

Part I. (125 points) I recommend reading through all the parts of the HW (with my adjustments) before starting; this may save you some work.

MMA-RSM Chapter 3: 3.1, 3.6, 3.7, 3.11, 3.12, 3.20, 3.26.

- For 3.11 replace part (b) with (d).
- For 3.20 (c), one of the alternative observations for Rep IV is wrong. Indicate what it should be, then answer the question.
- Where applicable, use the calibrated Lenth procedure to assess significance with SMOE to pool errors.

General: Try to do all calculations in R. All R code for the assignment should be included with the part of the problem it addresses (for code and output use a fixed-width font, such as Courier). Code is used to calculate result; text is used to report and interpret results – do not report or interpret results in the code.

- (20^{pts}) 1. **3.1** A router is used to cut registration notches on a printed circuit board. The vibration 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 inch) and two speeds (40 and 90 rpm) are selected, and four boards are cut at each set of conditions as shown in Table E3.1. The response variable is vibration measured as the resultant vector of three accelerometers (x , y , and z) on each test circuit board.

- (a) (5 pts) Analyze the data from this experiment.

Solution: Read data.

```
#### 3.1
fn.data <- "http://statacumen.com/teach/RSM/data/RSM_HW_03-01.txt"
df.3.1 <- read.table(fn.data, header=TRUE)
str(df.3.1)

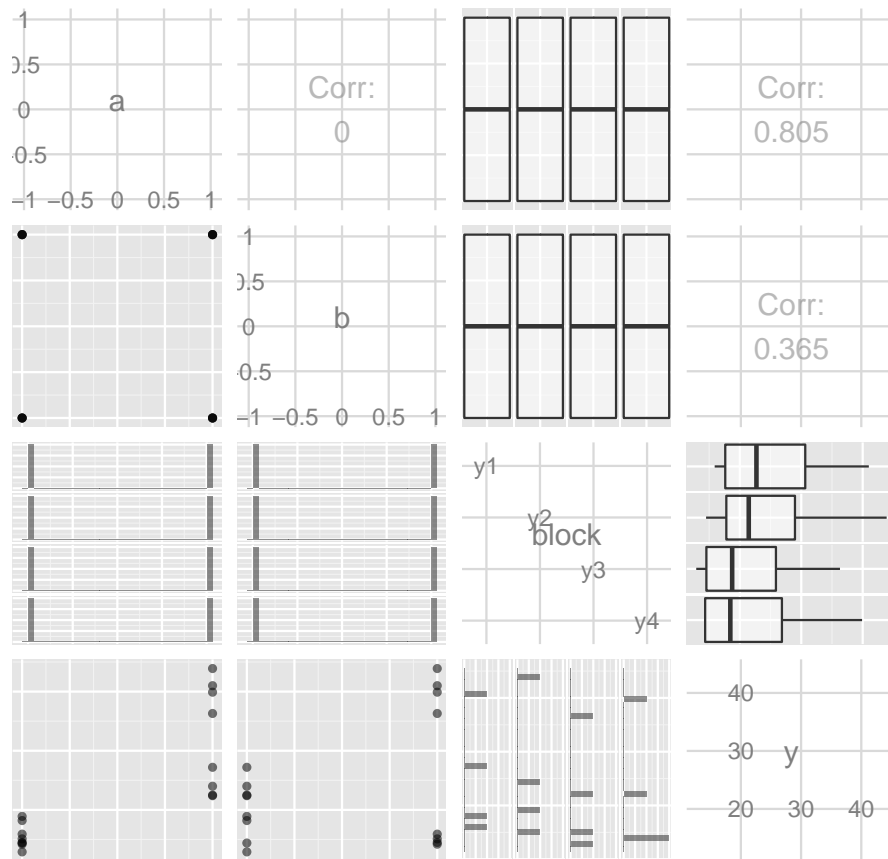
## 'data.frame': 4 obs. of 6 variables:
## $ a : int -1 1 -1 1
## $ b : int -1 -1 1 1
## $ y1: num 18.2 27.2 15.9 41
## $ y2: num 18.9 24 14.5 43.9
## $ y3: num 12.9 22.4 15.1 36.3
## $ y4: num 14.4 22.5 14.2 39.9

# reshape data into long format
library(reshape2)
df.3.1 <- melt(df.3.1, id.vars = c("a", "b"), variable.name = "block", value.name = "y")
```

Scatterplot matrix shows some relationships between PITCH and other variables.

```
library(ggplot2)
suppressMessages(suppressWarnings(library(GGally)))
p <- ggpairs(df.3.1, alpha = 0.1)
# put scatterplots on top so y axis is vertical
#p <- ggpairs(df.3.1, upper = list(continuous = "points")
#           , lower = list(continuous = "cor")
#           )
print(p)

# detach package after use so reshape2 works (old reshape (v.1) conflicts)
detach("package:GGally", unload=TRUE)
detach("package:reshape", unload=TRUE)
```



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

```
library(rsm)
rsm.3.1.y.TWIab <- rsm(y ~ FO(a, b) + TWI(a, b), data = df.3.1)
# externally Studentized residuals
rsm.3.1.y.TWIab$residuals <- rstudent(rsm.3.1.y.TWIab)
summary(rsm.3.1.y.TWIab)

##
## Call:
## rsm(formula = y ~ FO(a, b) + TWI(a, b), data = df.3.1)
##
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  23.831     0.611   38.99 5.2e-14 ***
## a             8.319     0.611   13.61 1.2e-08 ***
## b             3.769     0.611    6.17 4.8e-05 ***
## a:b           4.356     0.611    7.13 1.2e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.958, Adjusted R-squared:  0.948
## F-statistic: 91.4 on 3 and 12 DF,  p-value: 1.57e-08
##
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value Pr(>F)
## FO(a, b)    2  1334    667  111.6 1.8e-08
## TWI(a, b)   1   304    304   50.8 1.2e-05
## Residuals  12    72     6
## Lack of fit 0     0
```

```
## Pure error 12 72 6
##
## Stationary point of response surface:
## a b
## -0.8651 -1.9096
##
## Eigenanalysis:
## $values
## [1] 2.178 -2.178
##
## $vectors
## [,1] [,2]
## a 0.7071 -0.7071
## b 0.7071 0.7071
```

The ANOVA indicates that the model is significant with an F statistic of 91.36 and a p-value of 1.569×10^{-8} . The parameter estimate table indicates that the two main effects, A and B , and their interaction, AB , are all highly significantly different from zero.

- (b) (5 pts) Construct a normal probability plot of the residuals and a plot of the residuals versus the predicted vibration level. Interpret these plots.

