Chapter 6 (8th Edition): 6.1, 6.3, 6.5, 6.6, 6.10, 6.11, 6.17, 6.22, 6.23, 6.25, 6.36, 6.37

Jeremy Ling & Emmanuel Mejia

April 24, 2018

```
# loading libraries
library(car)
library(gplots)
```

6.1

An engineer is interested in the effects of cutting speed (A), tool geometry (B), and cutting angle (C) on the life (in hours) of a machine tool. Two levels of each factor are chosen, and three replicates of a 2^3 factorial design are run. The results are as follows:

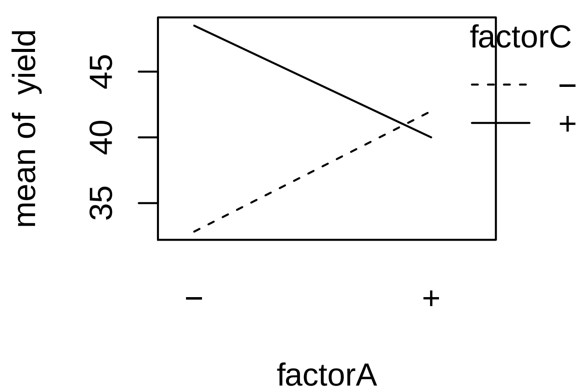
(a) Estimate the factor effects. Which effects appear to be large?

```
# creating data table
factorA = rep(c("-","+","-","+","-","+","-","+"), times = 3)
factorB = rep(c("-","-","+","+","-","-","+","+"), times = 3)
factorC = rep(c("-","-","-","-","+","+","+","+"), times = 3)
Rep = rep(c("I", "II", "III"), each = 8)
yield = c(22,32,35,55,44,40,60,39,31,43,34,47,45,37,50,41,25,29,50,46,38,36,54,47)

cutting.speed.long = data.frame(factorA, factorB, factorC, Rep, yield)

# defining coded
coded=function(x) #a function to code variable x
{
  ifelse(x=="+", 1, -1)
}

# linear regression
cutting.speed.lm=lm(yield ~ coded(factorA) * coded(factorB) * coded(factorC), cutting.speed.long)
summary(cutting.speed.lm)
```

```
##
## Call:
## lm(formula = yield ~ coded(factorA) * coded(factorB) * coded(factorC),
##     data = cutting.speed.long)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.667 -3.500 -1.167  3.167 10.333
##
## Coefficients:
##                                                Estimate Std. Error t value
## (Intercept)                                     40.8333     1.1211  36.421
## coded(factorA)                                   0.1667     1.1211   0.149
## coded(factorB)                                   5.6667     1.1211   5.054
```

```
## coded(factorC)                                  3.4167     1.1211   3.048
## coded(factorA):coded(factorB)                  -0.8333     1.1211  -0.743
## coded(factorA):coded(factorC)                  -4.4167     1.1211  -3.939
## coded(factorB):coded(factorC)                  -1.4167     1.1211  -1.264
## coded(factorA):coded(factorB):coded(factorC)   -1.0833     1.1211  -0.966
##                                               Pr(>|t|)
## (Intercept)                                    < 2e-16 ***
## coded(factorA)                                0.883680
## coded(factorB)                                0.000117 ***
## coded(factorC)                                0.007679 **
## coded(factorA):coded(factorB)                 0.468078
## coded(factorA):coded(factorC)                 0.001172 **
## coded(factorB):coded(factorC)                 0.224475
## coded(factorA):coded(factorB):coded(factorC)  0.348282
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.492 on 16 degrees of freedom
## Multiple R-squared:  0.7696, Adjusted R-squared:  0.6689
## F-statistic: 7.637 on 7 and 16 DF,  p-value: 0.0003977
```

```
# interaction plot
with(cutting.speed.long, interaction.plot(factorA, factorC, yield))
```



The effects of tool geometry and cutting angle are statistically significant. While cutting speed alone isn't statistically significant, its interaction with cutting angle is. Therefore cutting speed should remain in the model.

(b) Use the analysis of variance to confirm your conclusions for part (a).

```
# ANOVA test
cutting.speed.aov=aov(yield ~ factorA * factorB * factorC, cutting.speed.long)
summary(cutting.speed.aov)
```

```
##                         Df Sum Sq Mean Sq F value   Pr(>F)
## factorA                  1    0.7     0.7   0.022 0.883680
## factorB                  1  770.7   770.7  25.547 0.000117 ***
```

```
## factorC                    1  280.2   280.2   9.287 0.007679 **
## factorA:factorB            1   16.7    16.7   0.552 0.468078
## factorA:factorC            1  468.2   468.2  15.519 0.001172 **
## factorB:factorC            1   48.2    48.2   1.597 0.224475
## factorA:factorB:factorC  1   28.2    28.2   0.934 0.348282
## Residuals               16  482.7    30.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mse=summary(cutting.speed.aov)[[1]][8,3]
mse
```

```
## [1] 30.16667
```

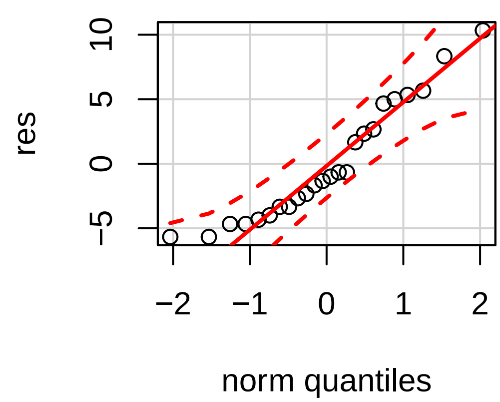
The variables that we find statistically significant also match our results from estimating factor effects in part a.

(c) Write down a regression model for predicting tool life (in hours) based on the results of this experiment.

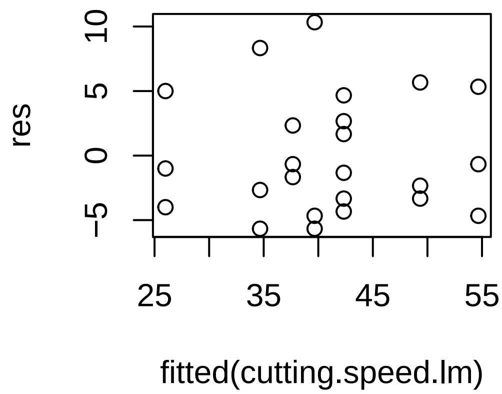
$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{1,3} X_1 X_3$$

(d) Analyze the residuals. Are there any obvious problems?

```r
res=cutting.speed.long$yield-fitted(cutting.speed.lm)
qqPlot(res)
```




```r
plot(fitted(cutting.speed.lm), res)
```



We take a look at our normality plot and can state that normality is good. We take a look at our residual plot and see no patterns, our model is good.

(e) On the basis of an analysis of main effect and interaction plots, what coded factor levels of A, B, and C would you recommend using?

Because the coefficient for factorB is positive, cutting angle should be high. In addition, the interaction plot reveals that lower cutting speed and higher life of a machine tool also produce a higher yield.

6.3

Find the standard error of the factor effects and approximate 95 percent confidence limits for the factor effects in Problem 6.1. Do the results of this analysis agree with the conclusions from the analysis of variance?

```
#checking Standard error=sqrt(mse/N)
n=3;a=b=c=2;N=a*b*c*n
alpha=0.05
sqrt(mse/N)
```

```
## [1] 1.121135
```

```
#consturct CI for regression coefficient (example, for coded(A))
se=sqrt(mse/N)
df=a*b*c*(n-1)
hat.beta1=cutting.speed.lm$coefficients[2]
CI.beta=hat.beta1+c(-1,1)*qt(alpha/2,df,lower.tail = F)*se
CI.beta
```

```
## [1] -2.210034  2.543367
```

```
2*CI.beta #CI for main effect A
```

```
## [1] -4.420068  5.086735
```

Standard Error is 1.12 and the confidence interval for the factor effects are (-4.42, 5.0867).

6.5

A router is used to cut locating notches on a printed circuit board. The vibration level at the surface of the board as it is cut is considered to be a major source of dimensional variation in the notches. Two factors are thought to influence vibration: bit size (A) and cutting speed (B). Two bit sizes (1/16 and 1/8 in.) and two speeds (40 and 90 rpm) are selected, and four boards are cut at each set of conditions shown below. The response variable is vibration measured as the resultant vector of three accelerometers (x, y, and z) on each test circuit board.

(a) Analyze the data from this experiment.

