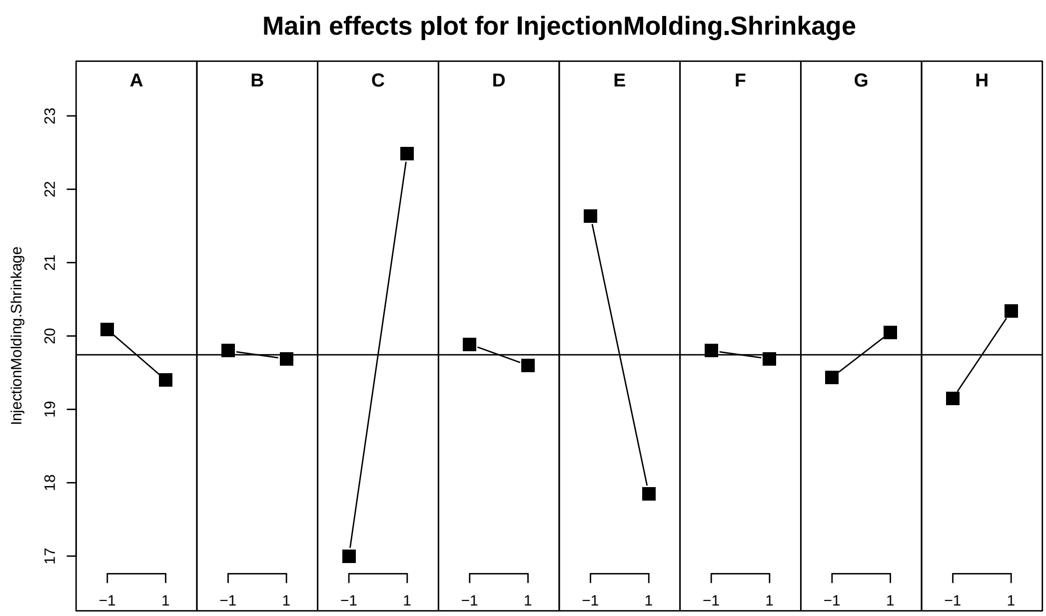
```
##          C:D        A:B:C        A:B:D        A:C:D        B:C:D
##   -1.916e-47    4.314e-32    1.000e+00    1.020e-16    4.588e-17
##      A:B:C:D
##   -1.103e-16

InjectionMoldingDesign <- FrF2(16, 8, randomize=FALSE,
     generators=c("BCD", "ACD", "ABC", "ABD"))

InjectionMolding.resp <- add.response(InjectionMoldingDesign,
                            InjectionMolding$Shrinkage)
InjectionMolding.resp

##     A  B  C  D  E  F  G  H InjectionMolding.Shrinkage
## 1  -1 -1 -1 -1 -1 -1 -1 -1                       20.3
## 2   1 -1 -1 -1 -1  1  1  1                       16.8
## 3  -1  1 -1 -1  1 -1  1  1                       15.0
## 4   1  1 -1 -1  1  1 -1 -1                       15.9
## 5  -1 -1  1 -1  1  1  1 -1                       17.5
## 6   1 -1  1 -1  1 -1 -1  1                       24.0
## 7  -1  1  1 -1 -1  1 -1  1                       27.3
## 8   1  1  1 -1 -1 -1  1 -1                       22.3
## 9  -1 -1 -1  1  1  1 -1  1                       14.0
## 10  1 -1 -1  1  1 -1  1 -1                       16.7
## 11 -1  1 -1  1 -1  1  1 -1                       21.9
## 12  1  1 -1  1 -1 -1 -1  1                       15.4
## 13 -1 -1  1  1 -1 -1  1  1                       27.6
## 14  1 -1  1  1 -1  1 -1 -1                       21.5
## 15 -1  1  1  1  1 -1 -1 -1                       17.1
## 16  1  1  1  1  1  1  1  1                       22.6
## class=design, type= FrF2.generators
```

(b) Analyse the experiment.