Solution: The diagnostic plots are below. The pattern of the residuals is consistent with the common variance assumption. The residuals are sufficiently normal. No observations have relatively overwhelmingly large Cook's D .

```
# plot diagnostics
par(mfrow=c(2,3))

plot(df.3.1$a, rsm.3.1.y.TWIab$studres, main="Residuals vs a")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(df.3.1$b, rsm.3.1.y.TWIab$studres, main="Residuals vs b")
# horizontal line at zero
abline(h = 0, col = "gray75")

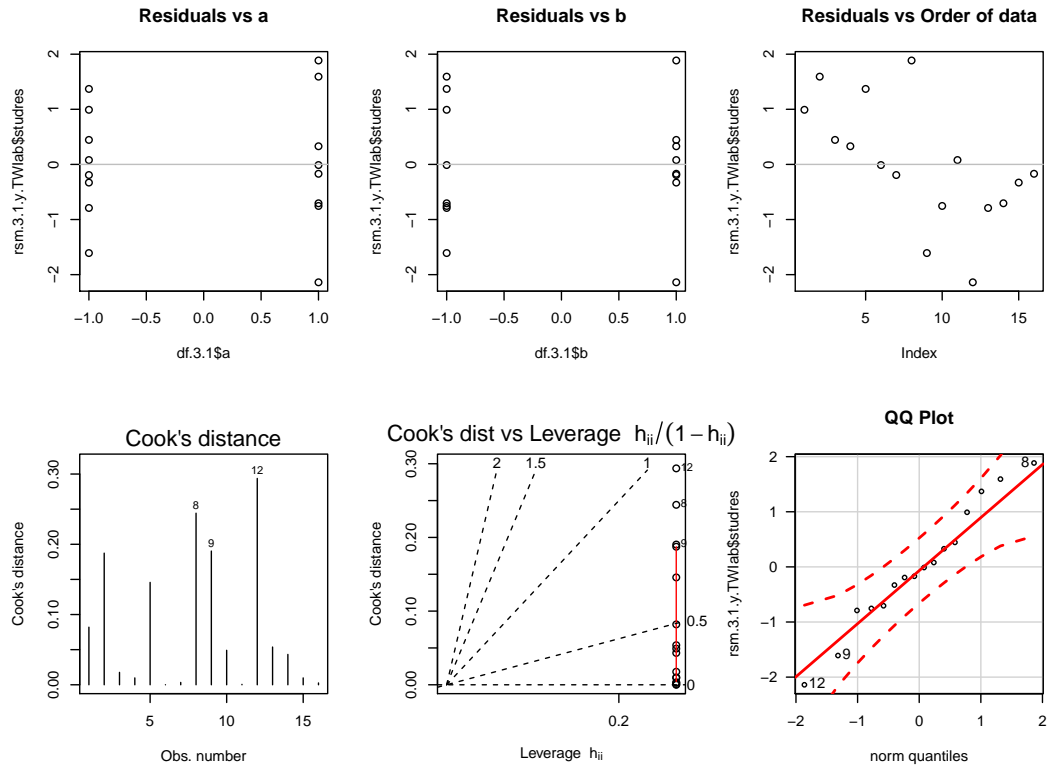
# residuals vs order of data
plot(rsm.3.1.y.TWIab$studres, main="Residuals vs Order of data")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(rsm.3.1.y.TWIab, which = c(4,6))

# Normality of Residuals
library(car)
qqPlot(rsm.3.1.y.TWIab$studres, las = 1, id.n = 3, main="QQ Plot")

## 12 8 9
## 1 16 2

cooks.distance(rsm.3.1.y.TWIab)
## 1 2 3 4 5 6 7 8 9
## 8.198e-02 1.874e-01 1.767e-02 9.771e-03 1.457e-01 1.162e-05 3.358e-03 2.443e-01 1.904e-01
## 10 11 12 13 14 15 16
## 4.909e-02 5.693e-04 2.937e-01 5.373e-02 4.323e-02 9.771e-03 2.614e-03
```

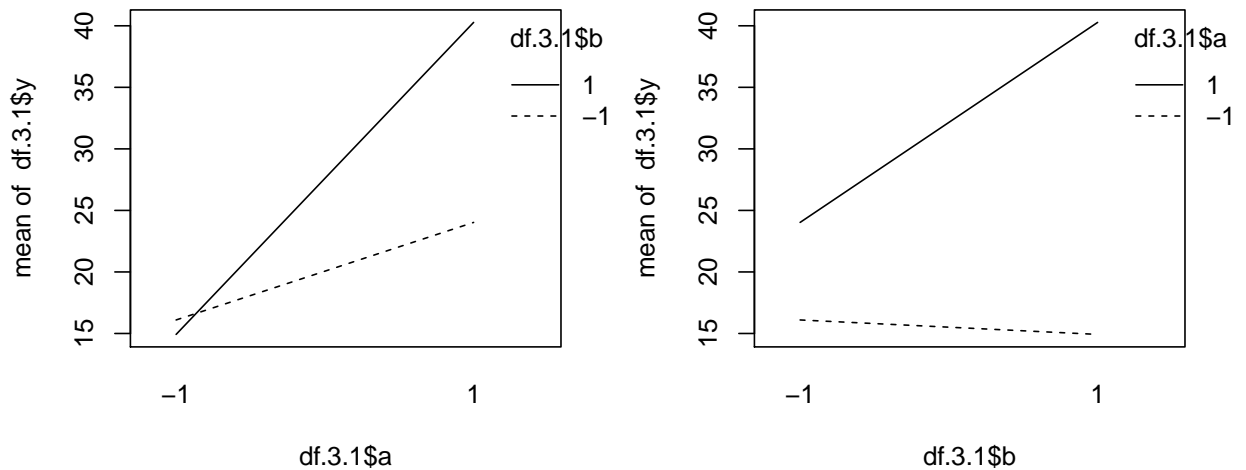


- (c) (5 pts) Draw the AB interaction plot. Interpret this plot. What levels of bit size and speed would you recommend for routine operation?

Solution: The interaction plots below indicate that when a is at its low level, the vibration stays roughly the same for the low and high level of b , but when a is at its high level, the vibration increases substantially as b goes from its low to high level.

Setting a at the low level and then b at a convenient level seems like a sensible choice to reduce vibration.

```
# interaction plot
par(mfrow=c(1,2))
interaction.plot(df.3.1$a, df.3.1$b, df.3.1$y)
interaction.plot(df.3.1$b, df.3.1$a, df.3.1$y)
```