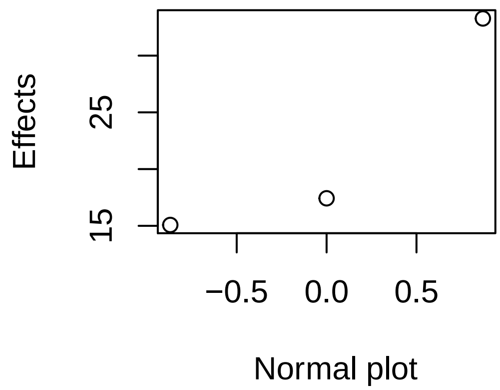
```r
# creating data table
A <- rep(c("-","+","-","+"), times = 4)
B <- rep(c("-","-","+","+"), times = 4)
Rep <- rep(c("I","II","III","IV"), each = 4)
Vibes <- c(18.2, 27.2, 15.9, 41.0, 18.9, 24.0, 14.5, 43.9, 12.9, 22.4, 15.1, 36.3, 14.4, 22.5, 14.2, 39
router.long <- data.frame(A, B, Rep, Vibes)

# defining coded
coded=function(x) #a function to code variable x
{
  ifelse(x=="+", 1, -1)
}

# linear regression
router.lm=lm(Vibes ~ coded(A) * coded(B), router.long)
summary(router.lm)
```

```
##
## Call:
## lm(formula = Vibes ~ coded(A) * coded(B), data = router.long)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.975 -1.550 -0.200  1.256  3.625
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        23.8312     0.6112  38.991 5.22e-14 ***
## coded(A)            8.3187     0.6112  13.611 1.17e-08 ***
## coded(B)            3.7687     0.6112   6.166 4.83e-05 ***
## coded(A):coded(B)   4.3562     0.6112   7.127 1.20e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.445 on 12 degrees of freedom
## Multiple R-squared:  0.9581, Adjusted R-squared:  0.9476
## F-statistic: 91.36 on 3 and 12 DF,  p-value: 1.569e-08
```

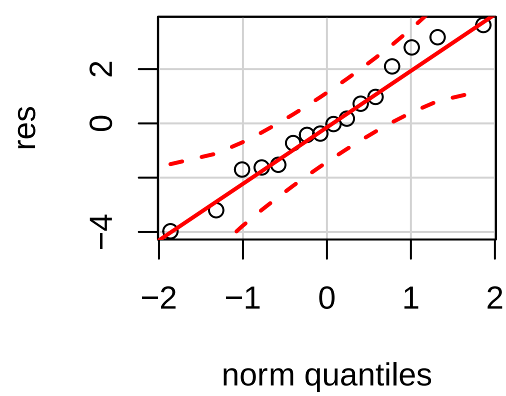
Our linear regression reveals that both treatments are statistically significant, with both variables positively correlated with vibration levels. In addition, there is evidence of interaction between the two.

(b) Construct a normal probability plot of the residuals, and plot the residuals versus the predicted vibration level. Interpret these plots.

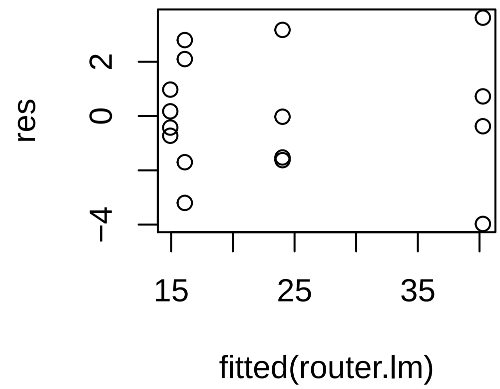
```r
router.aov = aov(Vibes ~ coded(A) * coded(B), router.long)
qqnorm(router.aov, full=T)
```




```r
res=router.long$Vibes-fitted(router.lm)
qqPlot(res)
```



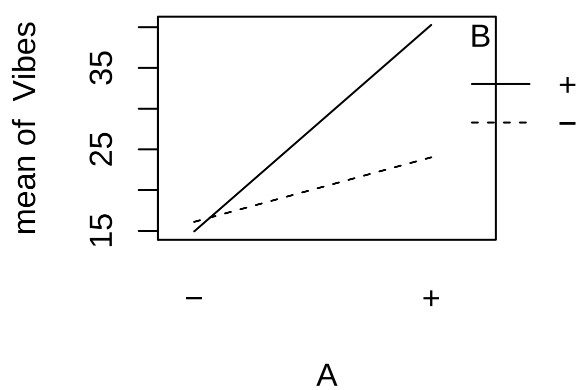

```r
plot(fitted(router.lm), res)
```



Observing our plot we see that normality is on check and nothing unusual going on in our residual plot.

(c) Draw the AB interaction plot. Interpret this plot. What levels of bit size and speed would you recommend for routine operation?

```
# interaction plot
with(router.long, interaction.plot(A, B, Vibes))
```



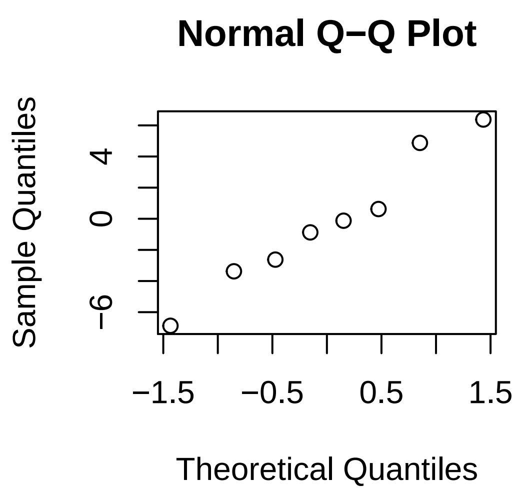
This plot reaffirms the notion that there is an interaction effect present between both variables. We'd recommend a $\frac{1}{16}$ in. bit size and 40rpm speed to minimize vibrations in this operation.

6.6

Reconsider the experiment described in Problem 6.1. Suppose that the experimenter only performed the eight trials from replicate I. In addition, he ran four center points and obtained the following response values: 36, 40, 43, 45.

(a) Estimate the factor effects. Which effects are large?

```
# defining coded
coded=function(x) #a function to code variable x
{
  ifelse(x=="+", 1,
         ifelse(x == "0", 0, -1))
}

# creating data table
factorA <- c(coded(cutting.speed.long$factorA[1:8]), 0, 0, 0, 0)
factorB <- c(coded(cutting.speed.long$factorB[1:8]), 0, 0, 0, 0)
factorC <- c(coded(cutting.speed.long$factorC[1:8]), 0, 0, 0, 0)
yield <- c(cutting.speed.long[1:8, "yield"], 36, 40, 43, 45)

cutting.speed.small <- data.frame(cbind(factorA, factorB, factorC, yield))

#linear regression
cutting.speed.lm2=lm(yield ~ factorA * factorB * factorC + I(factorA^2) + I(factorB^2) + I(factorC^2), 
#prob plot
coef=cutting.speed.lm2$coefficients[-1]
variables=names(coef)
plot=qqnorm(coef)
variables[identify(plot)]
```




```
## character(0)
```

After reviewing the probability plot we see that Factor B, Factor C, and FactorA * FactorC are the largest factors that stand out.

