Factorial Designs at Two Levels

David M. Rocke

February 9, 18, 2026

Factorial Designs

- A *factorial design* with p factors each of which is defined at a discrete number of options is one in which all possible combinations of the choices for each factor are tested.
- In this context, a factor can be actual discrete options (catalyst A, catalyst B, catalyst C) or distinct numerical setting (temp 250 °C, 260 °C, 270 °C). A factorial design with these choices would run each of the nine combinations of factor levels, one each, or perhaps twice each.

Factorial Designs

- A simple and efficient way to start is with all the factors at two levels (catalyst A, catalyst B), (temp 250 °C, 270 °C).
- This will give some idea of whether process yield goes up or down with temperature and this can be refined in subsequent experiments.
- With p factors each at two levels, the number of runs is a multiple of 2^p.
- So with 5 factors, there are 32 runs, or 64 if replicated.

A Conceptual Example

- Suppose that the clarity of floor wax is influenced by the formulation, with factors and a binary qualitative readout of Clear/Cloudy

Name	Definition of Factor	Levels
A	Amount of Emulsifier A less/more	$-1/1$
B	Amount of Emulsifier B less/more	$-1/1$
C	Amount of Catalyst C less/more	$-1/1$

- We can code the readout as 0/1 and incorporate all this into a data frame. This is example 1 in section 5.2 of the text.

```
> film
   A  B  C clarity
1 -1 -1 -1       1
2  1 -1 -1       1
3 -1  1 -1       0
4  1  1 -1       0
5 -1 -1  1       1
6  1 -1  1       1
7 -1  1  1       0
8  1  1  1       0
> film.lm1 <- lm(clarity~A+B+C,data=film)
```

Note the pattern of factor levels in the 8 runs. The first listed one has alternating -1 and $+1$, the second has alternating repeats $-1, -1$ and $+1$, $+1$, and the third has first 4 of -1 and then 4 of $+1$. Also, the dot product of any two is zero, so the vectors are orthogonal.

$$A \cdot B = 1 + (-1) + (-1) + 1 + 1 + (-1) + (-1) + 1 = 0$$

```
> film
   A  B  C clarity
1 -1 -1 -1       1
2  1 -1 -1       1
3 -1  1 -1       0
4  1  1 -1       0
5 -1 -1  1       1
6  1 -1  1       1
7 -1  1  1       0
8  1  1  1       0
> film.lm1 <- lm(clarity~A+B+C,data=film)
```

Note that the output (clarity) perfectly reflects factor B. We will see in a formal analysis that only the B coefficient is different from 0, and that the error in prediction is 0. This is an extreme case of detecting a factor that is important and two that are not important (*inert*).

```
> summary(film.lm1)

Call:
lm(formula = clarity ~ A + B + C, data = film)

Residuals:
         1          2          3          4
-9.529e-17  1.530e-17  5.455e-17  2.544e-17
         5          6          7          8
 8.812e-17 -8.129e-18 -4.738e-17 -3.261e-17     #Residuals are all essentially 0

Coefficients:
             Estimate Std. Error    t value Pr(>|t|)
(Intercept)  5.000e-01  2.743e-17  1.823e+16   <2e-16 ***        # 0.5
A            2.179e-33  2.743e-17  0.000e+00     1.00            # 0.0
B           -5.000e-01  2.743e-17 -1.823e+16   <2e-16 ***        #-0.5
C            2.902e-17  2.743e-17  1.058e+00     0.35            # 0.0

Residual standard error: 7.758e-17 on 4 degrees of freedom   #MSE essentially 0

Warning message:
In summary.lm(film.lm1) :
  essentially perfect fit: summary may be unreliable
```

Scaling

In section 5.4 of the text, a pilot plant investigation is presented. The goal is to obtain the best response yield from the setup (average of two replicates).

Factor	Name	Values	$-1/1$ Coding	$0/1$ Coding
Temperature (°C)	T	160	-1	0
—	—	180	1	1
Concentration (%)	C	20	-1	0
—	—	40	1	1
Catalyst	K	A	-1	0
—	—	B	1	1

-1/1 Coding as in the Book

```
> pilot
  run  T  C  K  y
1   1 -1 -1 -1 60
2   2  1 -1 -1 72
3   3 -1  1 -1 54
4   4  1  1 -1 68
5   5 -1 -1  1 52
6   6  1 -1  1 83
7   7 -1  1  1 45
8   8  1  1  1 80
```

Actual Units

```
> pilot.a
  run   T  C K  y
1   1 160 20 A 60
2   2 180 20 A 72
3   3 160 40 A 54
4   4 180 40 A 68
5   5 160 20 B 52
6   6 180 20 B 83
7   7 160 40 B 45
8   8 180 40 B 80
```

0/1 Coding

```
> pilot.b
  run T C K  y
1   1 0 0 0 60
2   2 1 0 0 72
3   3 0 1 0 54
4   4 1 1 0 68
5   5 0 0 1 52
6   6 1 0 1 83
7   7 0 1 1 45
8   8 1 1 1 80
```

```
> summary(pilot.lm)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   64.250      2.531  25.385 1.43e-05 ***
T             11.500      2.531   4.544   0.0105 *
C             -2.500      2.531  -0.988   0.3792
K              0.750      2.531   0.296   0.7817
---------------------------------------------------
> summary(pilot.lma)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -124.5000    43.8392  -2.840   0.0469 *
T              1.1500     0.2531   4.544   0.0105 *
C             -0.2500     0.2531  -0.988   0.3792
KB             1.5000     5.0621   0.296   0.7817
---------------------------------------------------
> summary(pilot.lmb)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   54.500      5.062  10.766 0.000422 ***
T             23.000      5.062   4.544 0.010469 *
C             -5.000      5.062  -0.988 0.379201
K              1.500      5.062   0.296 0.781735
```

Inferences are all the same (except for Intercept). Coefficients and standard errors depend on coding of values. In the first and third cases, all coefficients have the same SE. Effect of T is 11.5 yield per 10°change, 23 yield per 20°change, and is 1.15 per 1°change.

```
> cbind(pilot,predict(pilot.lm))
  run  T  C  K  y predict(pilot.lm)
1   1 -1 -1 -1 60              54.5
2   2  1 -1 -1 72              77.5
3   3 -1  1 -1 54              49.5
4   4  1  1 -1 68              72.5
5   5 -1 -1  1 52              56.0
6   6  1 -1  1 83              79.0
7   7 -1  1  1 45              51.0
8   8  1  1  1 80              74.0
> pilot.pred <-cbind(pilot,predict(pilot.lm))
> pilot.pred[c(2,4,6,8),] - pilot.pred[c(1,3,5,7),]
  run T C K  y predict(pilot.lm)
2   1 2 0 0 12                23
4   1 2 0 0 14                23
6   1 2 0 0 31                23
8   1 2 0 0 35                23
```

The predicted change in yield between 160 °C to 180 °C is 23 averaged across levels of C and K. If we want predictions of that change for specific values of C and K, we get 12, 14, 31, and 35. Interactions are difference of those four values from 23: 11, 9, -8, and -12.

```
> summary(lm(y~T*C*K,data=pilot))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.425e+01        NaN     NaN      NaN
T            1.150e+01        NaN     NaN      NaN
C           -2.500e+00        NaN     NaN      NaN
K            7.500e-01        NaN     NaN      NaN
T:C          7.500e-01        NaN     NaN      NaN
T:K          5.000e+00        NaN     NaN      NaN
C:K         -1.963e-15        NaN     NaN      NaN
T:C:K        2.500e-01        NaN     NaN      NaN
```

