stat430 a4 q3

Yiming Shen 20891774

25/11/2023

```
x1 = rep(c(-1,1), times=8)
x2 = rep(c(-1,1), times=4, each=2)
x3 = rep(c(-1,1), times=2, each=4)
x4 = rep(c(-1,1), each=8)
x5 = x1 * x2 * x3
x6 = x2 * x3 * x4
x7 = x1 * x3 * x4
y = c(52,74,84,97,117,52,128,106,146,87,85,100,120,171,132,100)
datq3 = data.frame(x1,x2,x3,x4,x5,x6,x7,y)
datq3
```

```
##    x1 x2 x3 x4 x5 x6 x7   y
## 1  -1 -1 -1 -1 -1 -1 -1  52
## 2   1 -1 -1 -1  1 -1  1  74
## 3  -1  1 -1 -1  1  1 -1  84
## 4   1  1 -1 -1 -1  1  1  97
## 5  -1 -1  1 -1  1  1  1 117
## 6   1 -1  1 -1 -1  1 -1  52
## 7  -1  1  1 -1 -1 -1  1 128
## 8   1  1  1 -1  1 -1 -1 106
## 9  -1 -1 -1  1 -1  1  1 146
## 10  1 -1 -1  1  1  1 -1  87
## 11 -1  1 -1  1  1 -1  1  85
## 12  1  1 -1  1 -1 -1 -1 100
## 13 -1 -1  1  1  1 -1 -1 120
## 14  1 -1  1  1 -1 -1  1 171
## 15 -1  1  1  1 -1  1 -1 132
## 16  1  1  1  1  1  1  1 100
```

(a)

```
mod <- lm(y ~ x1 * x2 * x3 * x4, data = datq3)
summary(mod)
```

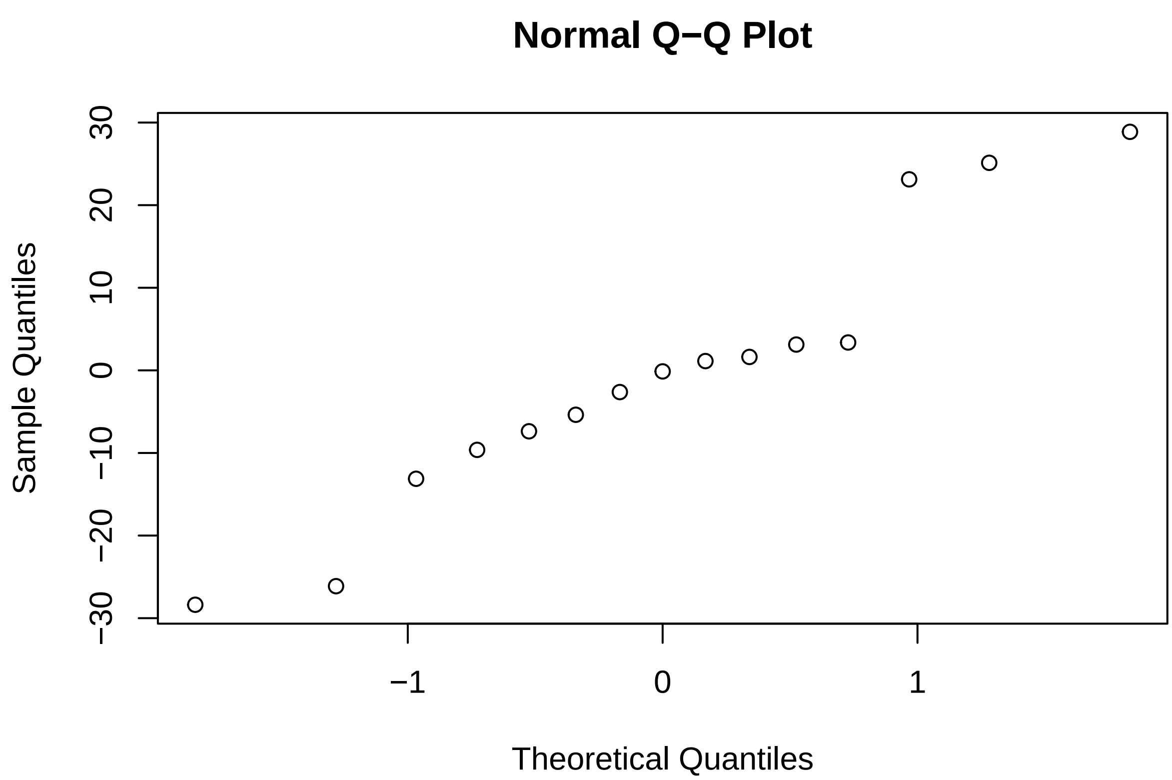
```
##
## Call:
## lm(formula = y ~ x1 * x2 * x3 * x4, data = datq3)
##
## Residuals:
## ALL 16 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 103.1875        NaN     NaN      NaN
## x1           -4.8125        NaN     NaN      NaN
## x2            0.8125        NaN     NaN      NaN
## x3           12.5625        NaN     NaN      NaN
## x4           14.4375        NaN     NaN      NaN
## x1:x2         1.5625        NaN     NaN      NaN
## x1:x3        -3.6875        NaN     NaN      NaN
## x2:x3        -0.0625        NaN     NaN      NaN
## x1:x4         1.6875        NaN     NaN      NaN
## x2:x4       -14.1875        NaN     NaN      NaN
## x3:x4         0.5625        NaN     NaN      NaN
## x1:x2:x3     -6.5625        NaN     NaN      NaN
## x1:x2:x4     -2.6875        NaN     NaN      NaN
## x1:x3:x4     11.5625        NaN     NaN      NaN
## x2:x3:x4     -1.3125        NaN     NaN      NaN
## x1:x2:x3:x4 -13.0625        NaN     NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 15 and 0 DF,  p-value: NA
```