(b) Perform an analysis of variance, including a check for pure quadratic curvature. What are your conclusions?

```
# center vs. factorial averages
yc_bar <- mean(cutting.speed.small$yield[9:12])
test <- mean(cutting.speed.small$yield[1:8])

# ANOVA test
```

```r
cutting.speed.aov2 <- aov(yield ~ factorA * factorB * factorC + I(factorA^2) + I(factorB^2) + I(factorC
summary(cutting.speed.aov2)
```

```
##                         Df Sum Sq Mean Sq F value Pr(>F)
## factorA                  1    3.1     3.1   0.204 0.6823
## factorB                  1  325.1   325.1  21.204 0.0193 *
## factorC                  1  190.1   190.1  12.399 0.0389 *
## I(factorA^2)             1    0.0     0.0   0.003 0.9617
## factorA:factorB          1    6.1     6.1   0.399 0.5722
## factorA:factorC          1  378.1   378.1  24.660 0.0157 *
## factorB:factorC          1   55.1    55.1   3.595 0.1542
## factorA:factorB:factorC  1   91.1    91.1   5.943 0.0927 .
## Residuals                3   46.0    15.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA test reveals that there is no statistically significant reason to suspect that there is any quadratic curvature in our model.

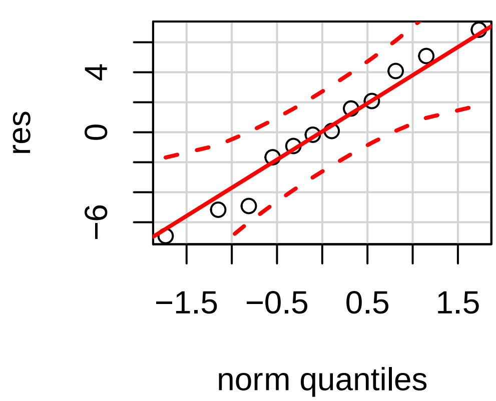
(c) Write down an appropriate model for predicting tool life, based on the results of this experiment. Does this model differ in any substantial way from the model in Problem 6.1, part (c)?

Because we know that quadratic curvature should not be introduced to our model, our appropriate model is the same as in Problem 6.1.

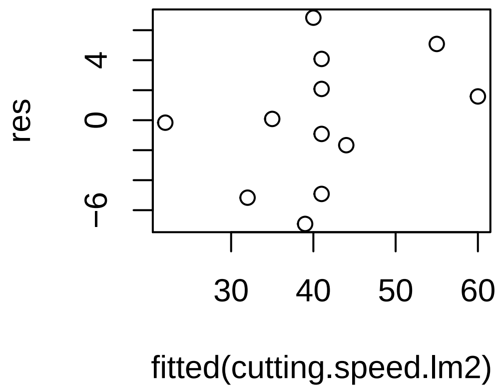
$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{1,3} X_1 X_3$$

(d) Analyze the residuals.

```r
cutting.speed.lm3 = lm(yield ~ factorA + factorB + factorC + factorA*factorC, cutting.speed.small)
res=cutting.speed.small$yield-fitted(cutting.speed.lm3)
qqPlot(res)
```




```r
plot(fitted(cutting.speed.lm2), res)
```



There is no unusual activity going on in our residuals, normality seems good as well. This is a good model.

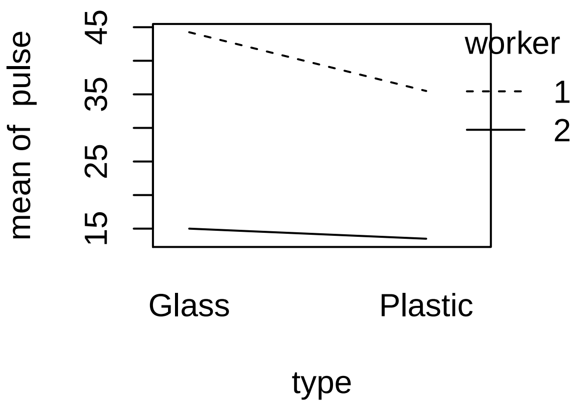
(e) What conclusions would you draw about the appropriate operating conditions for this process?

Exactly like in Problem 6.1, a high cutting angle, low cutting speed, and high life of a machine tool will produce a higher yield.

6.10

In Problem 6.9, the engineer was also interested in potential fatigue differences resulting from the two types of bottles. As a measure of the amount of effort required, he measured the elevation of the heart rate (pulse) induced by the task. The results follow. Analyze the data and draw conclusions. Analyze the residuals and comment on the model's adequacy.

```r
# creating data table
type <- rep(c("Glass", "Plastic"), each = 4)
worker <- rep(c("1", "2"), each = 8)
pulse <- c(39, 45, 58, 35, 44, 35, 42, 21, 20, 13, 16, 11, 13, 10, 16, 15)
bottle.long <- data.frame(type, worker, pulse)

# interaction plot
with(bottle.long, interaction.plot(type, worker, pulse))
```




```
# defining coded
coded=function(x) #a function to code variable x
{
  ifelse(x=="Glass" | x=="1", 1, -1)
}

# linear regression
bottle.lm=lm(pulse ~ coded(type) * coded(worker), bottle.long)
summary(bottle.lm)
```

```
##
## Call:
## lm(formula = pulse ~ coded(type) * coded(worker), data = bottle.long)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.500  -3.625   0.125   3.125  13.750
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 27.062      1.902  14.227 7.11e-09 ***
## coded(type)                  2.563      1.902   1.347    0.203
## coded(worker)               12.812      1.902   6.736 2.09e-05 ***
## coded(type):coded(worker)    1.812      1.902   0.953    0.359
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.609 on 12 degrees of freedom
## Multiple R-squared:  0.8003, Adjusted R-squared:  0.7504
## F-statistic: 16.03 on 3 and 12 DF,  p-value: 0.0001693
```

```
# ANOVA test
bottle.aov=aov(pulse ~ type * worker, bottle.long)
summary(bottle.aov)
```

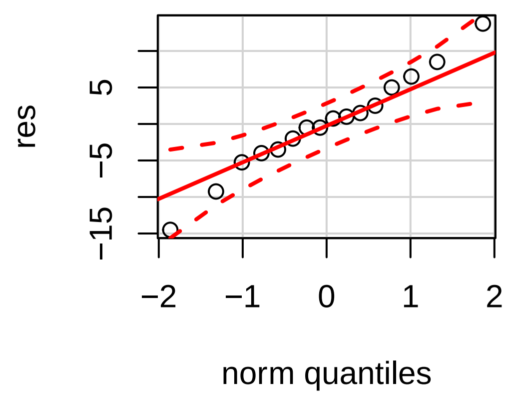
```
##             Df Sum Sq Mean Sq F value   Pr(>F)
```

```
## type          1  105.1   105.1   1.815    0.203
## worker        1 2626.6  2626.6  45.367 2.09e-05 ***
## type:worker   1   52.6    52.6   0.908    0.359
## Residuals    12  694.7    57.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

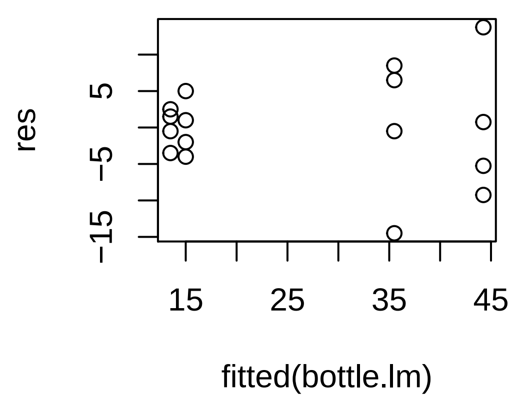
```
mse=summary(bottle.aov)[[1]][4,3]
mse
```

```
## [1] 57.89583
```

```
# checking model adequacy
res=bottle.long$pulse-fitted(bottle.lm)
qqPlot(res)
```




```
plot(fitted(bottle.lm), res)
```



When estimating factor effects, we don't suspect any interaction effects between bottle type and worker.

Both our linear model and ANOVA test suggest that worker is statistically significant in predicting heart rate when performing the task, and reaffirm the conclusion we made in our interaction plot. When checking model adequacy, our qqplot reveals that our data is normally distributed, while the variances of residuals in our residual plot don't seem homogenous. As a result, the model may not be adequate.

6.11

Calculate approximate 95 percent confidence limits for the factor effects in Problem 6.10. Do the results of this analysis agree with the analysis of variance performed in Problem 6.10?

```
#checking Standard error=sqrt(mse/N)
n=4;a=b=2;N=a*b*n
alpha=0.05
sqrt(mse/N)
```

```
## [1] 1.902233
```

```
#consturct CI for regression coefficient (example, for coded(A))
se=sqrt(mse/N)
df=a*b*(n-1)

hat.beta1=bottle.lm$coefficients[2]
CI.beta=hat.beta1+c(-1,1)*qt(alpha/2,df,lower.tail = F)*se
CI.beta
```

```
## [1] -1.582109  6.707109
```

```
2*CI.beta #CI for main effect A
```

```
## [1] -3.164218 13.414218
```

```
hat.beta1=bottle.lm$coefficients[3]
CI.beta=hat.beta1+c(-1,1)*qt(alpha/2,df,lower.tail = F)*se
CI.beta
```

```
## [1]  8.667891 16.957109
```

```
2*CI.beta #CI for main effect B
```

```
## [1] 17.33578 33.91422
```

```
hat.beta1=bottle.lm$coefficients[4]
CI.beta=hat.beta1+c(-1,1)*qt(alpha/2,df,lower.tail = F)*se
CI.beta
```

```
## [1] -2.332109  5.957109
```

```
2*CI.beta #CI for main effect AB
```

```
## [1] -4.664218 11.914218
```

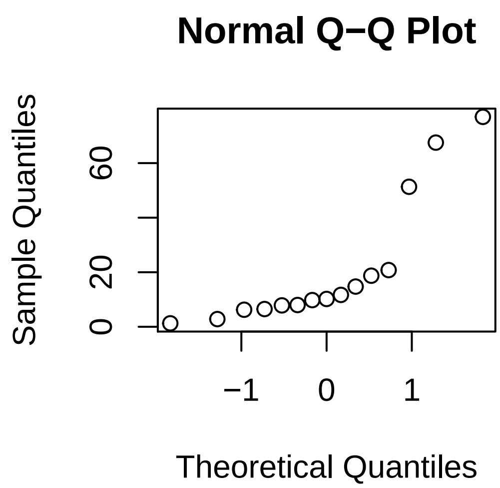
Confidence interval for main effect A is (-3.16, 13.4), for main effect B is (17.33, 33.91), and for interaction effect AB is (-4.66, 11.9). The only confidence interval the doesn't contain 0 is the one associated with 'worker'. The results agree with those receive from the ANOVA test we ran earlier.