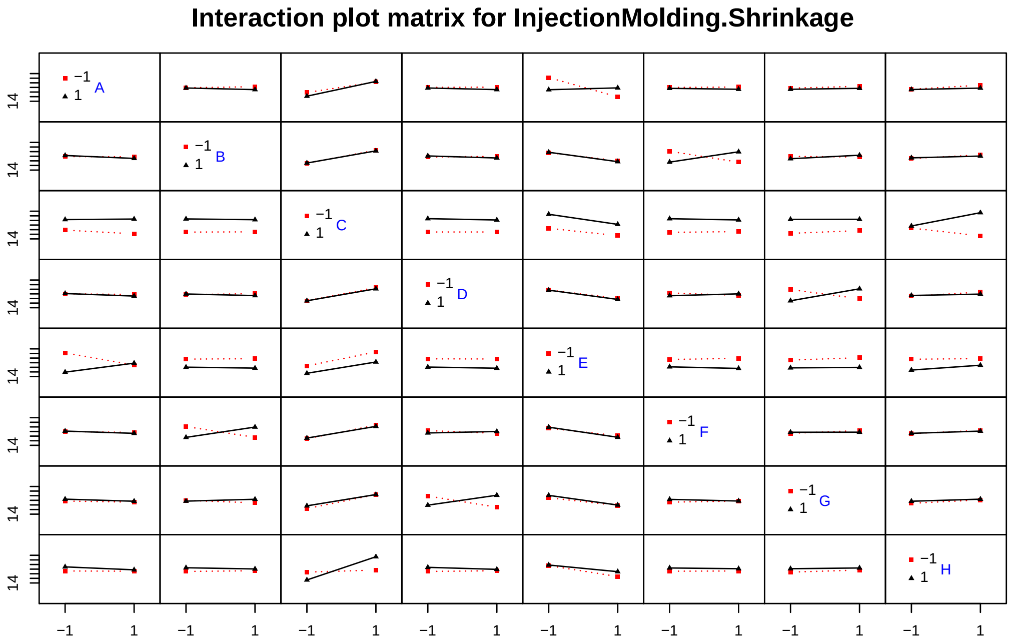
```
coef(lm(InjectionMolding.resp, degree=2))

## (Intercept)          A1          B1          C1          D1          E1
##    19.74375    -0.34375    -0.05625     2.74375    -0.14375    -1.89375
##          F1          G1          H1       A1:B1       A1:C1       A1:D1
##    -0.05625     0.30625     0.59375    -0.29375     0.45625    -0.20625
##       A1:E1       A1:F1       A1:G1       A1:H1
##     2.29375    -0.14375    -0.10625    -0.29375

DanielPlot(InjectionMolding.resp)
MEPlot(InjectionMolding.resp)
IAPlot(InjectionMolding.resp)
```



Normal Plot for InjectionMolding.Shrinkage, alpha=0.05
normal scores
effects
C
A:E
E
−4
−2
0
2
4
6
−1
0
1



Main effects plot for InjectionMolding.Shrinkage
InjectionMolding.Shrinkage
A
B
C
D
E
F
G
H
17
18
19
20
21
22
23
−1
1



C and E have significant effects with a significant interaction string involving AE+BF+CH+DG. Follow up experiments are probably required.

6. Response Surface Example TR18532:2009 Pages 156-158.

> Technical and operational considerations indicate that three factors, gas ratio, power and pulse may influence oxide uniformity. Non-linearity and interactions are expected so each factor was investigated at three levels using an 18 run central composite (face-centred cube) design. The three levels of each factor were coded −1, 0, and 1 for convenience. Results are shown in Table 8.

(a) Read the design and results into R and set up the Experimental Design.

```
oxide <- read.csv("rsmexample.csv")
```

(b) Analyse the experiment.