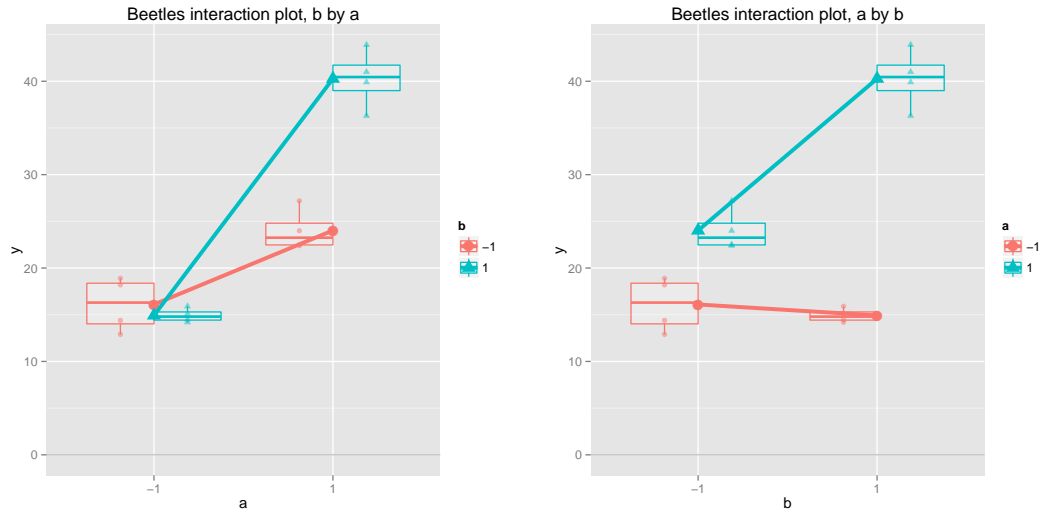
```
# Interaction plots, ggplot
library(plyr)
# Calculate the cell means for each (a, b) combination
# create factor version for ggplot categories
df.3.1.factor <- df.3.1
df.3.1.factor$a <- factor(df.3.1.factor$a)
df.3.1.factor$b <- factor(df.3.1.factor$b)

#mean(df.3.1.factor[, "y"])
df.3.1.factor.mean <- ddply(df.3.1.factor, .(), summarise, m = mean(y))
#df.3.1.factor.mean
df.3.1.factor.mean.d <- ddply(df.3.1.factor, .(a), summarise, m = mean(y))
#df.3.1.factor.mean.d
df.3.1.factor.mean.i <- ddply(df.3.1.factor, .(b), summarise, m = mean(y))
#df.3.1.factor.mean.i
df.3.1.factor.mean.di <- ddply(df.3.1.factor, .(a,b), summarise, m = mean(y))
#df.3.1.factor.mean.di

p <- ggplot(df.3.1.factor, aes(x = a, y = y, colour = b, shape = b))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
, linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = df.3.1.factor.mean.di, aes(y = m), size = 4)
p <- p + geom_line(data = df.3.1.factor.mean.di, aes(y = m, group = b), size = 1.5)
p <- p + labs(title = "Beetles interaction plot, b by a")
print(p)

## ymax not defined: adjusting position using y instead
p <- ggplot(df.3.1.factor, aes(x = b, y = y, colour = a, shape = a))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
, linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = df.3.1.factor.mean.di, aes(y = m), size = 4)
p <- p + geom_line(data = df.3.1.factor.mean.di, aes(y = m, group = a), size = 1.5)
p <- p + labs(title = "Beetles interaction plot, a by b")
print(p)

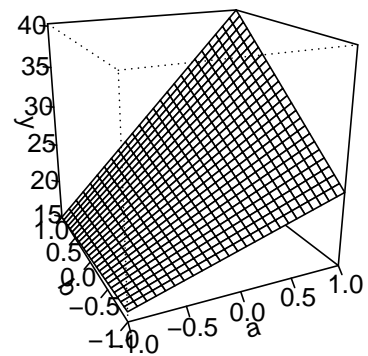
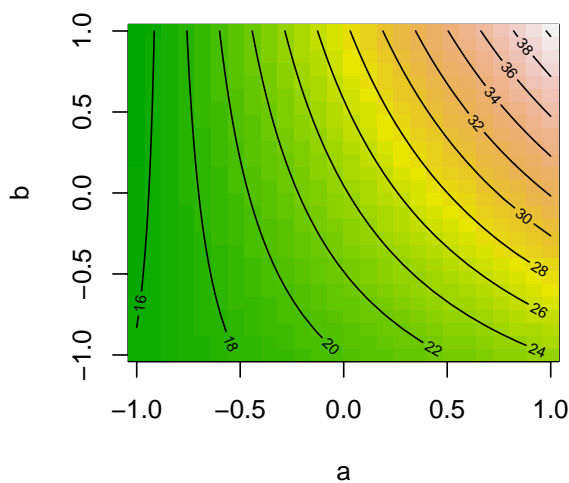
## ymax not defined: adjusting position using y instead
```



- (d) (5 pts) Construct a contour plot of vibration as a function of speed and bit size.

Solution: The contour plot below indicates low vibration is predicted for low values of a regardless of the value of b , however, high vibration with high values of both a and b . Recommended is low bit size and high speed.

```
par(mfrow=c(1,2))
contour(rsm.3.1.y.TWIab, ~ a + b, image = TRUE)
persp(rsm.3.1.y.TWIab, b ~ a, zlab = "y")
```



- (20^{pts}) 2. 3.6 An experiment was performed to improve the yield of a chemical process. Four factors were selected, and one replicate of a completely randomized experiment was run. The results are shown in Table E3.4.

- (a) (5 pts) Estimate the factor effects. Construct a normal probability plot of these effects. Which effects appear large?

To assess significance use the calibrated Lenth procedure with SMOE to pool errors.

Solution: Read data.

```
#### 3.6
fn.data <- "http://statacumen.com/teach/RSM/data/RSM_HW_03-06.txt"
df.3.6 <- read.table(fn.data, header=TRUE)
str(df.3.6)

## 'data.frame': 16 obs. of 5 variables:
## $ a: int -1 1 -1 1 -1 1 -1 1 -1 1 ...
## $ b: int -1 -1 1 1 -1 -1 1 1 -1 -1 ...
## $ c: int -1 -1 -1 -1 1 1 1 1 -1 -1 ...
## $ d: int -1 -1 -1 -1 -1 -1 -1 -1 1 1 ...
## $ y: int 90 64 81 63 77 61 88 53 98 62 ...
```

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

```
lm.3.6.y.4WIabcd <- lm(y ~ (a + b + c + d)^4, data = df.3.6)
## externally Studentized residuals
#lm.3.6.y.4WIabcd$residuals <- rstudent(lm.3.6.y.4WIabcd)
#summary(lm.3.6.y.4WIabcd)
```

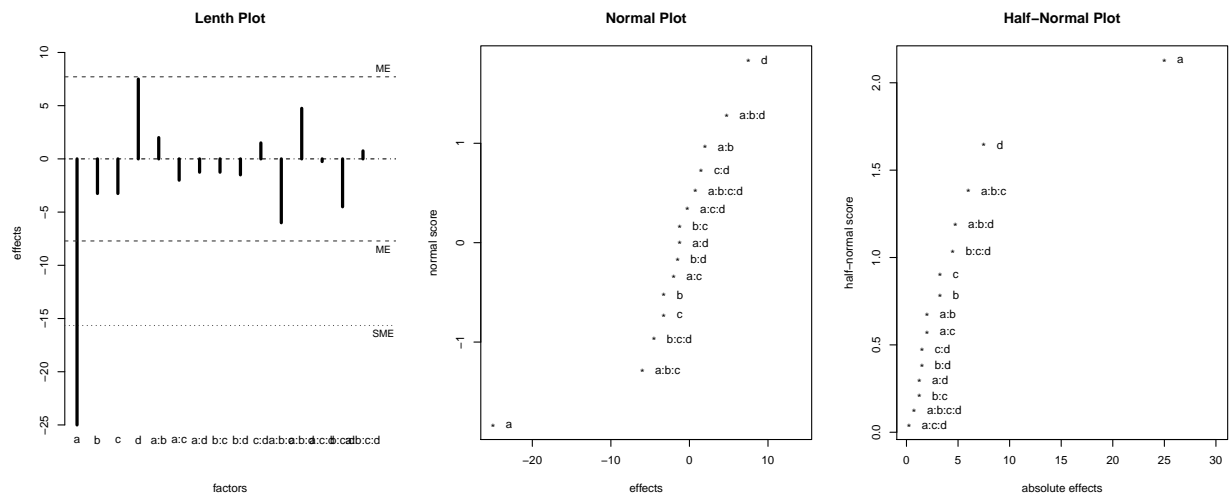
The Lenth and normal plots below indicate that a is by far the most important effect, followed by a small effect d .