6.17

An experimenter has run a single replicate of a 2^4 design. The following effect estimates have been calculated:

(a) Construct a normal probability plot of these effects.

```
letter = c("A","B","C","D","AB","AC","AD","BC","BD","CD","ABC","ABD","ACD","BCD","ABCD")
number = c(76.95,-67.52,-7.84,-18.73,-51.32,11.69,9.78,20.79,14.74,1.27,-2.82,-6.50,10.20,-7.98,-6.25)
experiment = data.frame(letter, number)

experiment.aov = aov(number ~ coded(letter), experiment)


plot = qqnorm(abs(number))
```

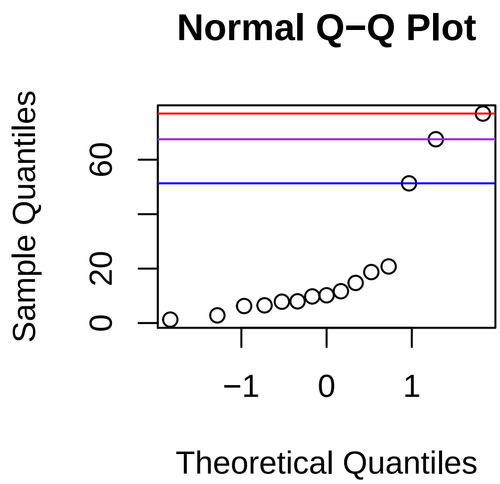



```
## click at the "outlier" points and then click "Finish" button
lister = sort(abs(number))
lister
```

```
##  [1]  1.27  2.82  6.25  6.50  7.84  7.98  9.78 10.20 11.69 14.74 18.73
## [12] 20.79 51.32 67.52 76.95
```

The largest values are 51.32 as interaction effect factor A * factor B, 67.52 as factor B, and 76.95 as factor A.

```
qqnorm(lister)
abline(h = 76.95, col = "red")
abline(h = 67.52, col = "purple")
abline(h = 51.32, col = "blue")
```




```
#factor A, B, and AB
```

(b) Identify a tentative model, based on the plot of the effects in part (a).

$$\hat{y} = \beta_0 + 75.95x_a + 67.52x_b + 51.32x_{ab}$$

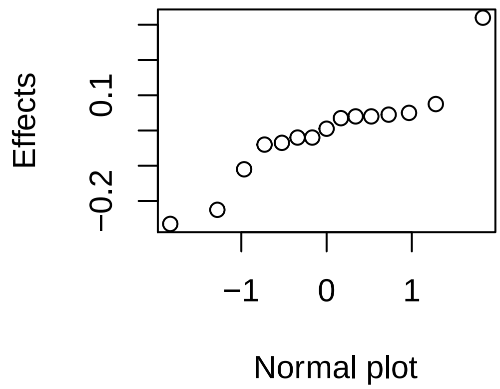
6.22

Semiconductor manufacturing processes have long and complex assembly flows, so matrix marks and automated 2d-matrix readers are used at several process steps throughout factories. Unreadable matrix marks negatively affect factory run rates because manual entry of part data is required before manufacturing can resume. A 2^4 factorial experiment was conducted to develop a 2d-matrix laser mark on a metal cover that protects a substrate-mounted die. The design factors are A = laser power (9 and 13 W), B = laser pulse frequency (4000 and 12,000 Hz), C = matrix cell size (0.07 and 0.12 in.), and D = writing speed (10 and 20 in./sec), and the response variable is the unused error correction (UEC). This is a measure of the unused portion of the redundant information embedded in the 2d-matrix. A UEC of 0 represents the lowest reading that still results in a decodable matrix, while a value of 1 is the highest reading. A DMX Verifier was used to measure UEC. The data from this experiment are shown in Table P6.5.

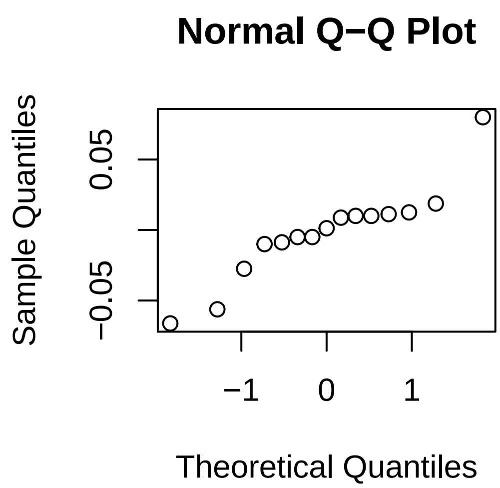
```
Standard.Order = c(8,10,12,9,7,15,2,6,16,13,5,14,1,3,4,11)
Run.Order = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)
A = Laser.Power = c(1,1,1,-1,-1,-1,1,1,1,-1,-1,1,-1,-1,1,-1)
B = Pulse.Freq = c(1,-1,1,-1,1,1,-1,-1,1,-1,-1,-1,-1,1,1,1)
C = Cell.Size = c(1,-1,-1,-1,1,1,-1,1,1,1,1,1,-1,-1,-1,-1)
D = Writing.Speed = c(-1,1,1,1,-1,1,-1,-1,1,1,-1,1,-1,-1,-1,1)
UEC = c(0.8,0.81,0.79,0.6,0.65,0.55,0.98,0.67,0.69,0.56,0.63,0.65,0.75,0.72,0.98,0.63)
error = data.frame(Standard.Order,Run.Order,A,B,C,D,UEC)
```

(a) Analyze the data from this experiment. Which factors significantly affect UEC?

```
error.lm = lm(UEC ~A*B*C*D, error)
qqnorm(aov(UEC ~ A * B * C * D, error), label=T, full=T)
```




```r
coef=error.lm$coefficients[-1]
variables=names(coef)
plot=qqnorm(coef)
variables[identify(plot)]
```




```
## character(0)
```

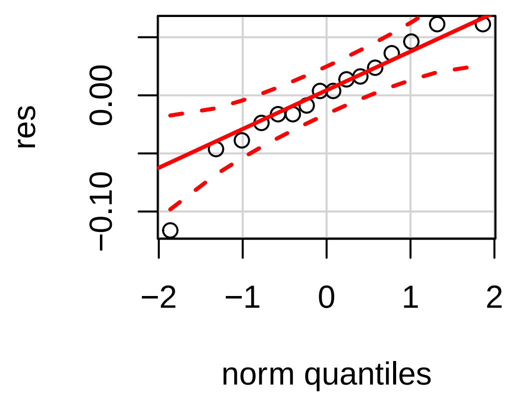
```r
##new model
error.aov = aov(UEC ~ A+C+D, error)
summary(error.aov)
```

```
##             Df  Sum Sq Mean Sq F value   Pr(>F)    
## A            1 0.10240 0.10240   40.52 3.58e-05 ***
## C            1 0.07022 0.07022   27.79 0.000197 ***
## D            1 0.05063 0.05063   20.03 0.000758 ***
## Residuals   12 0.03033 0.00253                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

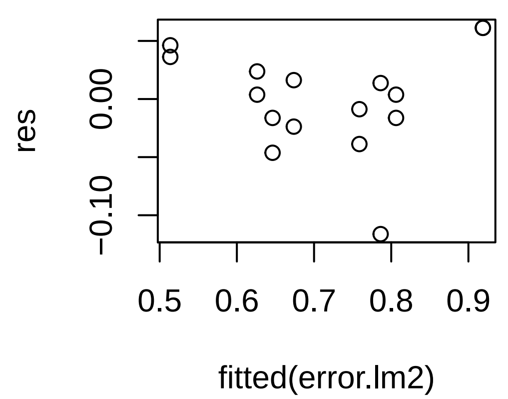
With the help of the normal probability plot we are able to identify that "A", "C", "D" are important. After choosing my factors, I plug them into a ANOVA and discover that all factors are significantly different.

(b) Analyze the residuals from this experiment. Are there any indications of model inadequacy?