When C and K are -1 and T changes from -1 to 1, the predicted yield changes from
$64.25 - 11.5 + 2.5 - 0.75 + 0.75 + 5 - 0.25$ to
$64.25 + 11.5 + 2.5 - 0.75 - 0.75 - 5 + 0.25$, a change of
$23 - 1.5 - 10 + .5 = 12$.

```
> pilot.b
  run T C K  y
1   1 0 0 0 60
2   2 1 0 0 72
3   3 0 1 0 54
4   4 1 1 0 68
5   5 0 0 1 52
6   6 1 0 1 83
7   7 0 1 1 45
8   8 1 1 1 80
> summary(lm(y~T*C*K,data=pilot.b))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)       60        NaN     NaN      NaN
T                 12        NaN     NaN      NaN
C                 -6        NaN     NaN      NaN
K                 -8        NaN     NaN      NaN
T:C                2        NaN     NaN      NaN
T:K               19        NaN     NaN      NaN
C:K               -1        NaN     NaN      NaN
T:C:K              2        NaN     NaN      NaN
```

Change in T when $C = K = 0$ is 12. Change in T when $K = 0$, $C = 1$ is $12 + 2 = 14$. Easy in this coding.

Coding

- Coding on the original scale leads to predictions from values on the original scale.
- $-1/1$ coding has equal variances of coefficients and is centered on the middle of the design. **Most of the time, you should use this coding; the same one as in the book.**
- $0/1$ coding has equal variance of coefficients and makes interactions easily predict conditional changes: result of changing temperature when C is at the low level and K is at the default level. Terms disappear if any of the variables is at the zero level.
- Inferences about coefficients and models are the same for any of them (except for the intercept).

- All of the codings have orthogonal main effects, meaning that coefficients for the main effect of a particular variable don't change if other variables are omitted.
- The -1/1 coding has all main effects and interactions orthogonal, which is a strong reason to use it.
- With 8 data points and 8 coefficients (intercept + 3 main + 3 two-way interactions + 1 three-way interaction). All predictions are the data point and the errors are 0.
- Any of the codings can be used and the choice may depend on habit or on the application.

```
> pilot2 <- read.table("tab0503.dat",header=T)
> pilot2
   run  T  C  K  y
1    6 -1 -1 -1 59
2    2  1 -1 -1 74
3    1 -1  1 -1 50
4    5  1  1 -1 69
5    8 -1 -1  1 50
6    9  1 -1  1 81
7    3 -1  1  1 46
8    7  1  1  1 79
9   13 -1 -1 -1 61
10   4  1 -1 -1 70
11  16 -1  1 -1 58
12  10  1  1 -1 67
13  12 -1 -1  1 54
14  14  1 -1  1 85
15  11 -1  1  1 44
16  15  1  1  1 81
```

Here we have the same design, but with replication at each value of T, C, and K. Replication requires each step to be re-performed, not just the final sample analyzed twice. The replication should not immediately follow the original; all 16 must be done in a random order.,

```
> summary(lm(y~T*C*K,data=pilot2))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.425e+01  7.071e-01  90.863 2.40e-13 ***
T            1.150e+01  7.071e-01  16.263 2.06e-07 ***
C           -2.500e+00  7.071e-01  -3.536 0.007670 **
K            7.500e-01  7.071e-01   1.061 0.319813
T:C          7.500e-01  7.071e-01   1.061 0.319813
T:K          5.000e+00  7.071e-01   7.071 0.000105 ***
C:K          7.216e-16  7.071e-01   0.000 1.000000
T:C:K        2.500e-01  7.071e-01   0.354 0.732810
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.828 on 8 degrees of freedom
Multiple R-squared:  0.9763,    Adjusted R-squared:  0.9555
F-statistic: 47.05 on 7 and 8 DF,  p-value: 7.071e-06
```

Temperature is important, higher is better. Concentration is important, lower is better. The T*K interaction is important, with catalyst B being better.

```
> summary(lm(y~T*C*K,data=pilot2))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.425e+01  7.071e-01  90.863 2.40e-13 ***
T            1.150e+01  7.071e-01  16.263 2.06e-07 ***
C           -2.500e+00  7.071e-01  -3.536 0.007670 **
K            7.500e-01  7.071e-01   1.061 0.319813
T:C          7.500e-01  7.071e-01   1.061 0.319813
T:K          5.000e+00  7.071e-01   7.071 0.000105 ***
C:K          7.216e-16  7.071e-01   0.000 1.000000
T:C:K        2.500e-01  7.071e-01   0.354 0.732810

> round(coef(lm(y~T*C*K,data=pilot2)),3)
(Intercept)           T           C           K
      64.25       11.50       -2.50        0.75
        T:C         T:K         C:K       T:C:K
       0.75        5.00        0.00        0.25
```

These coefficients are exactly half the size of the ones in Table 5.4. (23, -5, 1.5, 1.5, 10, 0, 0.5) because the ones above are per unit change, and -1 to 1 is a unit change of 2. The coefficients in Table 5.4 are the predicted change from the low level to the high level.

```
> summary(lm(y~T*C*K,data=pilot2))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.425e+01  7.071e-01  90.863 2.40e-13 ***
T            1.150e+01  7.071e-01  16.263 2.06e-07 ***
C           -2.500e+00  7.071e-01  -3.536 0.007670 **
K            7.500e-01  7.071e-01   1.061 0.319813
T:C          7.500e-01  7.071e-01   1.061 0.319813
T:K          5.000e+00  7.071e-01   7.071 0.000105 ***
C:K          7.216e-16  7.071e-01   0.000 1.000000
T:C:K        2.500e-01  7.071e-01   0.354 0.732810
```

When T goes up by 1 (which is 10°), the T coefficient adds 11.5 to the predicted yield. If catalyst A is used, this increase is lowered by 5 to 6.5 and when catalyst B is used this increase is raised by 5 to 16.5. So higher temperature is better in either case, but has a higher effect on yield with catalyst B.

- This interaction effect was surprising.
- Catalysts A and B were supposedly identical from different suppliers.
- But the yield from catalyst B at 180 °C was the highest that had been seen to date:

```
> pilot2
   run  T  C  K  y
6    9  1 -1  1 81
8    7  1  1  1 79
14  14  1 -1  1 85
16  15  1  1  1 81
```

```
> pilot2c <- with(pilot2,data.frame(cbind(T*0+1,T,C,K,T*C,T*K,C*K,T*C*K)))
> names(pilot2c) <- c("I","T","C","K","TC","TK","CK","TCK")
   I  T  C  K TC TK CK TCK
1  1 -1 -1 -1  1  1  1  -1
2  1  1 -1 -1 -1 -1  1   1
3  1 -1  1 -1 -1  1 -1   1
4  1  1  1 -1  1 -1 -1  -1
5  1 -1 -1  1  1 -1 -1   1
6  1  1 -1  1 -1  1 -1  -1
7  1 -1  1  1 -1 -1  1  -1
8  1  1  1  1  1  1  1   1
9  1 -1 -1 -1  1  1  1  -1
10 1  1 -1 -1 -1 -1  1   1
11 1 -1  1 -1 -1  1 -1   1
12 1  1  1 -1  1 -1 -1  -1
13 1 -1 -1  1  1 -1 -1   1
14 1  1 -1  1 -1  1 -1  -1
15 1 -1  1  1 -1 -1  1  -1
16 1  1  1  1  1  1  1   1
```

The columns are orthogonal in the sense that the dot product of each two is zero, and all are orthogonal to the intercept vector which is all $+1$ (contrasts).