I choose to project the design to a 2^2 replicated design in factors a and d .

```
# BsMD package has unreplicated factorial tests (Daniel plots (aka normal), and Lenth)
library(BsMD)
par(mfrow=c(1,3))
LenthPlot(lm.3.6.y.4WIabcd, alpha = 0.05, main = "Lenth Plot") # , adj = 0.2

## alpha PSE ME SME
## 0.050 3.000 7.712 15.656

DanielPlot(lm.3.6.y.4WIabcd, main = "Normal Plot")
DanielPlot(lm.3.6.y.4WIabcd, half = TRUE, main = "Half-Normal Plot")
```



- (b) (5 pts) Prepare an analysis of variance table using the information obtained in part (a).

Solution: I run a regression in factors a and d , with interaction.

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

```
library(rsm)
rsm.3.6.y.TWIad <- rsm(y ~ FO(a, d) + TWI(a, d), data = df.3.6)
## externally Studentized residuals
rsm.3.6.y.TWIad$residuals <- rstudent(rsm.3.6.y.TWIad)
summary(rsm.3.6.y.TWIad)

##
## Call:
## rsm(formula = y ~ FO(a, d) + TWI(a, d), data = df.3.6)
```

```
##
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  75.875      1.545   49.10 3.3e-15 ***
## a           -12.500      1.545  -8.09 3.4e-06 ***
## d            3.750      1.545   2.43  0.032 *
## a:d          -0.625      1.545  -0.40  0.693
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.856, Adjusted R-squared:  0.82
## F-statistic: 23.8 on 3 and 12 DF,  p-value: 2.42e-05
##
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value Pr(>F)
## FO(a, d)   2  2725    1362    35.66 8.9e-06
## TWI(a, d)   1     6         6     0.16  0.69
## Residuals  12   459         38
## Lack of fit  0     0         Inf
## Pure error  12   459         38
##
## Stationary point of response surface:
##   a   d
##  6 -20
##
## Eigenanalysis:
## $values
## [1]  0.3125 -0.3125
##
## $vectors
##      [,1] [,2]
## a -0.7071 -0.7071
## d  0.7071 -0.7071
```

The plot below provides diagnostic plots. The pattern of the residuals is consistent with the constant variance assumption. However, there is a suggestion that the residuals are not normal, they have short tails. Otherwise, model assumptions sufficiently hold.

```
# plot diagnostics
par(mfrow=c(2,3))

plot(df.3.6$a, rsm.3.6.y.TWIad$studres, main="Residuals vs a")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(df.3.6$d, rsm.3.6.y.TWIad$studres, main="Residuals vs d")
# horizontal line at zero
abline(h = 0, col = "gray75")

# residuals vs order of data
plot(rsm.3.6.y.TWIad$studres, main="Residuals vs Order of data")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(rsm.3.6.y.TWIad, which = c(4,6))

# Normality of Residuals
library(car)
qqPlot(rsm.3.6.y.TWIad$studres, las = 1, id.n = 3, main="QQ Plot")

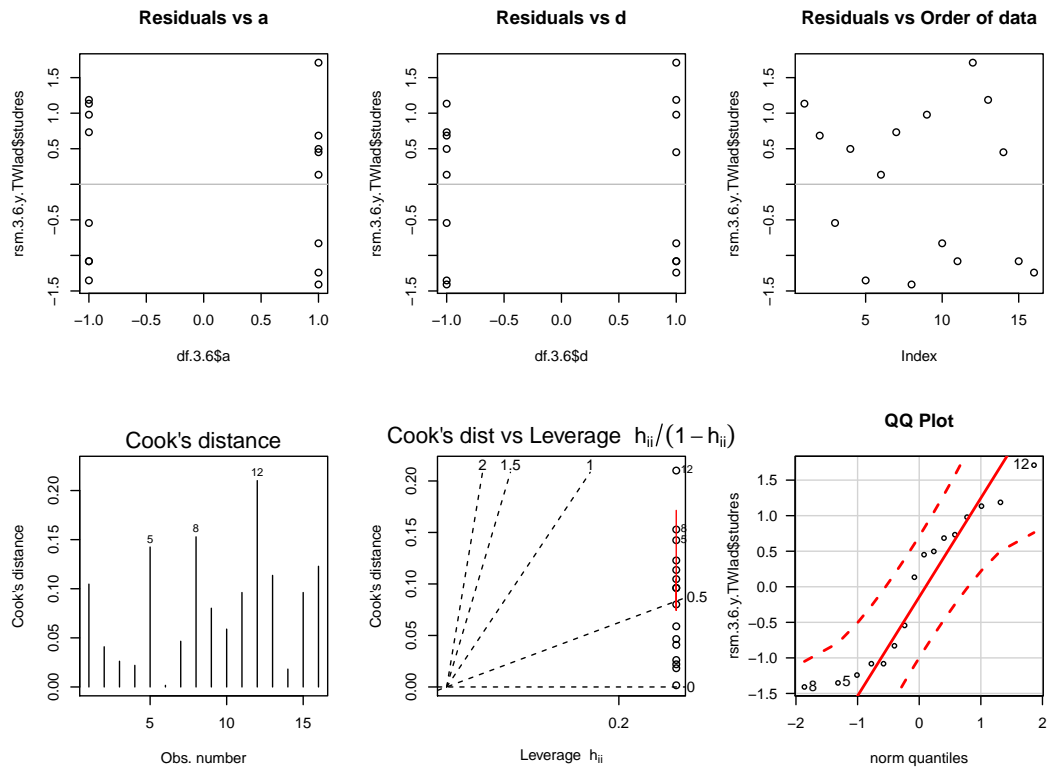
## 12 8 5
## 16 1 2

cooks.distance(rsm.3.6.y.TWIad)

##      1      2      3      4      5      6      7      8      9     10
## 0.104689 0.040894 0.026172 0.021992 0.142494 0.001636 0.046529 0.152854 0.080153 0.058888
##     11     12     13     14     15     16
```



```
## 0.096147 0.210105 0.113595 0.018175 0.096147 0.122864
```



The interaction is not significant.

Refitting the model with main effects only has a highly significant and d marginally significant. The residual plots look similar for this model.

```
library(rsm)
rsm.3.6.y.FOad <- rsm(y ~ FO(a, d), data = df.3.6)
# externally Studentized residuals
rsm.3.6.y.FOad$studres <- rstudent(rsm.3.6.y.FOad)
summary(rsm.3.6.y.FOad)

##
## Call:
## rsm(formula = y ~ FO(a, d), data = df.3.6)
##
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   75.88      1.49   50.76 2.5e-16 ***
## a             -12.50      1.49  -8.36 1.4e-06 ***
## d              3.75      1.49   2.51 0.026 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.854, Adjusted R-squared:  0.832
## F-statistic: 38.1 on 2 and 13 DF,  p-value: 3.65e-06
##
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value Pr(>F)
## FO(a, d)   2  2725   1362   38.11 3.7e-06
## Residuals 13    465     36
## Lack of fit 1     6      6    0.16  0.69
## Pure error 12    459     38
##
```

```
## Direction of steepest ascent (at radius 1):
##      a      d
## -0.9578 0.2873
##
## Corresponding increment in original units:
##      a      d
## -0.9578 0.2873
```