```
error.lm2 = lm(UEC~ A+C+D, error)
res=error$UEC-fitted(error.lm2)
library(car)
qqPlot(res)
```




```
plot(fitted(error.lm2), res)
```



After studying the normality and residuals we can state that normality is good and residuals are patternless and random. We may conclude that model is good.

6.23

Reconsider the experiment described in Problem 6.20. Suppose that four center points are available and that the UEC response at these four runs is 0.98, 0.95, 0.93, and 0.96, respectively. Reanalyze the experiment incorporating a test for curvature into the analysis. What conclusions can you draw? What recommendations would you make to the experimenters?

```r
# creating data table
A = Laser.Power = c(1,1,1,-1,-1,-1,1,1,1,-1,-1,1,-1,-1,1,-1,0,0,0,0)
B = Pulse.Freq = c(1,-1,1,-1,1,1,-1,-1,1,-1,-1,-1,-1,1,1,1,0,0,0,0)
C = Cell.Size = c(1,-1,-1,-1,1,1,-1,1,1,1,1,1,-1,-1,-1,-1,0,0,0,0)
D = Writing.Speed = c(-1,1,1,1,-1,1,-1,-1,1,1,-1,1,-1,-1,-1,1,0,0,0,0)
UEC = c(0.8,0.81,0.79,0.6,0.65,0.55,0.98,0.67,0.69,0.56,0.63,0.65,0.75,0.72,0.98,0.63,0.98,0.95,0.93,0.9
error2 = data.frame(A,B,C,D,UEC)

#linear regression
error.lm21 <- lm(UEC ~ A * B * C * D + I(A^2) + I(B^2) + I(C^2) + I(D^2), error2)
summary(error.lm21)
```

```
##
## Call:
## lm(formula = UEC ~ A * B * C * D + I(A^2) + I(B^2) + I(C^2) +
##     I(D^2), data = error2)
##
## Residuals:
##          1          2          3          4          5          6
##  1.518e-18 -1.024e-18  1.520e-18 -1.039e-19 -2.227e-19  6.996e-19
##          7          8          9         10         11         12
##  1.821e-18 -1.493e-18 -1.747e-18 -1.020e-19  1.567e-19  1.472e-18
##         13         14         15         16         17         18
## -3.361e-19  3.145e-19 -1.456e-18 -3.714e-19  2.500e-02 -5.000e-03
##         19         20
## -2.500e-02  5.000e-03
##
## Coefficients: (3 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.955000   0.010408  91.753 2.85e-06 ***
## A            0.080000   0.005204  15.372 0.000598 ***
## B            0.010000   0.005204   1.922 0.150410
## C           -0.066250   0.005204 -12.730 0.001046 **
## D           -0.056250   0.005204 -10.809 0.001694 **
## I(A^2)      -0.238750   0.011637 -20.517 0.000253 ***
## I(B^2)             NA         NA      NA       NA
## I(C^2)             NA         NA      NA       NA
## I(D^2)             NA         NA      NA       NA
## A:B          0.008750   0.005204   1.681 0.191287
## A:C         -0.027500   0.005204  -5.284 0.013219 *
## B:C          0.012500   0.005204   2.402 0.095709 .
## A:D         -0.005000   0.005204  -0.961 0.407544
## B:D         -0.005000   0.005204  -0.961 0.407544
## C:D          0.018750   0.005204   3.603 0.036687 *
## A:B:C        0.011250   0.005204   2.162 0.119381
## A:B:D       -0.008750   0.005204  -1.681 0.191287
## A:C:D        0.010000   0.005204   1.922 0.150410
## B:C:D       -0.010000   0.005204  -1.922 0.150410
```

```
## A:B:C:D      0.001250   0.005204   0.240 0.825659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02082 on 3 degrees of freedom
## Multiple R-squared:  0.997,  Adjusted R-squared:  0.9812
## F-statistic: 62.88 on 16 and 3 DF,  p-value: 0.00285
```

```r
# ANOVA test
error.aov21 <- aov(UEC ~ A * B * C * D + I(A^2) + I(B^2) + I(C^2) + I(D^2), error2)
summary(error.aov21)
```

```
##              Df  Sum Sq Mean Sq F value   Pr(>F)
## A             1 0.10240 0.10240 236.308 0.000598 ***
## B             1 0.00160 0.00160   3.692 0.150410
## C             1 0.07023 0.07023 162.058 0.001046 **
## D             1 0.05062 0.05062 116.827 0.001694 **
## I(A^2)        1 0.18240 0.18240 420.935 0.000253 ***
## A:B           1 0.00123 0.00123   2.827 0.191287
## A:C           1 0.01210 0.01210  27.923 0.013219 *
## B:C           1 0.00250 0.00250   5.769 0.095709 .
## A:D           1 0.00040 0.00040   0.923 0.407544
## B:D           1 0.00040 0.00040   0.923 0.407544
## C:D           1 0.00563 0.00563  12.981 0.036687 *
## A:B:C         1 0.00202 0.00202   4.673 0.119381
## A:B:D         1 0.00122 0.00122   2.827 0.191287
## A:C:D         1 0.00160 0.00160   3.692 0.150410
## B:C:D         1 0.00160 0.00160   3.692 0.150410
## A:B:C:D       1 0.00003 0.00003   0.058 0.825659
## Residuals     3 0.00130 0.00043
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# rebuild model
error.lm22 <- lm(UEC ~ A*C + C*D + I(A^2) + I(B^2) + I(C^2) + I(D^2), error2)
summary(error.lm22)
```

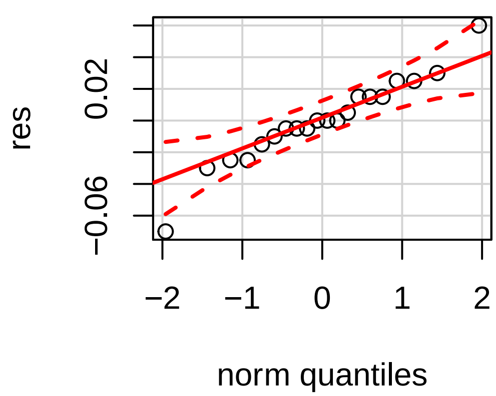
```
##
## Call:
## lm(formula = UEC ~ A * C + C * D + I(A^2) + I(B^2) + I(C^2) +
##     I(D^2), data = error2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.07000 -0.01125  0.00000  0.01500  0.06000
##
## Coefficients: (3 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.955000   0.016350  58.411  < 2e-16 ***
## A            0.080000   0.008175   9.786 2.31e-07 ***
## C           -0.066250   0.008175  -8.104 1.94e-06 ***
## D           -0.056250   0.008175  -6.881 1.12e-05 ***
## I(A^2)      -0.238750   0.018279 -13.061 7.52e-09 ***
## I(B^2)             NA         NA      NA       NA
## I(C^2)             NA         NA      NA       NA
## I(D^2)             NA         NA      NA       NA
```

```
## A:C         -0.027500   0.008175  -3.364  0.00508 **
## C:D          0.018750   0.008175   2.294  0.03912 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0327 on 13 degrees of freedom
## Multiple R-squared:  0.9682, Adjusted R-squared:  0.9535
## F-statistic: 65.99 on 6 and 13 DF,  p-value: 5.544e-09
```

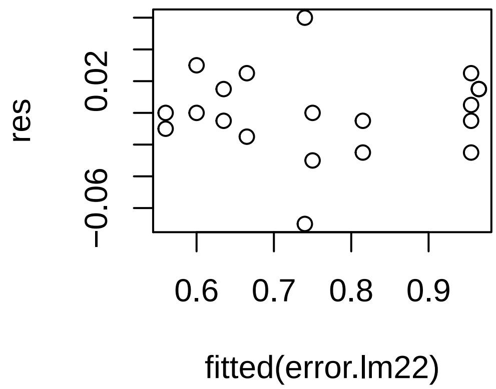
```
error.aov22 <- aov(UEC ~A*C + C*D + I(A^2) + I(B^2) + I(C^2) + I(D^2), error2)
summary(error.aov22)
```