- Orthogonality means that the coefficient of each term stays the same regardless of which other terms are in the model.
- This is not generally true in regression.
- This makes interpretation of effects much easier than if they variables were not orthogonal.
- Of course, the vectors are geometrically orthogonal since the dot product is zero, meaning that they are perpendicular geometrically in dimension 16.

```
> pilot2c2
   I  T  C  K TC TK CK TCK TCTK
1  1 -1 -1 -1  1  1  1  -1    1
2  1  1 -1 -1 -1 -1  1   1    1
3  1 -1  1 -1 -1  1 -1   1   -1
4  1  1  1 -1  1 -1 -1  -1   -1
5  1 -1 -1  1  1 -1 -1   1   -1
6  1  1 -1  1 -1  1 -1  -1   -1
7  1 -1  1  1 -1 -1  1  -1    1
8  1  1  1  1  1  1  1   1    1
9  1 -1 -1 -1  1  1  1  -1    1
10 1  1 -1 -1 -1 -1  1   1    1
11 1 -1  1 -1 -1  1 -1   1   -1
12 1  1  1 -1  1 -1 -1  -1   -1
13 1 -1 -1  1  1 -1 -1   1   -1
14 1  1 -1  1 -1  1 -1  -1   -1
15 1 -1  1  1 -1 -1  1  -1    1
16 1  1  1  1  1  1  1   1    1
```

The last column is the element-wise (Hadamard) product of the TC column and the TK column. Note that this is identical to the CK column. The eight vectors form a closed group under multiplication, with $T^2 = C^2 = K^2 = I$. Any repeated factor cancels out, so, for example, $TC \times TK = TCTK = T^2CK = CK$ and $TK \times TCK = C$. Oddly enough, this relationship between particular experimental designs and closed groups under multiplication persists as we look to more advanced experimental designs such as fractional factorials.

	I	T	C	K	TC	TK	CK	TCK
1	1	-1	-1	-1	1	1	1	-1
2	1	1	-1	-1	-1	-1	1	1
3	1	-1	1	-1	-1	1	-1	1
4	1	1	1	-1	1	-1	-1	-1
5	1	-1	-1	1	1	-1	-1	1
6	1	1	-1	1	-1	1	-1	-1
7	1	-1	1	1	-1	-1	1	-1
8	1	1	1	1	1	1	1	1
9	1	-1	-1	-1	1	1	1	-1
10	1	1	-1	-1	-1	-1	1	1
11	1	-1	1	-1	-1	1	-1	1
12	1	1	1	-1	1	-1	-1	-1
13	1	-1	-1	1	1	-1	-1	1
14	1	1	-1	1	-1	1	-1	-1
15	1	-1	1	1	-1	-1	1	-1
16	1	1	1	1	1	1	1	1

The last 7 columns are all of the same type. We could have assigned T, C, and K to the last three columns instead of their current ones and all the math would be similar. What if we used 2 blocks with T, C, and K as in the table, and assigned the blocks according to the last, TCK, column?

Worsted Yarn

- Table 5.6 in the text shows the results of a study on the *durance* of strands of yarn, in which strands of 250mm and 350mm, were repeatedly stretched by 8mm or 10mm, under a load of 40gm or 50gm until failure. The response is recorded as the log of the number of cycles to failure.
- We can analyze the data in -1/1 coding, 0/1 coding and original units. If only one coding is used, mostly it should be -1/1, but the other codings have advantages as well.
- With numerical predictors, we can ask new questions, such as "What do the combinations of A, B, and C look like that have equal predicted durance?" and "How do we change A, B, and C to go maximally uphill, that is to the most increased durance?"

```
> yarn
  run  A  B  C  y
1   1 -1 -1 -1 28
2   2  1 -1 -1 36
3   3 -1  1 -1 22
4   4  1  1 -1 31
5   5 -1 -1  1 25
6   6  1 -1  1 33
7   7 -1  1  1 19
8   8  1  1  1 26
```

```
> yarn.a
  run   A  B  C  y
1   1 250  8 40 28
2   2 350  8 40 36
3   3 250 10 40 22
4   4 350 10 40 31
5   5 250  8 50 25
6   6 350  8 50 33
7   7 250 10 50 19
8   8 350 10 50 26
```

```
> yarn.b
  run A B C  y
1   1 0 0 0 28
2   2 1 0 0 36
3   3 0 1 0 22
4   4 1 1 0 31
5   5 0 0 1 25
6   6 1 0 1 33
7   7 0 1 1 19
8   8 1 1 1 26
```

We can check if interactions seem important in any of these codings.

```
> summary(lm(y~A*B*C,data=yarn))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.75e+01        NaN     NaN      NaN
A            4.00e+00        NaN     NaN      NaN
B           -3.00e+00        NaN     NaN      NaN
C           -1.75e+00        NaN     NaN      NaN
A:B         -3.16e-15        NaN     NaN      NaN
A:C         -2.50e-01        NaN     NaN      NaN
B:C         -2.50e-01        NaN     NaN      NaN
A:B:C       -2.50e-01        NaN     NaN      NaN

 round(coef(lm(y~A*B*C,data=yarn)),3)
(Intercept)           A           B           C
      27.50        4.00       -3.00       -1.75

        A:B         A:C         B:C       A:B:C
       0.00       -0.25       -0.25       -0.25
```

The main effects seem more important so we will eliminate the interactions. To see what combinations of variables lead to the same predicted durance, let's use the original units.

```
> summary(lm(y~A+B+C,data=yarn.a))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 46.25000    3.05420  15.143 0.000111 ***
A            0.08000    0.00433  18.475 5.05e-05 ***
B           -3.00000    0.21651 -13.856 0.000157 ***
C           -0.35000    0.04330  -8.083 0.001273 **
```

$$\hat{y} = 46.25 + 0.08A - 3B - 0.35C$$

When is y predicted to be 25? If A is 250 and C is 40, then $B = [46.25 - 25 + (0.08)(250) - (0.35)(40)]/3 = 9.08$ If B is 10 and C is 50 then A is 328.125. We can set two of A, B, and C and determine the value of the third that makes $y = 25$. Some are inside the original design and some not.