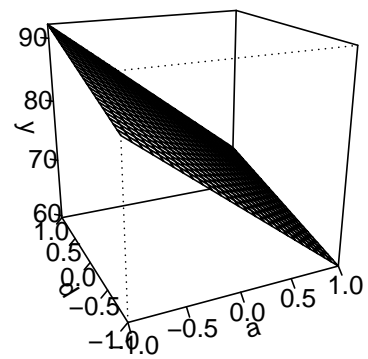
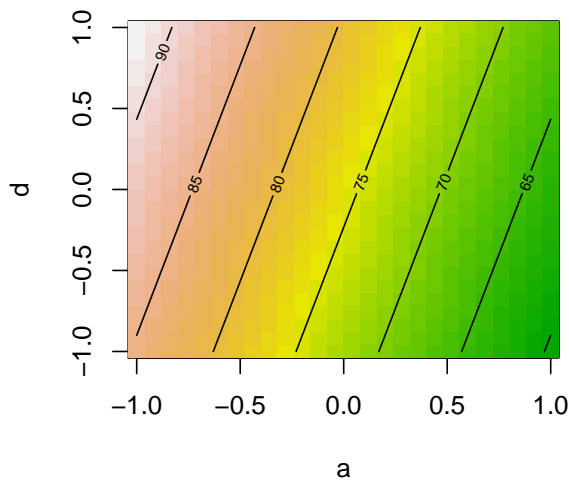
- (c) (5 pts) Plot the residuals versus the predicted yield and on normal probability paper. Does the residual analysis appear satisfactory?

Solution: As above, the residual normal plots are not normal, having short tails.

- (d) (5 pts) Construct a contour plot of yield as a function of the important process variables.

Solution: The contour plot in Figure ?? indicates chemical yield is improved with low values of a and high values of d .

```
par(mfrow=c(1,2))
contour(rsm.3.6.y.F0ad, ~ a + d, image = TRUE)
persp(rsm.3.6.y.F0ad, d ~ a, zlab = "y")
```



- (10^{pts}) **3. 3.7** Consider the design of Exercise 3.6. Suppose that four additional runs were made at the center of the region of experimentation. The response values at these center points were 94, 90, 99, and 87. Use this additional information to test for curvature in the response function. What are your conclusions?

Solution: Read data.

```
#### 3.7
fn.data <- "http://statacumen.com/teach/RSM/data/RSM_HW_03-06.txt"
df.3.7 <- read.table(fn.data, header=TRUE)
df.3.7 <- rbind(df.3.7
  , data.frame(
    a = rep(0,4)
  , b = rep(0,4)
  , c = rep(0,4)
  , d = rep(0,4)
  , y = c(94, 90, 99, 87)
  )
```

```

)
tail(df.3.7)
##      a b c d y
## 15 -1 1 1 1 87
## 16  1 1 1 1 60
## 17  0 0 0 0 94
## 18  0 0 0 0 90
## 19  0 0 0 0 99
## 20  0 0 0 0 87

# label factorial and center points
df.3.7$point <- "factorial"
df.3.7$point[which(!apply(subset(df.3.7, select = c(a,b,c,d)), 1, all))] <- "center"

## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
## Warning: coercing argument of type 'double' to logical
df.3.7$point <- factor(df.3.7$point)
str(df.3.7)

## 'data.frame': 20 obs. of 6 variables:
## $ a : num -1 1 -1 1 -1 1 -1 1 -1 1 ...
## $ b : num -1 -1 1 1 -1 -1 1 1 -1 -1 ...
## $ c : num -1 -1 -1 -1 1 1 1 1 -1 -1 ...
## $ d : num -1 -1 -1 -1 -1 -1 -1 -1 1 1 ...
## $ y : num 90 64 81 63 77 61 88 53 98 62 ...
## $ point: Factor w/ 2 levels "center","factorial": 2 2 2 2 2 2 2 2 2 2 ...

```

From p. 112, we calculate the MS_E and $SS_{\text{curvature}}$ and perform an F -test for curvature. The test for curvature can be calculated with R in the same way.

```

## Hypothesis test for curvature
# center runs
y.cen <- subset(df.3.7, (point == "center"))$y
n.cen <- length(y.cen)
n.cen
## [1] 4
y.cen.bar <- mean(y.cen)
y.cen.bar
## [1] 92.5
cen.MSE <- var(y.cen)
cen.MSE
## [1] 27
dfE <- n.cen - 1

# factorial runs
y.fact <- subset(df.3.7, (point == "factorial"))$y
n.fact <- length(y.fact)
n.fact

```

```
## [1] 16
y.fact.bar <- mean(y.fact)
y.fact.bar
## [1] 75.88
SS.Curve <- n.fact * n.cen * (y.fact.bar - y.cen.bar)^2 / (n.fact + n.cen)
SS.Curve
## [1] 884.5
# F-test
F.Curve <- (SS.Curve / 1) / cen.MSE;
F.Curve
## [1] 32.76
F.Curve.p.val <- 1 - pf(F.Curve, 1, dfE)
F.Curve.p.val
## [1] 0.01059
```

$$\begin{aligned}
 MS_E &= \text{Var}[y] = 27 \\
 SS_{\text{Curvature}} &= \frac{n_F n_C (\bar{y}_F - \bar{y}_C)^2}{n_F + n_C} = \frac{16 \times 4 (75.875 - 92.5)^2}{16 + 4} = 884.45 \\
 F &= \frac{SS_{\text{Curvature}}/1}{MS_E} = \frac{884.45/3}{27} = 32.757
 \end{aligned}$$

The $F = 32.76$ with 1 and 3 degrees-of-freedom has an associated p-value = 0.0106. Therefore, at the 0.05 significance level, we reject the null hypothesis in favor of the alternative, concluding we have sufficient evidence of curvature in the experimental region.

A second-order model can now be fit to get a preliminary indication of the direction and degree of curvature.

- (25pts) 4. 3.11 An article in Solid State Technology (“Orthogonal Design for Process Optimization and its Application in Plasma Etching,” May 1987, pp. 127–132) describes the application of factorial designs in developing a nitride etch process on a single-wafer plasma etcher. The process uses C_2F_6 as the reactant gas. Four factors are of interest: anode-cathode gap (A), pressure in the reactor chamber (B), C_2F_6 gas flow (C), and power applied to the cathode (D). The response variable of interest is the etch rate for silicon nitride. A single replicate of a 2^4 design is run, and the resulting data are in Table E3.7.

- (a) (5 pts) Estimate the factor effects. Construct a normal probability plot of the factor effects. Which effects appear large?

Use the calibrated Lenth procedure to assess significance with SMOE to pool errors.