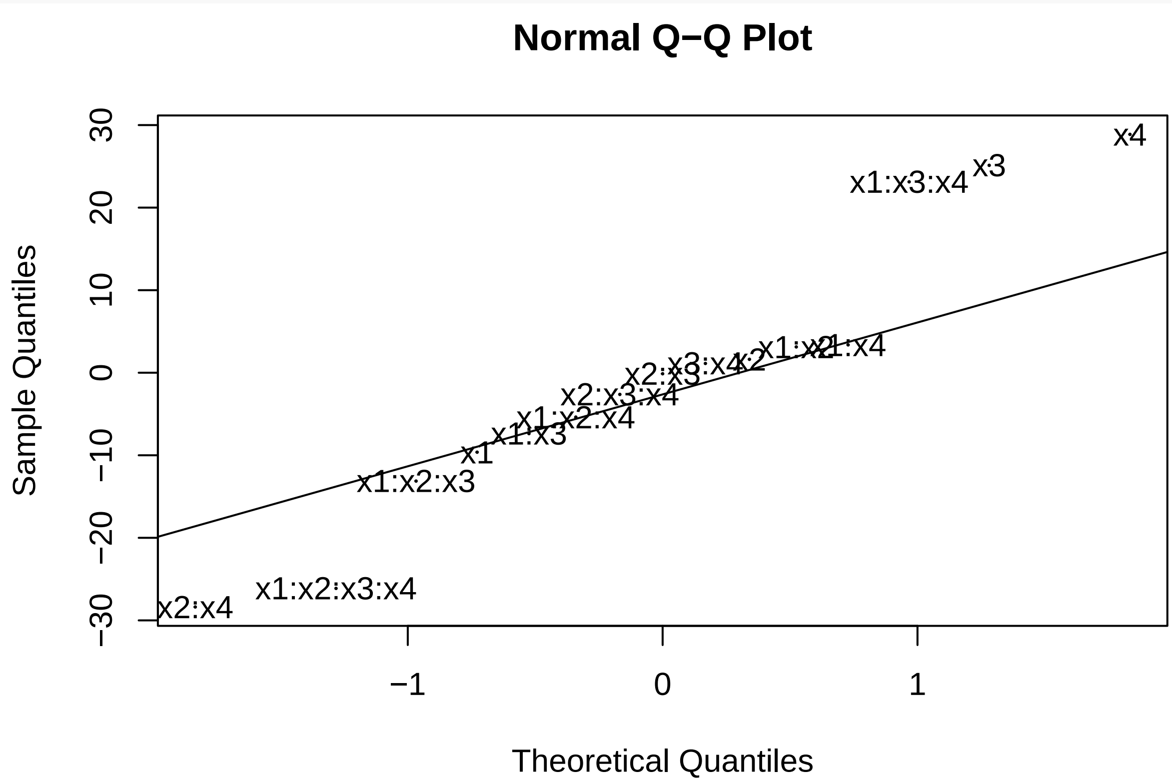
```
effects <- mod$coef * 2
effects = effects[-1]
effects
```

```
##          x1          x2          x3          x4       x1:x2       x1:x3
##      -9.625       1.625      25.125      28.875       3.125      -7.375
##       x2:x3       x1:x4       x2:x4       x3:x4    x1:x2:x3    x1:x2:x4
##      -0.125       3.375     -28.375       1.125     -13.125      -5.375
##    x1:x3:x4    x2:x3:x4 x1:x2:x3:x4
##      23.125      -2.625     -26.125
```

```
xpos <- qqnorm(effects)$x
```




```
ypos <- qqnorm(effects,cex=0.1)$y
text(x=xpos,y=ypos,labels=names(effects))
qqline(effects)
```



```r
# get the PSE
s0 <- 1.5 * median( abs(effects) )
noise <- which(abs(effects) < 2.5*s0)
pse <- 1.5 * median( abs(effects)[noise])
pse
```

```
## [1] 8.0625
```

```r
# get the merr
d = length(effects) / 3
merr <- -qt(0.025,df=d) * pse
merr
```

```
## [1] 20.72532
```

```r
# significant factors
which( abs(effects) > merr)
```

```
##          x3          x4       x2:x4    x1:x3:x4 x1:x2:x3:x4
##           3           4           9          13          15
```