Steepest Ascent

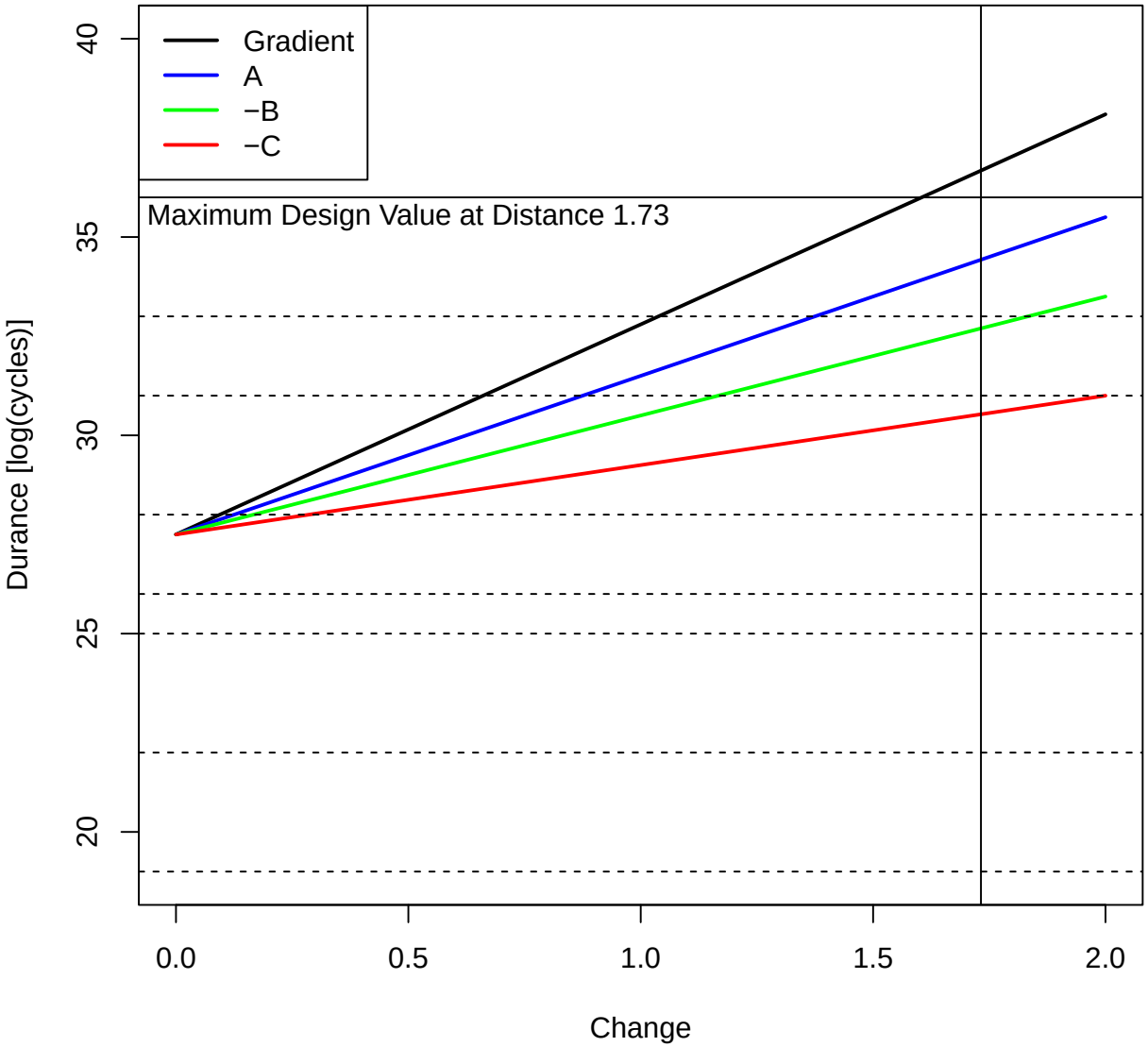
Suppose that we wanted to increase the durance by changing A, B, and C. The most uphill direction is the gradient, the vector of partial derivatives, which are just the coefficients in this case in a main effects model. This depends on the coding because "most uphill" means largest change in durance per unit length of the vector. For this calculation, the -1/1 is best because it treats each edge of the design rectangle as of equal length, and presumably, the design was set such that varying each factor had some effect, but not dominating.

```
> yarn
  run  A  B  C  y
1   1 -1 -1 -1 28
2   2  1 -1 -1 36
3   3 -1  1 -1 22
4   4  1  1 -1 31
5   5 -1 -1  1 25
6   6  1 -1  1 33
7   7 -1  1  1 19
8   8  1  1  1 26
> summary(lm(y~A+B+C,data=yarn))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  27.5000     0.2165 127.017 2.30e-08 ***
A             4.0000     0.2165  18.475 5.05e-05 ***
B            -3.0000     0.2165 -13.856 0.000157 ***
C            -1.7500     0.2165  -8.083 0.001273 **
```

Gradient vector is $G = (4, -3, -1.75)$, uphill variable values are $\lambda G = (4\lambda, -3\lambda, -1.75\lambda)$

$$
\begin{aligned}
P(a,b,c) &= 27.5 + 4a - 3b - 1.75c && \text{prediction} \\
G &= (4, -3, -1.75) && \text{gradient vector} \\
||G|| &= \sqrt{28.0625} = 5.297 && \text{length of} G \\
g &= G/5.297 \\
&= (0.7551, -0.5663, -0.3304) && \text{normalized gradient vector} \\
\lambda g &= (0.7551\lambda, -0.5663\lambda, -0.3304\lambda) && \text{uphill values} \\
P(\lambda g) &= 27.5 + (4)(0.7551)\lambda + (-3)(-0.5663)\lambda \\
&\quad + (-1.75)(-0.3394)\lambda \\
&= 27.5 + 3.0203\lambda + 1.6989\lambda + 0.5781\lambda && \text{prediction at uphill values} \\
\frac{dP(\lambda g)}{d\lambda} &= 3.0203 + 1.6989 + 0.5781 = 5.2974 && \text{slope uphill}
\end{aligned}
$$

Compare to uphill slopes in variable directions of 4, 3, and 1.75!



Colored lines are increases from the center in each coordinate and the gradient of equal lengths and direction leading to increase in durance. Design points are at distance of $\sqrt{3}$ from the center (vertical line) and achieve durances as shown in horizontal lines.

Multiple Responses

- Section 5.12 deals with a problem regarding pet rabbit food manufacture from Prat and Tort (1990), linked on the web site.
- During cooling and drying of the rabbit food pellets, loss of product in the form of fine powder was taking place. This should be reduced.
- After packaging, during manipulation and transportation, pellets eroded leaving more fine powder residue, which resulted in loss of product as well as digestive problems for the rabbits.
- Goals: quality, productivity, and cost.

Multiple Responses

- The main steps in manufacture of this pet food are as follows:
  - Decide the formula.
    - Ingredients
    - Proportions
    - Value of the PQF = glue material).
  - Mixing and Conditioning
    - Weight
    - Mix
    - Set temperature of water steam
    - Add water steam to mixture
  - Extruding and Cutting
    - Extrude mixture through metal die
    - Cut extruded material into small cylinders
  - Cooling and Drying
    - Small cylinders cooled and dried with air
    - Packaging

Wider Goals

- Educate the company in issues of product quality from the consumer point of view.
- Help the engineers and plant personnel realize how simple experimentation can improve process, quality, productivity, and cost.
- Educate plant personnel in the importance of collecting good data.
- Educate engineers in the usefulness of scientific feedback in addition to theories and hunches.

Design

This is a somewhat simplifed version from the paper, but illustrates the points.

	Variable	-1	$+1$
A	Conditioning Temperature	80% of max	Max
B	Flow	80% of max	Max
C	Compression zone	2"	2.5"

All eight combinations were run in a semi-random order (compression zone is hard to change so was set to change only three times). There was an additional variable that did not turn out to be useful and there were four additional runs that are used in the book just to estimate replication error.

Response Variables

Variable	Definition	
y_1	Powder in the product	After handling and transport
y_2	Powder in the process	After manufacturing
y_3	Yield	
y_4	Energy Consumption	

In general, to the extent possible, we would like to reduce the first two, particularly the consumer oriented y_2, increase yield and decrease energy consumption.