```
##             Df  Sum Sq Mean Sq F value   Pr(>F)
## A            1 0.10240 0.10240  95.770 2.31e-07 ***
## C            1 0.07023 0.07023  65.678 1.94e-06 ***
## D            1 0.05062 0.05062  47.347 1.12e-05 ***
## I(A^2)       1 0.18241 0.18241 170.595 7.52e-09 ***
## A:C          1 0.01210 0.01210  11.317  0.00508 **
## C:D          1 0.00563 0.00563   5.261  0.03912 *
## Residuals   13 0.01390 0.00107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# checking model adequacy
res=error2$UEC-fitted(error.lm22)
qqPlot(res)
```




```
plot(fitted(error.lm22), res)
```



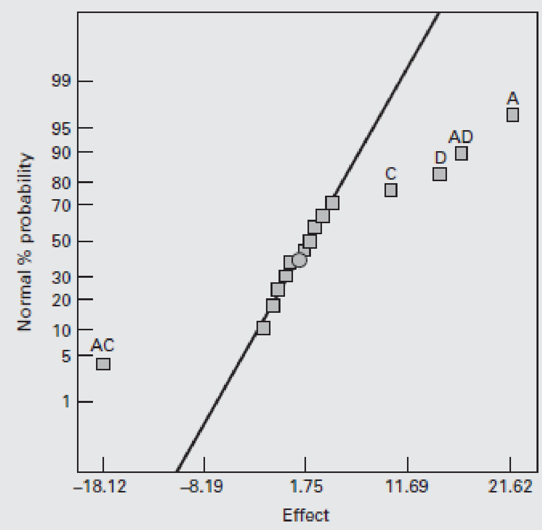
After testing for where or not our model should account for curvature, we do indeed find that squared terms are statistically significant in helping predict UEC, and should be included in the model. We'd recommend excluding laser pulse frequency from the model. Also the residuals and normality are good. This model is good.

6.25

Consider the single replicate of the 2^4 design in Example 6.2. Suppose that we had arbitrarily decided to analyze the data assuming that all three- and four-factor interactions were negligible. Conduct this analysis and compare your results with those obtained in the example. Do you think that it is a good idea to arbitrarily assume interactions to be negligible even if they are relatively high-order ones?

```
Run.Number = c(1:16)
Run.Label = c("(1)","a,","b","ab","C","ac","bc","abc","d","ad","bd","abd","cd","acd","bcd","abcd")
# creating data table
A <- rep(c("-", "+"), times = 8)
B <- rep(c("-", "+"), each = 2, times = 4)
C <- rep(c("-", "+"), each = 4, times = 2)
D <- rep(c("-", "+"), each = 8)
Filtration.Rate = c(45,71,48,65,68,60,80,65,43,100,45,104,75,86,70,96)
chem.long <- data.frame(Run.Number,A, B, C, D,Filtration.Rate)

# defining coded
coded=function(x) #a function to code variable x
{
  ifelse(x=="+", 1, -1)
}

chem.lm = lm(Filtration.Rate ~ coded(A)*coded(B)+coded(A)*coded(C)+coded(A)*coded(D)+coded(B)*coded(C)+c
summary(chem.lm)
```

```
##
## Call:
## lm(formula = Filtration.Rate ~ coded(A) * coded(B) + coded(A) *
```



■ **FIGURE 6.11** **Normal probability plot of the effects for the 2^4 factorial in Example 6.2**

Figure 1: Probability plot for Example 6.2

```
##     coded(C) + coded(A) * coded(D) + coded(B) * coded(C) + coded(B) *
##     coded(D) + coded(C) * coded(D), data = chem.long)
##
## Residuals:
##       1       2       3       4       5       6       7       8       9
## -0.1875  2.8125  1.8125 -4.4375 -3.9375  1.3125  2.3125  0.3125 -1.6875
##      10      11      12      13      14      15      16
## -0.9375  0.0625  2.5625  5.8125 -3.1875 -4.1875  1.5625
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        70.0625     1.2640  55.430 3.61e-08 ***
## coded(A)           10.8125     1.2640   8.554 0.000360 ***
## coded(B)            1.5625     1.2640   1.236 0.271297
## coded(C)            4.9375     1.2640   3.906 0.011337 *
## coded(D)            7.3125     1.2640   5.785 0.002172 **
## coded(A):coded(B)   0.0625     1.2640   0.049 0.962478
## coded(A):coded(C)  -9.0625     1.2640  -7.170 0.000821 ***
## coded(A):coded(D)   8.3125     1.2640   6.576 0.001220 **
## coded(B):coded(C)   1.1875     1.2640   0.939 0.390613
## coded(B):coded(D)  -0.1875     1.2640  -0.148 0.887871
## coded(C):coded(D)  -0.5625     1.2640  -0.445 0.674909
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.056 on 5 degrees of freedom
## Multiple R-squared:  0.9777, Adjusted R-squared:  0.9331
## F-statistic: 21.92 on 10 and 5 DF,  p-value: 0.001634
```

Looking at our p-values from our linear model. We see that A, C, D, AC, and AD are all significanlty different and important. Take a look at our plot we notice the exact same results. There is no difference in our linear model and probability plot results.

6.36

Resistivity on a silicon wafer is influenced by several factors. The results of a 2^4 factorial experiment performed during a critical processing step is shown in Table P6.10.

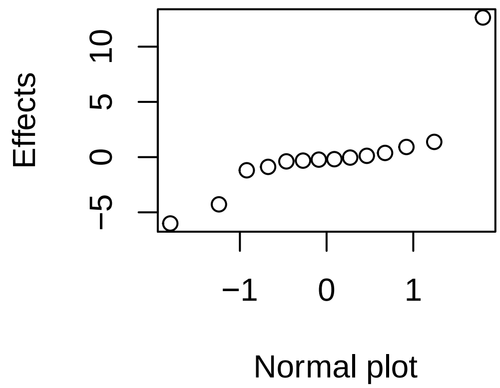
(a) Estimate the factor effects. Plot the effect estimates on a normal probability plot and select a tentative model.

```
# creating data table
A <- rep(c("-", "+"), times = 8)
B <- rep(c("-", "+"), each = 2, times = 4)
C <- rep(c("-", "+"), each = 4, times = 2)
D <- rep(c("-", "+"), each = 8)
Resistivity <- c(1.92, 11.28, 1.09, 5.75, 2.13, 9.53, 1.03, 5.35, 1.60, 11.73, 1.16, 4.68, 2.16, 9.11, 
wafer.long <- data.frame(Resistivity, A, B, C, D)

# defining coded
coded=function(x) #a function to code variable x
{
  ifelse(x=="+", 1, -1)
}

# linear regression
wafer.lm <- lm(Resistivity ~ coded(A) * coded(B) * coded(C) + coded(A) * coded(B) * coded(D) + coded(A)
summary(wafer.lm)
```

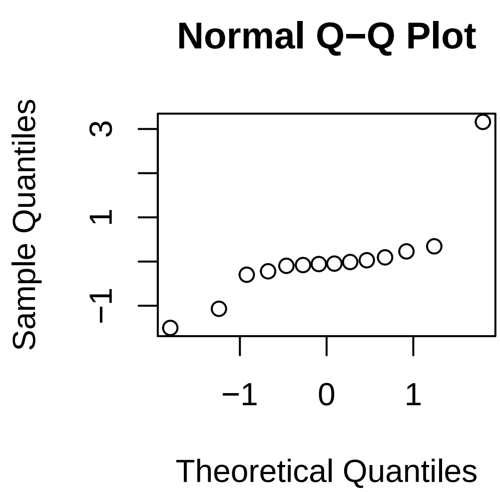
```
##
## Call:
## lm(formula = Resistivity ~ coded(A) * coded(B) * coded(C) + coded(A) *
##     coded(B) * coded(D) + coded(A) * coded(C) * coded(D) + coded(B) *
##     coded(C) * coded(D), data = wafer.long)
##
## Residuals:
##       1       2       3       4       5       6       7       8       9
##  0.1419 -0.1419 -0.1419  0.1419 -0.1419  0.1419  0.1419 -0.1419 -0.1419
##      10      11      12      13      14      15      16
##  0.1419  0.1419 -0.1419  0.1419 -0.1419 -0.1419  0.1419
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4.68062    0.14188  32.991   0.0193 *
## coded(A)                     3.16062    0.14188  22.278   0.0286 *
## coded(B)                    -1.50187    0.14188 -10.586   0.0600 .
## coded(C)                    -0.22062    0.14188  -1.555   0.3638
## coded(D)                    -0.07937    0.14188  -0.559   0.6753
## coded(A):coded(B)           -1.06937    0.14188  -7.537   0.0840 .
## coded(A):coded(C)           -0.29812    0.14188  -2.101   0.2828
## coded(B):coded(C)            0.22937    0.14188   1.617   0.3526
## coded(A):coded(D)           -0.05687    0.14188  -0.401   0.7573
## coded(B):coded(D)           -0.04688    0.14188  -0.330   0.7969
## coded(C):coded(D)            0.02937    0.14188   0.207   0.8700
```