Comments: Based on the Lenth's method, we found that the main effects of factors x3, x4, x7 (assumed that three-factor interactions are negligible) and the interaction between x2 & x4; x2 & x7 (x1:x2:x3:x4 driven by aliased interaction involving x7) are significant. Due to the principle of heredity, I would like to include that the main effect of x2 is significant as well.

(b)

Based on (a), we found that main effects of x3,x4,x7,x2:x4,x2:x7 (and x2) are significant, so we can build a model using only these significant effects as follows.

```r
mod2 <- lm(y ~ x3 + x4 + x7 + x2*x4 + x2*x7)
summary(mod2)
```

```
##
## Call:
## lm(formula = y ~ x3 + x4 + x7 + x2 * x4 + x2 * x7)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.312  -8.938   1.938   6.938  15.438
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 103.1875     3.2383  31.865 1.45e-10 ***
## x3           12.5625     3.2383   3.879  0.00374 **
## x4           14.4375     3.2383   4.458  0.00158 **
## x7           11.5625     3.2383   3.571  0.00602 **
## x2            0.8125     3.2383   0.251  0.80752
## x4:x2       -14.1875     3.2383  -4.381  0.00177 **
## x7:x2       -13.0625     3.2383  -4.034  0.00296 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.95 on 9 degrees of freedom
## Multiple R-squared:  0.9024, Adjusted R-squared:  0.8373
## F-statistic: 13.87 on 6 and 9 DF,  p-value: 0.0004293
```

Based on mod and mod2 above, we believe that x1,x5,x6 among the 2^(7-3) fractional factorial design can be reasonably treated as if it weren't factors at all. (We still think x2 is significant due to the significance of x2:x4 and x2:x7) We can project our 2^(7-3) fractional factorial design onto a projected 2^4 full factorial design using x2,x3,x4,x7 as our factors.

so we can design a full factorial design experiment as follows: 1. there are 4 factors, so it is a 2^4 full design. 2. there are 2^10 sample units, and it is a balanced design, so each condition will has 64 replicates.

```
# The full model can be designed in R:
# mod3 <- lm(y ~ x2 * x3 * x4 * x7, data = datq3)
# summary(mod3)
```

(c)

After factoring phase, we can use the method of steepest ascent/descent and response surface design to locate optimal settings of the factors we identified.

For the N = 1000 samples, we need ensure every condition has adequate replicates. Then we code factors and response surface experimentation can be used to reach the goal of response optimization: a second order model can be fitted and the optimum can be discovered via finding the stationary point.