Powder in the Product

```
> summary(lm(y1~A*B*C,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  110.375        NaN     NaN      NaN
A             -4.125        NaN     NaN      NaN
B              2.125        NaN     NaN      NaN
C             -9.125        NaN     NaN      NaN
A:B            4.625        NaN     NaN      NaN
A:C            0.875        NaN     NaN      NaN
B:C            2.125        NaN     NaN      NaN
A:B:C         -2.875        NaN     NaN      NaN

> summary(lm(y1~A+B+C,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  110.375      2.955  37.347 3.07e-06 ***
A             -4.125      2.955  -1.396   0.2353
B              2.125      2.955   0.719   0.5119
C             -9.125      2.955  -3.088   0.0367 *
```

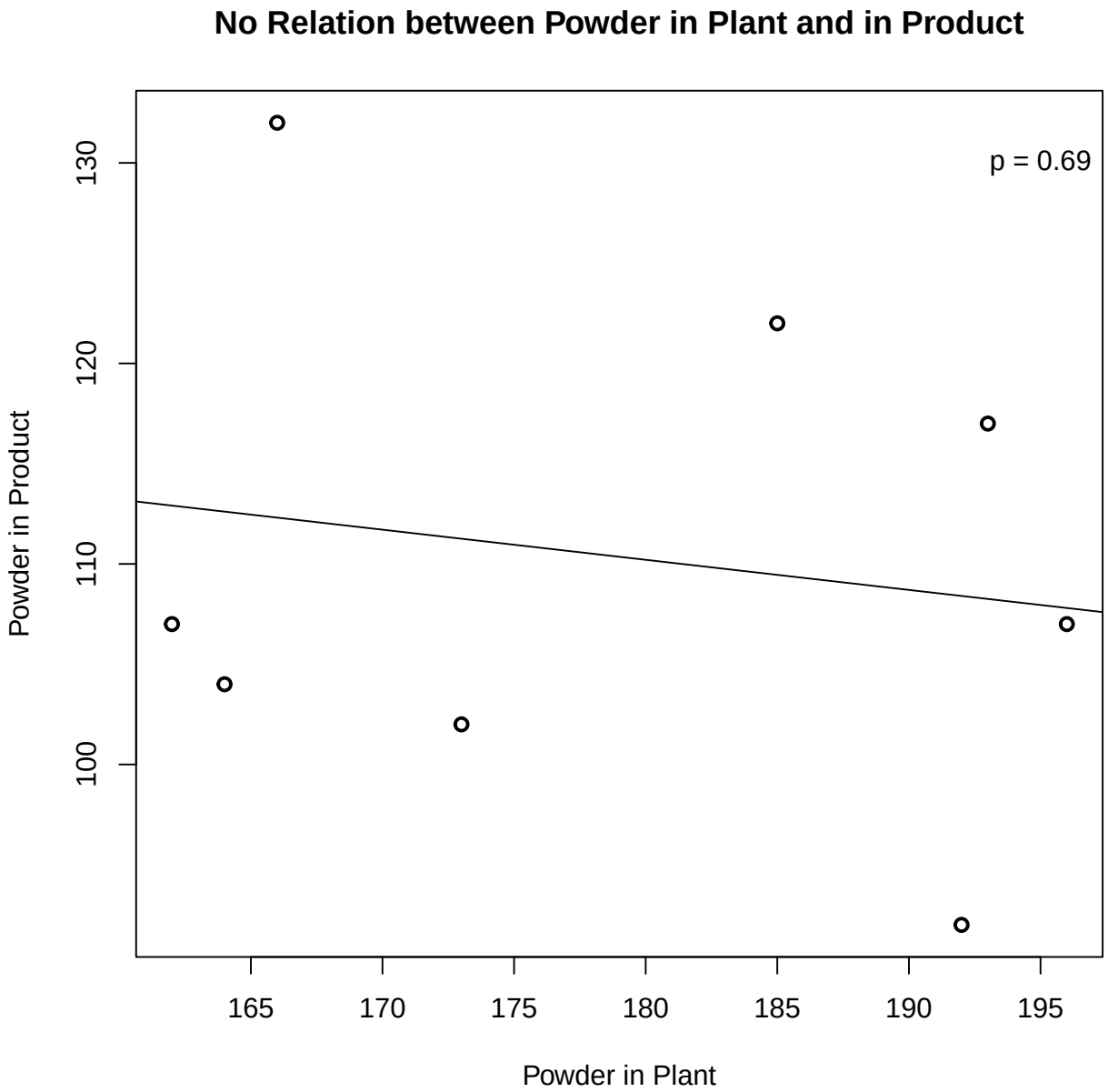
Replicate observations suggest that the standard error of the coefficients is around 2 (4 in the book where the coefficients are twice as large). The main effects model has it as around 3 (a model with the A:B interaction has it as about 2, but has p =0.12). None of the other interactions is large. Higher compression zone seems to reduce powder in product.

Powder in the Plant

```
> summary(lm(y2~A*B*C,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  178.875        NaN     NaN      NaN
A             -3.125        NaN     NaN      NaN
B              5.625        NaN     NaN      NaN
C              2.375        NaN     NaN      NaN
A:B           -6.875        NaN     NaN      NaN
A:C           -0.125        NaN     NaN      NaN
B:C           -6.875        NaN     NaN      NaN
A:B:C         -5.875        NaN     NaN      NaN

> summary(lm(y2~A+B+C,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  178.875      5.680  31.491 6.06e-06 ***
A             -3.125      5.680  -0.550    0.611
B              5.625      5.680   0.990    0.378
C              2.375      5.680   0.418    0.697
```

Replicate observations suggest that the standard error of the coefficients is around 3.7. The main effects model has it as around 5.68. None of the effects is large. Powder in the plant seems not to be related to any of the chosen production variables.


No Relation between Powder in Plant and in Product
p = 0.69
Powder in Product
100
110
120
130
165
170
175
180
185
190
195
Powder in Plant

Yield

```
> summary(lm(y3~A*B*C,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    82.00        NaN     NaN      NaN
A               1.75        NaN     NaN      NaN
B               6.50        NaN     NaN      NaN
C             -10.25        NaN     NaN      NaN
A:B            -2.75        NaN     NaN      NaN
A:C             0.50        NaN     NaN      NaN
B:C            -1.75        NaN     NaN      NaN
A:B:C          -3.00        NaN     NaN      NaN

> summary(lm(y3~A+B+C,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   82.000      2.229  36.787 3.26e-06 ***
A              1.750      2.229   0.785   0.4763
B              6.500      2.229   2.916   0.0434 *
C            -10.250      2.229  -4.598   0.0100 *
```

Replicate observations suggest that the standard error of the coefficients is around 2.5. The main effects model has it as around 2.2. Only the main effects of B = Flow and C = Compression Zone are large. Increasing the flow increases the yield, but increasing the compression zone reduces the yield.

Energy

```
> summary(lm(y4~A*B*C,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  239.875        NaN     NaN      NaN
A             -3.375        NaN     NaN      NaN
B             11.125        NaN     NaN      NaN
C             -1.125        NaN     NaN      NaN
A:B            1.875        NaN     NaN      NaN
A:C            0.625        NaN     NaN      NaN
B:C           -0.375        NaN     NaN      NaN
A:B:C          0.375        NaN     NaN      NaN

> summary(lm(y4~A+B+C,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 239.8750     0.4732 506.876 1.69e-08 ***
A            -3.3750     0.4732  -7.132 0.005675 **
B            11.1250     0.4732  23.508 0.000169 ***
C            -1.1250     0.4732  -2.377 0.097863 .
A:B           1.8750     0.4732   3.962 0.028716 *
```

Replicate observations and the reduced model suggest that the standard error of the coefficients is around 0.5. Large effects consist of A = Temperature, B = Flow Rate, and the A:B interaction. Increasing temperature reduces energy use (!), increasing flow rate increases energy use, with a significant interaction effect.

```
> summary(lm(y1~C,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  110.375      3.068  35.978 3.07e-08 ***
C             -9.125      3.068  -2.974   0.0248 *

> summary(lm(y3~B+C,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   82.000      2.142  38.285 2.29e-07 ***
B              6.500      2.142   3.035  0.02892 *
C            -10.250      2.142  -4.786  0.00495 **

> summary(lm(y4~A*B,data=rabbit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  239.875      0.696 344.663 4.25e-10 ***
A             -3.375      0.696  -4.849  0.00834 **
B             11.125      0.696  15.985 8.96e-05 ***
A:B            1.875      0.696   2.694  0.05443 .
```

Suppose that a one unit increase in yield saves 5 cents per package, and a one unit increase in energy use increases costs at 2 cents per package. Increasing flow rate (B) by one unit from the center reduces costs by about 10 cents. Increasing Compression Zone (C) by one unit reduces powder by around 9 per package at a cost of around 50 cents.

$$\begin{aligned}
\Delta\text{Powder} &= -9.125C \\
\Delta\text{Yield} &= 6.50B - 10.25C \\
\Delta\text{Energy} &= -3.375A + 11.25B + 1.875AB \\
\Delta\text{Cost} &= -7.75A + 10B - 51.25C + 3.75AB
\end{aligned}$$

Reduced powder per package has not been assigned a benefit, but the cost of such a reduction in powder can be computed.