```
## coded(A):coded(B):coded(C)  0.34437    0.14188   2.427   0.2488
## coded(A):coded(B):coded(D) -0.09688    0.14188  -0.683   0.6186
## coded(A):coded(C):coded(D) -0.01063    0.14188  -0.075   0.9524
## coded(B):coded(C):coded(D)  0.09438    0.14188   0.665   0.6263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5675 on 1 degrees of freedom
## Multiple R-squared:  0.9985, Adjusted R-squared:  0.978
## F-statistic: 48.72 on 14 and 1 DF,  p-value: 0.1119
```

```
qqnorm(aov((Resistivity) ~ coded(A) * coded(B)  * coded(C) + coded(A) * coded(B) * coded(D) + coded(A) *
```




```
coef=wafer.lm$coefficients[-1]
variables=names(coef)
plot=qqnorm(coef)
variables[identify(plot)]
```

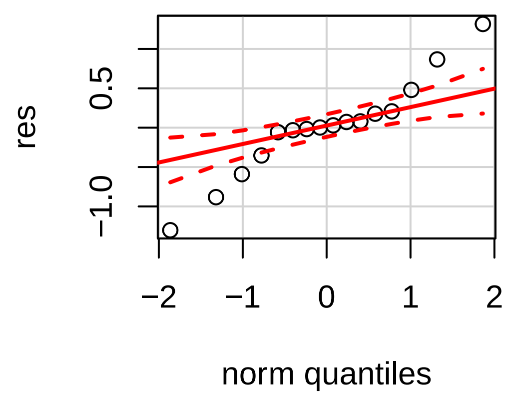


```
## character(0)
```

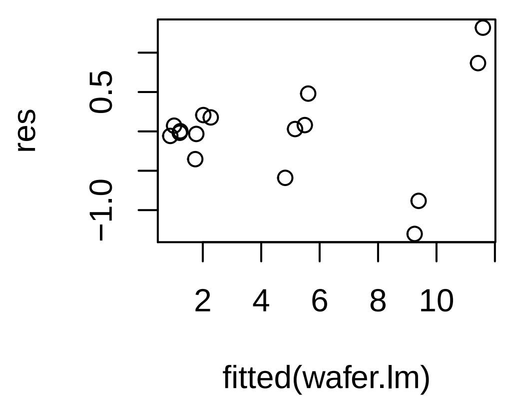
After plotting our ANOVA in a probability model we see that factor A, B, and interaction factor A*B are important. We create a linear model with these results. We look at our results for our model and see all p-values are small and all factors are significantly different in our model.

(b) Fit the model identified in part (a) and analyze the residuals. Is there any indication of model inadequacy.

```r
wafer.lm2 <- lm((Resistivity) ~ coded(A) + coded(B) + coded(A) * coded(B), wafer.long)
res=wafer.long$Resistivity-fitted(wafer.lm2)
qqPlot(res)
```




```r
plot(fitted(wafer.lm), res)
```

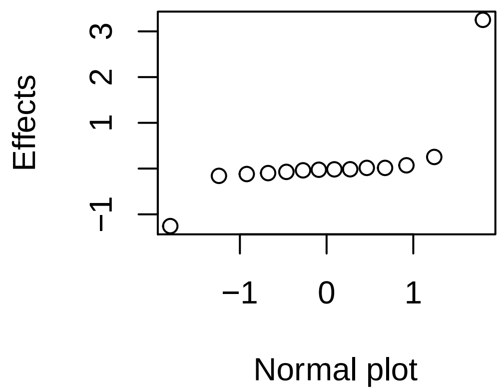


After reviewing normality and residual plot we see that our normality plot is off. As for our residual plot we notice there is some pattern going on. From the normality statement we can state that our model is not good.

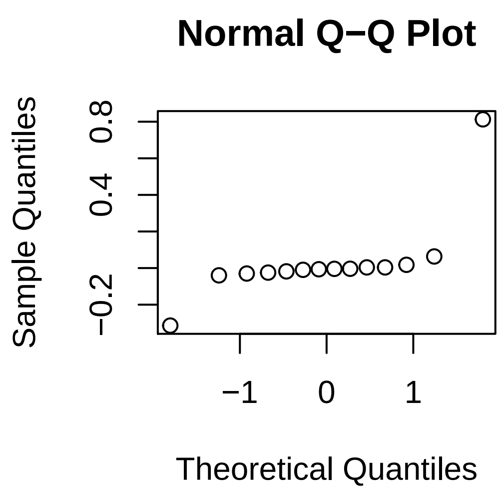
(c) Repeat the analysis from parts (a) and (b) using ln(y) as the response variable. Is there

an indication that the transforation has been useful?

```
wafer.lm3 <- lm(log(Resistivity) ~ coded(A) * coded(B) * coded(C) + coded(A) * coded(B) * coded(D) + co

qqnorm(aov(log(Resistivity) ~ coded(A) * coded(B)  * coded(C) + coded(A) * coded(B) * coded(D) + coded(
```

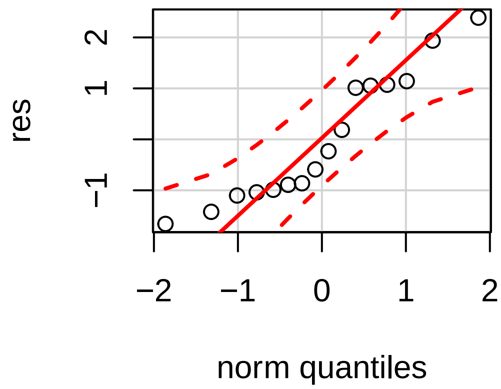



```
coef=wafer.lm3$coefficients[-1]
variables=names(coef)
plot=qqnorm(coef)
variables[identify(plot)]
```

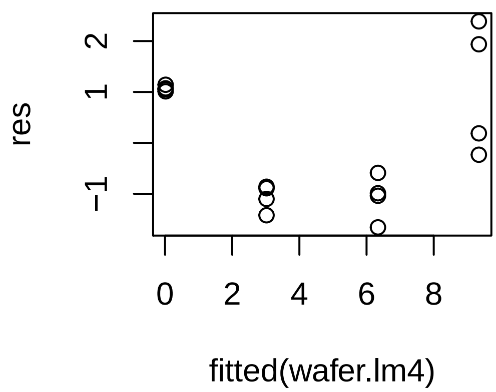



```
## character(0)
```

```
##A and B
##
wafer.lm4 = lm(Resistivity ~ coded(A) + coded(B), wafer.long)
res=wafer.long$Resistivity-fitted(wafer.lm4)
qqPlot(res)
```




```
plot(fitted(wafer.lm4), res)
```



Based on the suggestion of the exercise. We do see an improvement in our model when it comes to checking our normality and residuals. Our model is better.

(d) Fit a model in terms of the coded variables that can be used to predict the resistivity.

```
wafer.lm4 = lm(Resistivity ~ coded(A) + coded(B), wafer.long); summary(wafer.lm4)
```

```
##
## Call:
## lm(formula = Resistivity ~ coded(A) + coded(B), data = wafer.long)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6594 -1.0019 -0.4113  1.0569  2.3869
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.6806     0.3401  13.764 3.97e-09 ***
## coded(A)      3.1606     0.3401   9.294 4.18e-07 ***
## coded(B)     -1.5019     0.3401  -4.416 0.000696 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.36 on 13 degrees of freedom
## Multiple R-squared:  0.8906, Adjusted R-squared:  0.8738
## F-statistic: 52.94 on 2 and 13 DF,  p-value: 5.655e-07
```

After performing part (c) and checking model adequacy we can state that a model with factor A and factor B is a good model.

6.37

Continuation of Problem 6.36. Suppose that the experimenter had also run four center points along with the 16 runs in Problem 6.36. The resistivity measurements at the center points are 8.15, 7.63, 8.95, and 6.48. Analyze the experiment again incorporating the center points. What conclusions can you draw now?