Solution: Read data.

```
#### 3.11
fn.data <- "http://statacumen.com/teach/RSM/data/RSM_HW_03-11.txt"
df.3.11 <- read.table(fn.data, header=TRUE)
str(df.3.11)

## 'data.frame': 16 obs. of 7 variables:
## $ a      : int  -1 1 -1 1 -1 1 -1 1 -1 1 ...
## $ b      : int  -1 -1 1 1 -1 -1 1 1 -1 -1 ...
## $ c      : int  -1 -1 -1 -1 1 1 1 1 -1 -1 ...
## $ d      : int  -1 -1 -1 -1 -1 -1 -1 -1 1 1 ...
## $ y      : int  500 669 604 650 633 642 601 635 1037 749 ...
## $ runnumber: int  1 2 3 4 5 6 7 8 9 10 ...
## $ runorder : int  13 8 12 9 4 15 16 3 1 14 ...
```

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

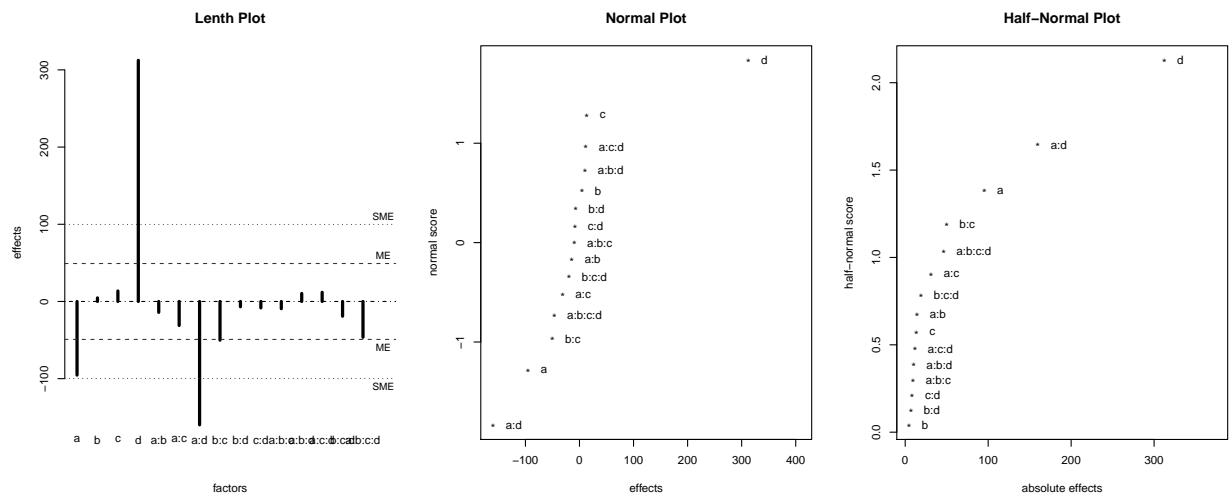
```
lm.3.11.y.4WIabcd <- lm(y ~ (a + b + c + d)^4, data = df.3.11)
## externally Studentized residuals
#lm.3.11.y.4WIabcd$studres <- rstudent(lm.3.11.y.4WIabcd)
#summary(lm.3.11.y.4WIabcd)
```

The Lenth and normal plots below indicate factors a , d , and their interaction, ad , are important factors, while others are unlikely to be.

```
# BsMD package has unreplicated factorial tests (Daniel plots (aka normal), and Lenth)
library(BsMD)
par(mfrow=c(1,3))
LenthPlot(lm.3.11.y.4WIabcd, alpha = 0.05, main = "Lenth Plot") # , adj = 0.2

## alpha PSE ME SME
## 0.05 19.13 49.16 99.81

DanielPlot(lm.3.11.y.4WIabcd, main = "Normal Plot")
DanielPlot(lm.3.11.y.4WIabcd, half = TRUE, main = "Half-Normal Plot")
```



(b) (5 pts) Conduct an analysis of variance to confirm your findings for part (a).

Solution: Fit first-order with two-way interaction linear model of factors a and d .

```
library(rsm)
rsm.3.11.y.TWIad <- rsm(y ~ FO(a, d) + TWI(a, d), data = df.3.11)
# externally Studentized residuals
rsm.3.11.y.TWIad$studres <- rstudent(rsm.3.11.y.TWIad)
summary(rsm.3.11.y.TWIad)

##
## Call:
## rsm(formula = y ~ FO(a, d) + TWI(a, d), data = df.3.11)
##
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   772.9         12.0   64.66 < 2e-16 ***
## a              -47.7         12.0   -3.99  0.0018 **
## d             156.2         12.0  13.07  1.9e-08 ***
## a:d            -79.9         12.0   -6.69  2.2e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.951, Adjusted R-squared:  0.938
## F-statistic: 77.1 on 3 and 12 DF,  p-value: 4.12e-08
##
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value  Pr(>F)
```

```
## F0(a, d)      2 426698 213349 93.3 4.9e-08
## TWI(a, d)    1 102240 102240 44.7 2.2e-05
## Residuals   12 27433 2286
## Lack of fit 0 0 -Inf
## Pure error  12 27433 2286
##
## Stationary point of response surface:
##      a      d
## 1.9539 -0.5966
##
## Eigenanalysis:
## $values
## [1] 39.97 -39.97
##
## $vectors
##      [,1] [,2]
## a -0.7071 -0.7071
## d  0.7071 -0.7071
```

The regression confirms that the regression is significant, with an F statistic of 77.13 and a p -value of 4.124×10^{-8} , and the main effects a and d and their interaction, ad , are all significant.

- (c) (5 pts) Analyze the residuals from this experiment. Comment on the model's adequacy.

Solution:

The third plot in the first row below indicates there may be an issue with homoscedasticity when plotted against predicted values. None of the Cook's D values appear large enough to investigate. The residuals appear slightly leptokurtotic compared to a normal distribution.

```
# plot diagnostics
par(mfrow=c(2,3))

plot(df.3.11$a, rsm.3.11.y.TWIad$studres, main="Residuals vs a")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(df.3.11$d, rsm.3.11.y.TWIad$studres, main="Residuals vs d")
# horizontal line at zero
abline(h = 0, col = "gray75")

## residuals vs order of data
# plot(rsm.3.11.y.TWIad$studres, main="Residuals vs Order of data")
# # horizontal line at zero
# abline(h = 0, col = "gray75")

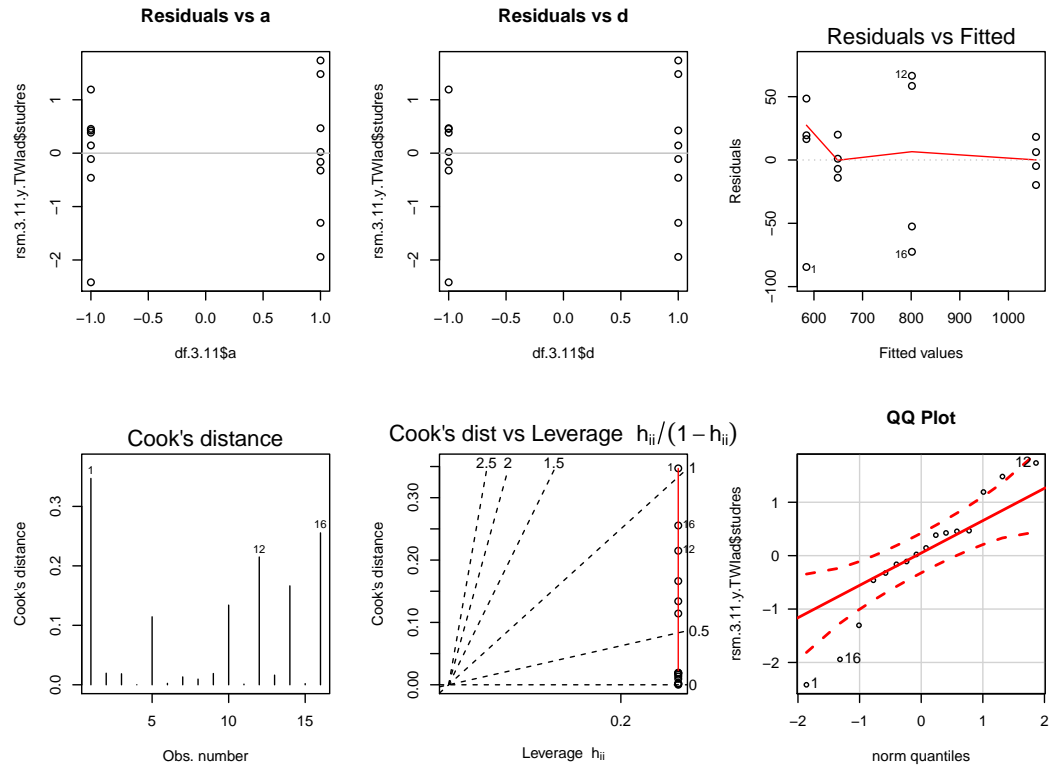
plot(rsm.3.11.y.TWIad, which = c(1,4,6))

# Normality of Residuals
library(car)
qqPlot(rsm.3.11.y.TWIad$studres, las = 1, id.n = 3, main="QQ Plot")

## 1 16 12
## 1 2 16

cooks.distance(rsm.3.11.y.TWIad)

##      1      2      3      4      5      6      7      8      9
## 0.3470426 0.0194415 0.0184816 0.0000486 0.1143281 0.0023816 0.0132324 0.0095263 0.0189585
##      10     11     12     13     14     15     16
## 0.1339640 0.0010966 0.2149377 0.0161881 0.1663340 0.0018986 0.2554732
```