Replicates

This study did not have true replicates. What it had is a fourth factor, which was the amount of PQF at 10 or 20. This had no apparent effect on any of the responses, which meant that as an inert factor it could generate replicates, in which two runs had the same values of A, B, and C but different values of the inert factor D. This is a major advantage of factorial type designs in screening for active factors.

Run	A	B	C	D	y_1	y_2	y_3	y_3
1	+	−	+	−	92	192	75	223
6	+	−	+	+	91	208	72	222
2	−	−	−	+	118	207	87	238
5	−	−	−	−	132	166	83	235
3	+	+	−	−	122	185	102	250
8	+	+	−	+	127	213	96	248
4	−	+	+	+	112	203	62	250
7	−	+	+	−	107	196	80	268

```
> reps
Pair   y
1 a  92
2 a  91
3 b 118
4 b 132
5 c 122
6 c 127
7 d 112
8 d 107
```

```
> summary(lm(y~Pair,data=reps))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   91.500      3.929  23.288 2.02e-05 ***
Pairb         33.500      5.557   6.029  0.00381 **
Pairc         33.000      5.557   5.939  0.00403 **
Paird         18.000      5.557   3.239  0.03169 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.557 on 4 degrees of freedom
Multiple R-squared:  0.9239,    Adjusted R-squared:  0.8669
F-statistic:  16.2 on 3 and 4 DF,  p-value: 0.01057
```

For response $y1$, we have four pairs of responses with the same values of A, B, and C. The analysis by the pairs gives us the pooled standard deviation of the four pairs of replicates. We can do the same for the other three responses.

A Four-Factor Factorial Design

The is from Section 5.13 of the text and is a 2^4 factorial exploration of factors that possibly influence the Percent Conversion in an industrial process. Here are the four factors:

Var #	Var Name	−	+
1	Catalyst charge (lb)	10	15
2	Temperature (°C)	220	240
3	Pressure (psi)	50	80
4	Concentration (%)	10	12

```
> process
   yatesOrd x1 x2 x3 x4 conversion randomOrd
1         1 -1 -1 -1 -1         70         8
2         2  1 -1 -1 -1         60         2
3         3 -1  1 -1 -1         89        10
4         4  1  1 -1 -1         81         4
5         5 -1 -1  1 -1         69        15
6         6  1 -1  1 -1         62         9
7         7 -1  1  1 -1         88         1
8         8  1  1  1 -1         81        13
9         9 -1 -1 -1  1         60        16
10       10  1 -1 -1  1         49         5
11       11 -1  1 -1  1         88        11
12       12  1  1 -1  1         82        14
13       13 -1 -1  1  1         60         3
14       14  1 -1  1  1         52        12
15       15 -1  1  1  1         86         6
16       16  1  1  1  1         79         7
```

"Yates Order" is when the first column alternates by 1 unit $-1/+1$, the second column alternates by pairs, the third column by 4's, and so on. The run order should always be randomized.

```
> processCont
   I x1 x2 x3 x4 x12 x13 x14 x23 x24 x34 x123 x124 x134 x234 x1234 conversion
1  1 -1 -1 -1 -1   1   1   1   1   1   1   -1   -1   -1   -1    1         70
2  1  1 -1 -1 -1  -1  -1  -1   1   1   1    1    1    1   -1   -1         60
3  1 -1  1 -1 -1  -1   1   1  -1  -1   1    1    1   -1    1   -1         89
4  1  1  1 -1 -1   1  -1  -1  -1  -1   1   -1   -1    1    1    1         81
5  1 -1 -1  1 -1   1  -1   1  -1   1  -1    1   -1    1    1   -1         69
6  1  1 -1  1 -1  -1   1  -1  -1   1  -1   -1    1   -1    1    1         62
7  1 -1  1  1 -1  -1  -1   1   1  -1  -1   -1    1    1   -1    1         88
8  1  1  1  1 -1   1   1  -1   1  -1  -1    1   -1   -1   -1   -1         81
9  1 -1 -1 -1  1   1   1  -1   1  -1  -1   -1    1    1    1   -1         60
10 1  1 -1 -1  1  -1  -1   1   1  -1  -1    1   -1   -1    1    1         49
11 1 -1  1 -1  1  -1   1  -1  -1   1  -1    1   -1    1   -1    1         88
12 1  1  1 -1  1   1  -1   1  -1   1  -1   -1    1   -1   -1   -1         82
13 1 -1 -1  1  1   1  -1  -1  -1  -1   1    1    1   -1   -1    1         60
14 1  1 -1  1  1  -1   1   1  -1  -1   1   -1   -1    1   -1   -1         52
15 1 -1  1  1  1  -1  -1  -1   1   1   1   -1   -1   -1    1   -1         86
16 1  1  1  1  1   1   1   1   1   1   1    1    1    1    1    1         79
```

These 16 columns form a closed group by element-wise multiplication in the same way as previously. Also, the dot product of each column times the conversion column is the effect as labeled.

```
> processCont$x124 * processCont$x234
 [1]  1 -1  1 -1 -1  1 -1  1  1 -1  1 -1 -1  1 -1  1
> processCont$x124 * processCont$x234 - processCont$x13   # (124)(234) = (13)
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
> summary(lm(conversion~x1*x2*x3*x4,data=process))
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.225e+01        NaN     NaN      NaN
x1          -4.000e+00        NaN     NaN      NaN
x2           1.200e+01        NaN     NaN      NaN
x3          -1.250e-01        NaN     NaN      NaN
x4          -2.750e+00        NaN     NaN      NaN
x1:x2        5.000e-01        NaN     NaN      NaN
x1:x3        3.750e-01        NaN     NaN      NaN
x2:x3       -6.250e-01        NaN     NaN      NaN
x1:x4       -7.138e-16        NaN     NaN      NaN
x2:x4        2.250e+00        NaN     NaN      NaN
x3:x4       -1.250e-01        NaN     NaN      NaN
x1:x2:x3    -3.750e-01        NaN     NaN      NaN
x1:x2:x4     2.500e-01        NaN     NaN      NaN
x1:x3:x4    -1.250e-01        NaN     NaN      NaN
x2:x3:x4    -3.750e-01        NaN     NaN      NaN
x1:x2:x3:x4 -1.250e-01        NaN     NaN      NaN
```

The four- and three-factor interactions seem small.