```r
# creating data table
A <- c(rep(c(-1, 1), times = 8), 0, 0, 0, 0)
B <- c(rep(c(-1, 1), each = 2, times = 4), 0, 0, 0, 0)
C <- c(rep(c(-1, 1), each = 4, times = 2), 0, 0, 0, 0)
D <- c(rep(c(-1, 1), each = 8), 0, 0, 0, 0)
Resistivity <- c(1.92, 11.28, 1.09, 5.75, 2.13, 9.53, 1.03, 5.35, 1.60, 11.73, 1.16, 4.68, 2.16, 9.11,
wafer.long2 <- data.frame(Resistivity, A, B, C, D)

# center vs. factorial averages
yc_bar <- mean(wafer.long2$Resistivity[1:16])
test <- mean(wafer.long2$Resistivity[17:20])

# linear regression
wafer.lm21 <- lm(Resistivity ~ A * B * C * D + I(A^2) + I(B^2) + I(C^2) + I(D^2), wafer.long2)
summary(wafer.lm21)
```

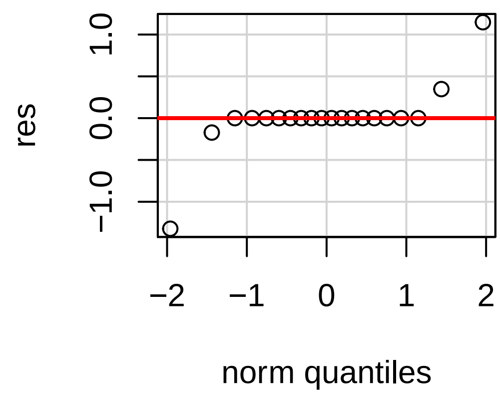
```
##
## Call:
## lm(formula = Resistivity ~ A * B * C * D + I(A^2) + I(B^2) +
##     I(C^2) + I(D^2), data = wafer.long2)
##
## Residuals:
##          1          2          3          4          5          6
## -5.551e-17  3.733e-18 -4.009e-17 -6.194e-19 -9.918e-18 -7.483e-18
##          7          8          9         10         11         12
##  3.342e-17 -1.128e-18  4.386e-17 -3.649e-17 -5.293e-18  1.096e-17
##         13         14         15         16         17         18
## -2.311e-17  2.824e-17 -6.202e-18 -4.708e-17  3.475e-01 -1.725e-01
##         19         20
##  1.147e+00 -1.322e+00
##
## Coefficients: (3 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  7.80250    0.51771  15.071 0.000634 ***
## A             3.16062    0.25885  12.210 0.001183 **
## B            -1.50187    0.25885  -5.802 0.010189 *
## C            -0.22063    0.25885  -0.852 0.456677
## D            -0.07937    0.25885  -0.307 0.779169
## I(A^2)       -3.12187    0.57882  -5.394 0.012490 *
## I(B^2)             NA         NA      NA       NA
## I(C^2)             NA         NA      NA       NA
## I(D^2)             NA         NA      NA       NA
## A:B          -1.06937    0.25885  -4.131 0.025731 *
## A:C          -0.29812    0.25885  -1.152 0.332897
## B:C           0.22938    0.25885   0.886 0.440821
## A:D          -0.05687    0.25885  -0.220 0.840192
## B:D          -0.04688    0.25885  -0.181 0.867843
## C:D           0.02937    0.25885   0.113 0.916818
## A:B:C         0.34437    0.25885   1.330 0.275482
## A:B:D        -0.09688    0.25885  -0.374 0.733109
## A:C:D        -0.01063    0.25885  -0.041 0.969838
## B:C:D         0.09437    0.25885   0.365 0.739604
## A:B:C:D       0.14188    0.25885   0.548 0.621781
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.035 on 3 degrees of freedom
## Multiple R-squared:  0.9874, Adjusted R-squared:  0.9199
## F-statistic: 14.64 on 16 and 3 DF,  p-value: 0.024
```

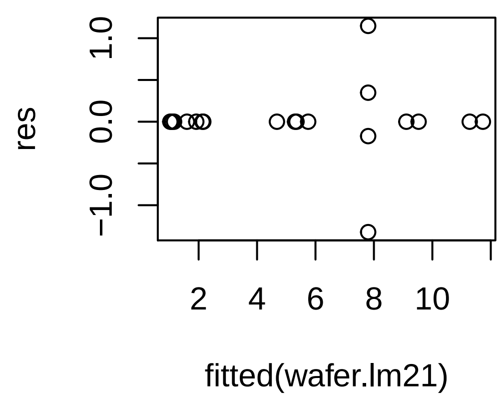
```r
# ANOVA test
wafer.aov21 <- aov(Resistivity ~ A * B * C * D + I(A^2) + I(B^2) + I(C^2) + I(D^2), wafer.long2)
summary(wafer.aov21)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## A            1 159.83  159.83 149.085 0.00118 **
## B            1  36.09   36.09  33.663 0.01019 *
## C            1   0.78    0.78   0.726 0.45668
## D            1   0.10    0.10   0.094 0.77917
## I(A^2)       1  31.19   31.19  29.090 0.01249 *
## A:B          1  18.30   18.30  17.067 0.02573 *
## A:C          1   1.42    1.42   1.326 0.33290
## B:C          1   0.84    0.84   0.785 0.44082
## A:D          1   0.05    0.05   0.048 0.84019
## B:D          1   0.04    0.04   0.033 0.86784
## C:D          1   0.01    0.01   0.013 0.91682
## A:B:C        1   1.90    1.90   1.770 0.27548
## A:B:D        1   0.15    0.15   0.140 0.73311
## A:C:D        1   0.00    0.00   0.002 0.96984
## B:C:D        1   0.14    0.14   0.133 0.73960
## A:B:C:D      1   0.32    0.32   0.300 0.62178
## Residuals    3   3.22    1.07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# checking model adequacy
res=wafer.long2$Resistivity-fitted(wafer.lm21)
qqPlot(res)
```




```r
plot(fitted(wafer.lm21), res)
```




```r
# Rebuild model
wafer.lm22 <- lm(Resistivity ~ A * B + I(A^2) + I(B^2), wafer.long2)
summary(wafer.lm22)
```

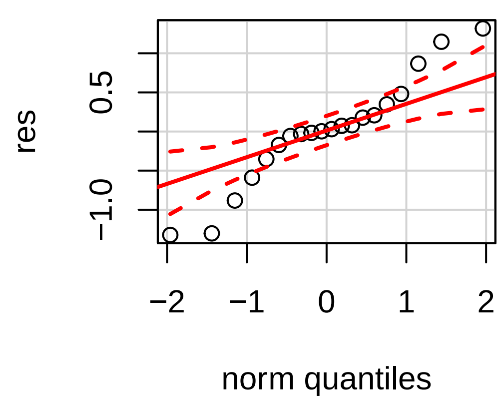
```
##
## Call:
## lm(formula = Resistivity ~ A * B + I(A^2) + I(B^2), data = wafer.long2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.32250 -0.21750  0.01625  0.24250  1.31750
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.8025     0.3867  20.175 2.78e-12 ***
```

```
## A              3.1606     0.1934  16.345 5.74e-11 ***
## B             -1.5019     0.1934  -7.767 1.24e-06 ***
## I(A^2)        -3.1219     0.4324  -7.220 2.97e-06 ***
## I(B^2)             NA         NA      NA       NA
## A:B           -1.0694     0.1934  -5.530 5.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7735 on 15 degrees of freedom
## Multiple R-squared:  0.9647, Adjusted R-squared:  0.9553
## F-statistic: 102.5 on 4 and 15 DF,  p-value: 1.052e-10
```

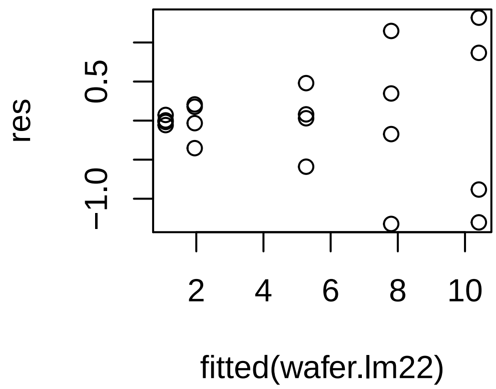
```
wafer.aov22 <- aov(Resistivity ~ A * B + I(A^2) + I(B^2), wafer.long2)
summary(wafer.aov22)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## A            1 159.83  159.83  267.14 5.74e-11 ***
## B            1  36.09   36.09   60.32 1.24e-06 ***
## I(A^2)       1  31.19   31.19   52.13 2.97e-06 ***
## A:B          1  18.30   18.30   30.58 5.77e-05 ***
## Residuals   15   8.97    0.60
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# checking model adequacy
res=wafer.long2$Resistivity-fitted(wafer.lm22)
qqPlot(res)
```




```
plot(fitted(wafer.lm22), res)
```



With the introduction of center points to our dataset, we find that therer is statistically significant evidence of curvature in our model. In addition, A, B, and AB are also statistically significant to our model. Residual analysis tells us that our model is inadquate.