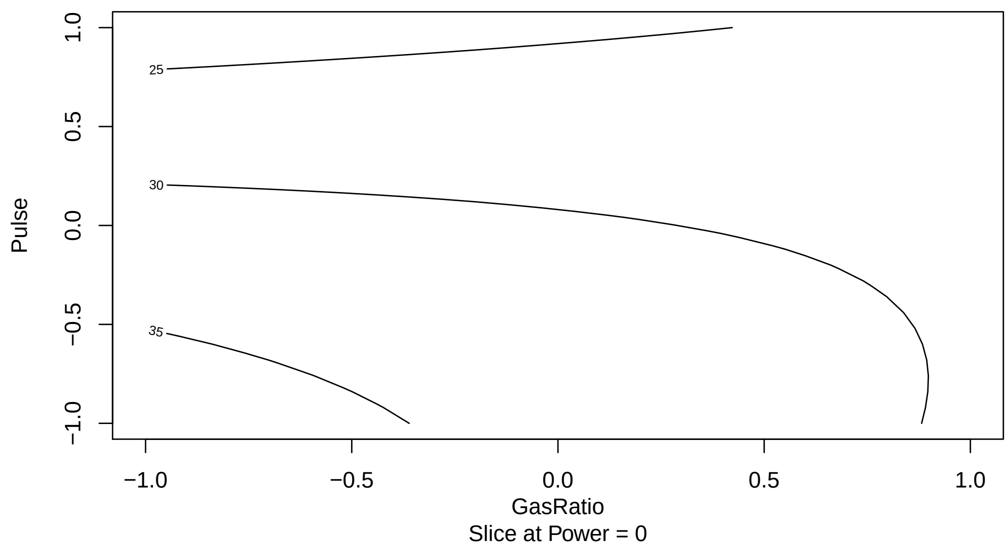
```
library(rsm)
oxide.rsm <- rsm(OxideUniformity ~
                   SO(GasRatio,Pulse, Power), data=oxide)
summary(oxide.rsm)

##
## Call:
## rsm(formula = OxideUniformity ~ SO(GasRatio, Pulse, Power), data = oxide)
##
##                   Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)      30.379762   0.484889  62.6531 4.683e-12 ***
## GasRatio         -1.280000   0.389771  -3.2840 0.0111216 *
```

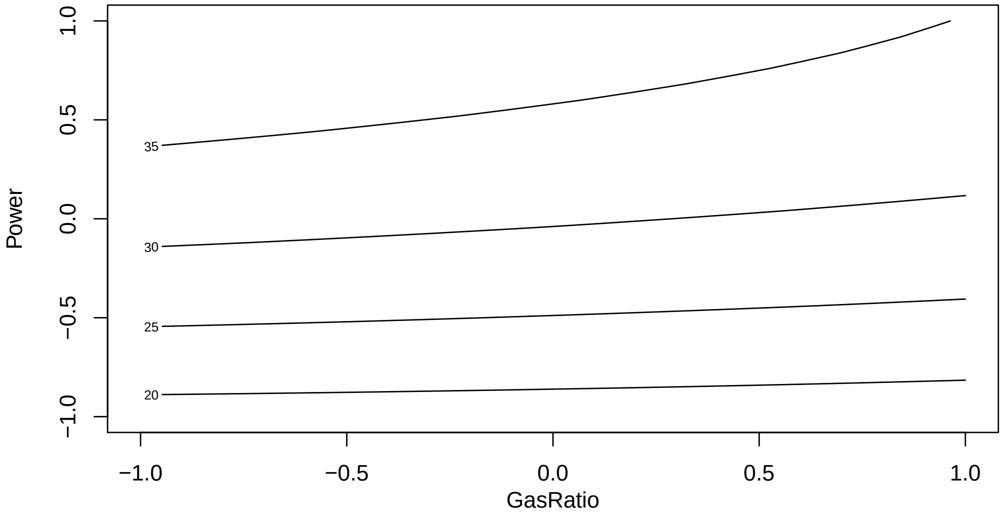
Gas Ratio	Pulse	Power	Oxide Uniformity
0	0	0	29.4
0	0	0	32.1
0	0	0	31.5
0	0	0	30.9
−1	−1	−1	16.9
−1	1	−1	17.2
−1	1	1	22.7
−1	−1	1	52.4
1	−1	−1	10.7
1	1	−1	22.6
1	1	1	23.8
1	−1	1	43.5
0	−1	0	32.7
0	0	−1	16.4
0	1	0	24.1
0	0	1	37.5
−1	0	0	27.6
1	0	0	31.8