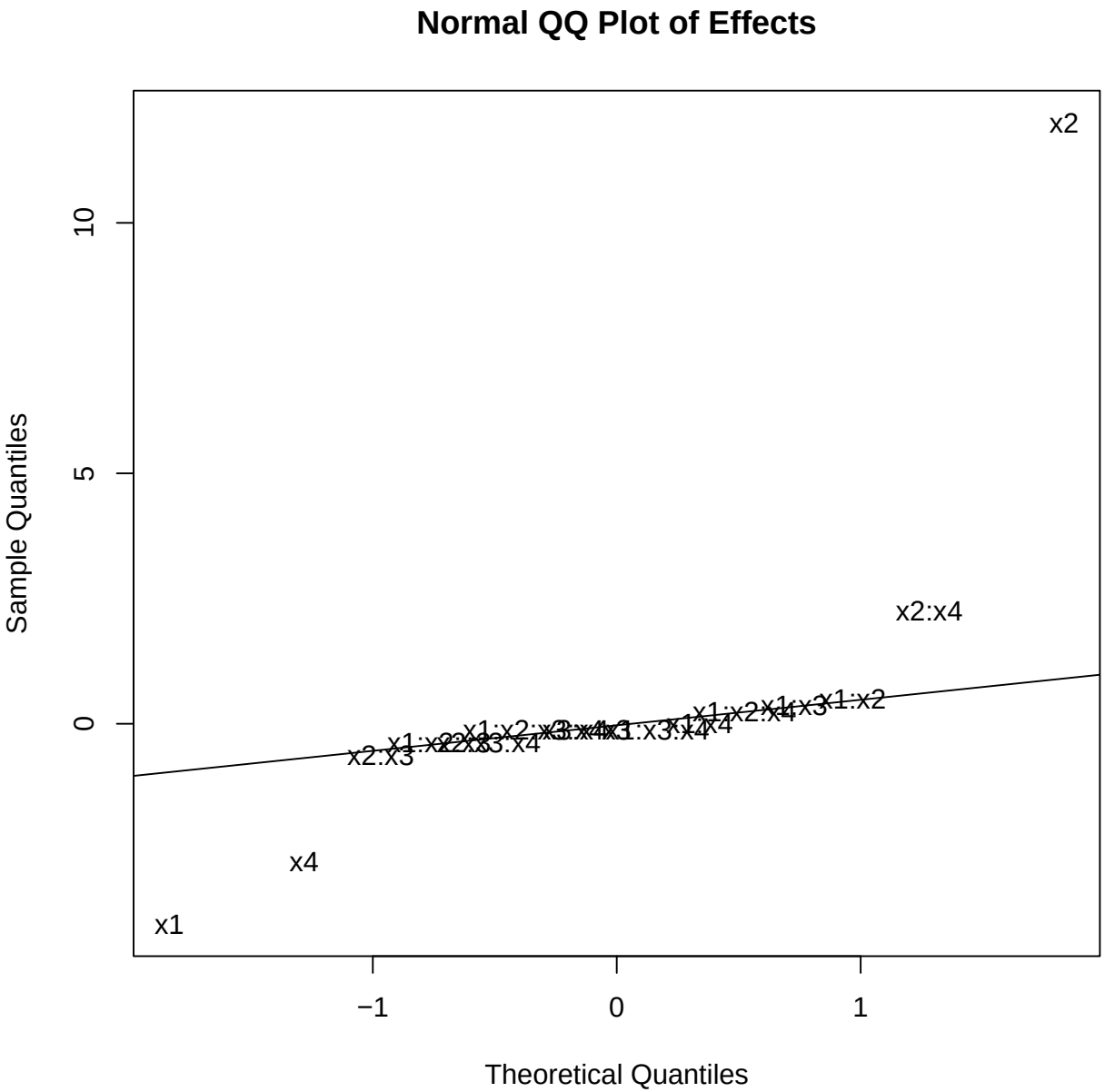
```
> summary(lm(conversion~x1*x2+x1*x3+x1*x4+x2*x3+x2*x4+x3*x4,data=process))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.225e+01  2.739e-01 263.820 1.48e-11 ***
x1          -4.000e+00  2.739e-01 -14.606 2.72e-05 ***
x2           1.200e+01  2.739e-01  43.818 1.17e-07 ***
x3          -1.250e-01  2.739e-01  -0.456 0.667219
x4          -2.750e+00  2.739e-01 -10.042 0.000168 ***
x1:x2        5.000e-01  2.739e-01   1.826 0.127464
x1:x3        3.750e-01  2.739e-01   1.369 0.229205
x1:x4       -8.257e-16  2.739e-01   0.000 1.000000
x2:x3       -6.250e-01  2.739e-01  -2.282 0.071344 .
x2:x4        2.250e+00  2.739e-01   8.216 0.000435 ***
x3:x4       -1.250e-01  2.739e-01  -0.456 0.667219

Residual standard error: 1.095 on 5 degrees of freedom
```

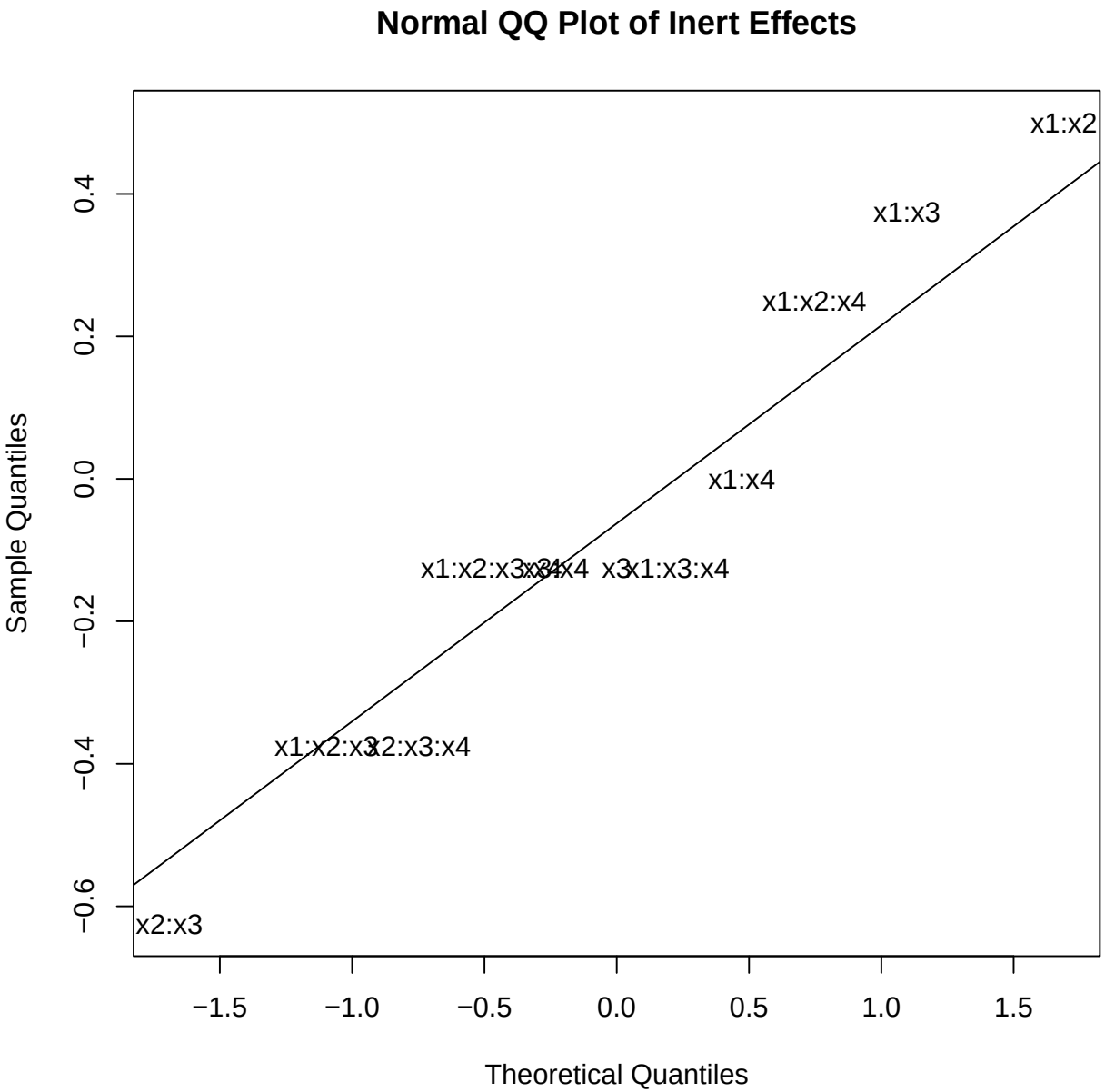
Catalyst charge ($x1$), Temperature ($x2$), Concentration ($x4$), and the Temperature by Concentration interaction ($x2 : x4$) seem important. Pressure is less so, but there is an almost-significant interaction of Temperature and Pressure. Directions leading to increased conversion are reducing Catalyst, increasing Temperature, and reducing Concentration.