- (d) (0 pts) If not all the factors are important, project the 2^4 design into a 2^k design with $k < 4$ and conduct the analysis of variance.

See part (b).

Solution: See part (b).

- (e) (5 pts) Draw graphs to interpret any significant interactions.

Solution: The interaction plots below indicates that when d is at its low level, the etch rate increases a little when a goes from its low to high level, but when d is at its high level, the etch rate decreases markedly as a goes from its low to high level. Setting a at the low level and then d at its high level seems like a sensible choice to increase the etch rate.

```
# Interaction plots, ggplot
library(plyr)
# Calculate the cell means for each (a, d) combination
# create factor version for ggplot categories
df.3.11.factor <- df.3.11
df.3.11.factor$a <- factor(df.3.11.factor$a)
df.3.11.factor$d <- factor(df.3.11.factor$d)

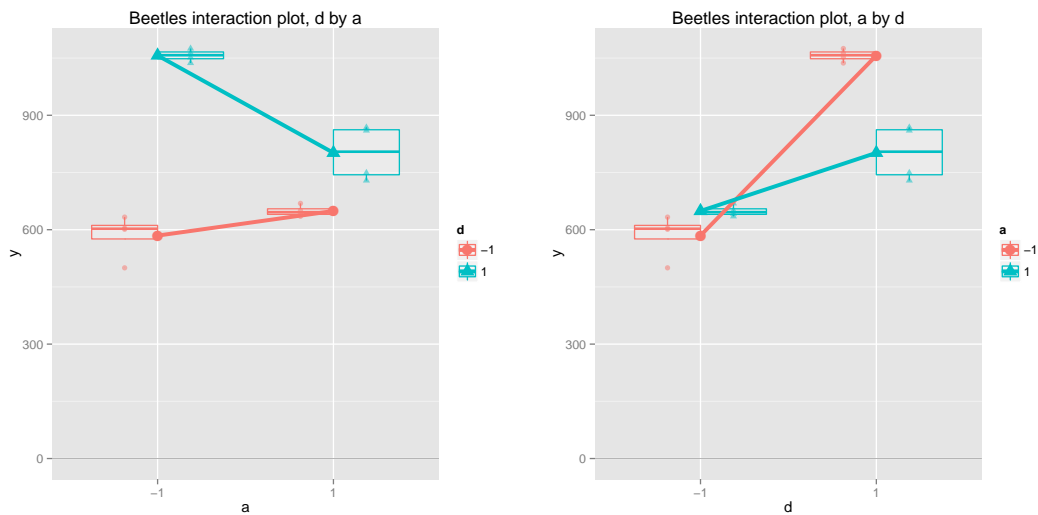
#mean(df.3.11.factor[, "y"])
df.3.11.factor.mean <- ddply(df.3.11.factor, .(), summarise, m = mean(y))
#df.3.11.factor.mean
df.3.11.factor.mean.a <- ddply(df.3.11.factor, .(a), summarise, m = mean(y))
#df.3.11.factor.mean.a
df.3.11.factor.mean.d <- ddply(df.3.11.factor, .(d), summarise, m = mean(y))
#df.3.11.factor.mean.d
df.3.11.factor.mean.di <- ddply(df.3.11.factor, .(a,d), summarise, m = mean(y))
#df.3.11.factor.mean.di

p <- ggplot(df.3.11.factor, aes(x = a, y = y, colour = d, shape = d))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
  , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = df.3.11.factor.mean.di, aes(y = m), size = 4)
p <- p + geom_line(data = df.3.11.factor.mean.di, aes(y = m, group = d), size = 1.5)
p <- p + labs(title = "Beetles interaction plot, d by a")
print(p)

## ymaz not defined: adjusting position using y instead

p <- ggplot(df.3.11.factor, aes(x = d, y = y, colour = a, shape = a))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
  , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
```

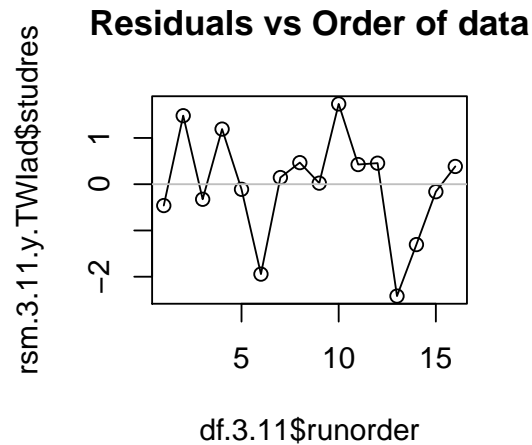
```
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = df.3.11.factor.mean.di, aes(y = m), size = 4)
p <- p + geom_line(data = df.3.11.factor.mean.di, aes(y = m, group = a), size = 1.5)
p <- p + labs(title = "Beetles interaction plot, a by d")
print(p)
## ymaz not defined: adjusting position using y instead
```



- (f) (5 pts) Plot the residuals versus the actual run order. What problems might be revealed by this plot?

Solution: The residual plot below does not seem to indicate a problem. (It is possible there is a negative AR(1) autocorrelation, which seems to be the case for all but two points.) I am relieved that we do not observe a pattern related to the differences in variability of residuals. It is now more likely that the variability of the response is different for different design points.

```
par(mfrow=c(1,1))
# residuals vs order of data
plot(df.3.11$runorder, rsm.3.11.y.TWIad$studres, main="Residuals vs Order of data")
lines(df.3.11$runorder[order(df.3.11$runorder)], rsm.3.11.y.TWIad$studres[order(df.3.11$runorder)])
# horizontal line at zero
abline(h = 0, col = "gray75")
```



- (10^{pts}) 5. 3.12 Consider the single replicate of the 2⁴ design in Example 3.11. Suppose 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?