Table 9: Results for experiment runs on etching process

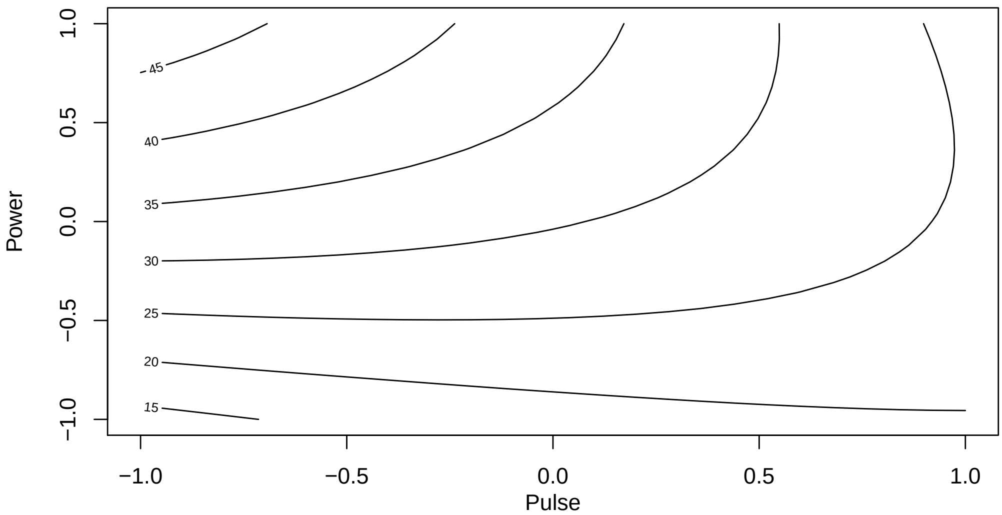
```
## Pulse          -4.580000   0.389771 -11.7505 2.516e-06 ***
## Power           9.610000   0.389771  24.6555 7.825e-09 ***
## GasRatio:Pulse  2.700000   0.435778   6.1958 0.0002606 ***
## GasRatio:Power -0.875000   0.435778  -2.0079 0.0795361 .
## Pulse:Power    -7.700000   0.435778 -17.6696 1.076e-07 ***
## GasRatio^2     -0.084524   0.748775  -0.1129 0.9129047
## Pulse^2        -1.384524   0.748775  -1.8491 0.1016243
## Power^2        -2.834524   0.748775  -3.7855 0.0053456 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.9931,Adjusted R-squared:  0.9854
## F-statistic: 128.3 on 9 and 8 DF,  p-value: 1.107e-07
##
## Analysis of Variance Table
##
## Response: OxideUniformity
##                              Df  Sum Sq Mean Sq  F value    Pr(>F)
## FO(GasRatio, Pulse, Power)    3 1149.67  383.22 252.2503 2.935e-08
## TWI(GasRatio, Pulse, Power)   3  538.76  179.59 118.2111 5.777e-07
## PQ(GasRatio, Pulse, Power)    3   65.19   21.73  14.3034  0.001403
## Residuals                     8   12.15    1.52
## Lack of fit                   5    8.13    1.63   1.2106  0.466502
## Pure error                    3    4.03    1.34
##
## Stationary point of response surface:
```

```
##  GasRatio     Pulse     Power
## 3.0761730 0.7375252 0.2186266
##
## Eigenanalysis:
## eigen() decomposition
## $values
## [1]  2.4915215 -0.7201349 -6.0749580
##
## $vectors
##                [,1]       [,2]        [,3]
## GasRatio  0.4599932  0.8832880 -0.09060114
## Pulse     0.7012337 -0.2987889  0.64729942
## Power    -0.5446812  0.3612859  0.75683215

contour(oxide.rsm, ~ GasRatio+Pulse+ Power)
```





Slice at Pulse = 0



Slice at GasRatio = 0