Filtering Effects by Normal QQ Plots

- If there are no real effects, then the collection of the 15 non-intercept coefficients looks like a sample from a normal distribution.
- Normal, because as the sum of 16 random errors, the central limit theorem says the distribution should be roughly normal.
- If a few of the effects are real, these will show up as "outliers" while the rest will look like the normal sample. Orthogonality keeps them separate.
- We can make the 15-coefficient normal QQ plot, pick out the "outliers" and then plot the remaining ones.



This confirms the choice of active effects. The remainder look like a random sample from a normal distribution. No sign of importance of the x3 or x1:x3 terms.



The "inert effects" look sufficiently like a random sample from a normal distribution.

Why Does this Work?

- Suppose that we don't vary anything but just measure under the same conditions 16 times.
- Then roughly $y_i \sim N(\mu, \sigma^2)$.
- The intercept will be near μ and the other coefficients will be near 0 and look like a sample from a normal distribution $N(0, 16\sigma^2)$.

- Suppose now that we change factor $\mathbf{1}$ so that when $\mathbf{1} = 1$, $y = \mu + \delta + \epsilon$ and when $\mathbf{1} = -1$, $y = \mu - \delta + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$.
- All the other effects are orthogonal to $\mathbf{1}$, so their average is $0 + 8\delta - 8\delta = 0$, and their variance is $N(0, 16\sigma^2)$, as before.
- Effects that are inert behave like effects that don't change anything, so all the inert effects look like a random sample from a normal distribution.

Blocking Factorial Designs

- Suppose we have a 2^3 factorial design for a process that uses a reagent for which a batch can only accommodate four runs.
- We need a way to assign the eight runs to batch A or batch B so that there are four of each and so that the batch effect interferes as little as possible with the three assigned factors.
- We can use one of the existing $-/+$ patterns, and the one that interferes the least is the **123** interaction.
- If we let the block factor be denoted by **4**, then we have $\mathbf{4} = \mathbf{123}$.

- What we mean by equating two expressions like $\mathbf{4 = 123}$ is that, however we label the calculated result, it is the sum of the effect of the blocks, and the three-way interaction effect of the three factors.
- These are confounded, meaning that if the effect is large, we can't be sure if it is large because of a large block effect or because of a large three-factor interaction $\mathbf{123}$, or both.
- Of course, if both effects are real and of opposite signs, then they could also cancel.
- Of the 15 possible effects, the best one to confound with blocks is the $\mathbf{123}$ interaction because is is least likely to be large.

We now have 8 effects, each of which confounds two effects, and we can figure out the pattern starting with $\mathbf{4 = 123}$ or $\mathbf{I = 1234}$.

$$\mathbf{1 = 234} \qquad \mathbf{12 = 34}$$
$$\mathbf{2 = 134} \qquad \mathbf{13 = 24}$$
$$\mathbf{3 = 124} \qquad \mathbf{23 = 14}$$

so that main effects are confounded only with three-way interactions, likely to be small. Two-way interactions of factors are confounded with two-way interactions between a factor and the blocks, which are often assumed to be negligible. We can therefore interpret (usually) the seven coefficients as three main effects, three two-factor interactions and one block effect.

Blocks of Size Two

- If we either had two blocking factors, or needed blocks smaller than 4, we could make blocks of size two instead of four.
- One way to do that would be to pick two of the factor interactions to confound with blocks.
- Suppose we have $\mathbf{4} = \mathbf{123}$ and we choose another interaction to assign to a second blocking factor such as $\mathbf{5} = \mathbf{23}$.
- Divide the eight runs into four blocks, we need a third relation, which is $\mathbf{45} = (\mathbf{123})(\mathbf{23}) = \mathbf{1}$, so we have unfortunately confounded a main effect with a contrast of the difference of blocks.

A better plan is to set $\mathbf{4 = 12}$, $\mathbf{5 = 23}$ so that $\mathbf{45 = 13}$. We still have 8 effects, but each of them confounds four effects, and we can figure out the pattern starting with $\mathbf{I = 124 = 235 = 1345}$.

$$
\begin{aligned}
\mathbf{1} &= \mathbf{24} = \mathbf{1235} = \mathbf{345} \\
\mathbf{2} &= \mathbf{14} = \mathbf{35} = \mathbf{12345} \\
\mathbf{3} &= \mathbf{1234} = \mathbf{25} = \mathbf{145} \\
\mathbf{4} &= \mathbf{12} = \mathbf{2345} = \mathbf{135} \\
\mathbf{5} &= \mathbf{1245} = \mathbf{23} = \mathbf{134} \\
\mathbf{45} &= \mathbf{125} = \mathbf{234} = \mathbf{13}
\end{aligned}
$$

Now blocks are confounded with the two-way interactions, but not with any main effects and main effects are confounded also with two-way interactions. We can reasonably estimate the three main effects and the collective size of the block effects. and one block effect.

Generators and Defining Relations

Generators are letters (we use letters instead of numbers just for this discussion) like $a, b, c, \ldots$. There is a unit I such that for all letters $mI = Im = m$. Letter sequences are commutative, meaning that $pm = mp$ and also for all letters $mm = m^2 = I$. If we have three letters a, b, c then there are eight possible elements, $I, a, b, c, ab, ac, bc, abc$. If there are four letters, a, b, c, d then there would be 16 elements, unless one of the elements is defined in terms of the others, like $d = abc$, which in standardized form is $abcd = I$. This *relation* cuts the number of distinct elements in half, so there are only eight, not 16.

If there are n letters, then there are 2^n possible elements. If there are k relations, then that reduces the size of the unique elements by up to a factor of 2^k, depending on how many unique relations are generated. For example, if there are six letters, a, b, c, d, e, f, the relations $ab = I$, $bc = I$, and $cd = I$ also imply $ac = I$, $bd = I$, $ad = I$, and $abcd = I$, making 8 relations (counting $I = I$) so the original size of $2^6 = 64$ is reduced by a factor of $2^3 = 8$, leaving $64/8 = 8$. On the other hand, the three relations $ab = I$, $bc = I$, and $ac = I$ generate only $2^2 = 4$, so the full set of elements has size $64/4 = 16$. We use this algebra to figure out how to add additional factors or blocks without adding to the size of the study. (See Table 5A.1).