Solution: Fit first-order with two-way interaction linear model of factors a and d .

```
library(rsm)
rsm.3.12.y.TWIabcd <- rsm(y ~ FO(a, b, c, d) + TWI(a, b, c, d), data = df.3.11)
# externally Studentized residuals
rsm.3.12.y.TWIabcd$studres <- rstudent(rsm.3.12.y.TWIabcd)
summary(rsm.3.12.y.TWIabcd)

##
## Call:
## rsm(formula = y ~ FO(a, b, c, d) + TWI(a, b, c, d), data = df.3.11)
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   772.94      11.94    64.72 1.7e-08 ***
## a              -47.69      11.94    -3.99 0.0104 *
## b               2.31      11.94     0.19 0.8541
## c               6.81      11.94     0.57 0.5931
## d             156.19      11.94    13.08 4.7e-05 ***
## a:b            -7.06      11.94    -0.59 0.5800
## a:c           -15.56      11.94    -1.30 0.2493
## a:d           -79.94      11.94    -6.69 0.0011 **
## b:c           -25.06      11.94    -2.10 0.0899 .
## b:d            -3.44      11.94    -0.29 0.7850
## c:d            -4.19      11.94    -0.35 0.7402
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.979, Adjusted R-squared:  0.938
## F-statistic: 23.9 on 10 and 5 DF,  p-value: 0.00133
##
## Analysis of Variance Table
##
## Response: y
##              Df Sum Sq Mean Sq F value  Pr(>F)
## FO(a, b, c, d)  4 427526  106882   46.83 0.00037
## TWI(a, b, c, d)  6 117433   19572    8.58 0.01614
## Residuals       5  11412    2282
## Lack of fit     5  11412    2282
## Pure error      0     0
##
## Stationary point of response surface:
##           a           b           c           d
## 2.0146 -0.9064 -0.4157 -0.4356
##
## Eigenanalysis:
## $values
## [1] 40.38 12.50 -10.85 -42.02
##
## $vectors
##           [,1]      [,2]      [,3]      [,4]
## a  0.709720  0.01862  0.1264  0.6928
## b -0.001283 -0.70266 -0.6961  0.1472
## c -0.100220  0.70662 -0.6695  0.2058
## d -0.697318 -0.08131  0.2262  0.6753
```

Plot diagnostics.

```
# plot diagnostics
par(mfrow=c(2,4))

plot(df.3.11$a, rsm.3.12.y.TWIabcd$studres, main="Residuals vs a")
# horizontal line at zero
abline(h = 0, col = "gray75")
```

```

plot(df.3.11$b, rsm.3.12.y.TWIabcd$studres, main="Residuals vs b")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(df.3.11$c, rsm.3.12.y.TWIabcd$studres, main="Residuals vs c")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(df.3.11$d, rsm.3.12.y.TWIabcd$studres, main="Residuals vs d")
# horizontal line at zero
abline(h = 0, col = "gray75")

## residuals vs order of data
# plot(rsm.3.12.y.TWIabcd$studres, main="Residuals vs Order of data")
# # horizontal line at zero
# abline(h = 0, col = "gray75")

plot(rsm.3.12.y.TWIabcd, which = c(1,4,6))

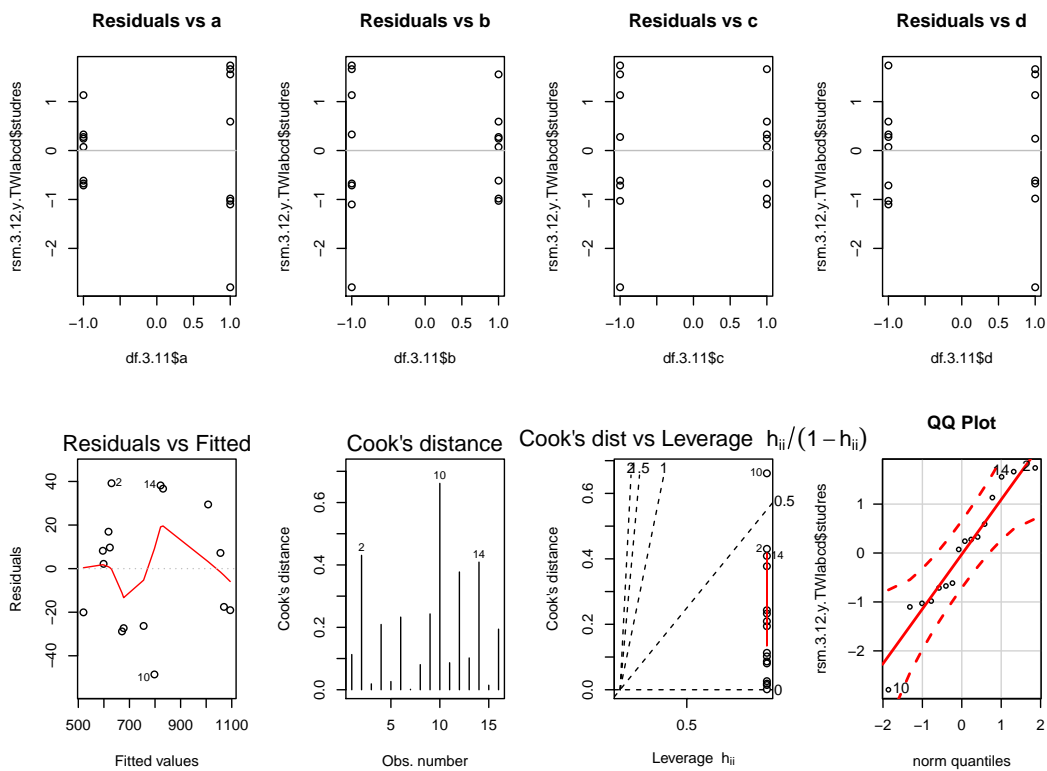
# Normality of Residuals
library(car)
qqPlot(rsm.3.12.y.TWIabcd$studres, las = 1, id.n = 3, main="QQ Plot")

## 10 2 14
## 1 16 15

cooks.distance(rsm.3.12.y.TWIabcd)

##      1      2      3      4      5      6      7      8      9     10
## 0.112867 0.430616 0.018797 0.209179 0.026316 0.232786 0.001342 0.080444 0.242995 0.661298
##      11     12     13     14     15     16
## 0.086490 0.377426 0.101896 0.408919 0.014486 0.194142

```



The ANOVA and parameter estimate table provides conclusions that are similar to 3.11 (*a*, *d*, and *ad*), both in variable selection and ANOVA validation. The Cook's D values aren't worrisome.

In general, I think it may be reckless to assume high-order interactions are ignorable. It takes very little

effort to test high-order interactions and so we should test them, even though unreplicated designs have very little power to find high-order interactions significant.

- (15^{pts}) **6. 3.20** Consider the experiment described in Exercise 3.1. Suppose that only one replicate (four runs) could be obtained in a single 4-hr time period, and the experimenters were concerned about unknown factors that could vary from one time period to another.

- (a) (5 pts) Set up a design in four blocks that will minimize the time effects.

Solution: Assuming that each replicate is performed at a moment of time, it makes sense to block each replicate to control for time effects of when the experiments were run.

- (b) (5 pts) Analyze the data assuming that the design had been run as in part (a). Compare the results of your analysis with the original analysis in Exercise 3.1.

Solution:

```
library(rsm)
rsm.3.20.y.TWIab <- rsm(y ~ block + FO(a, b) + TWI(a, b), data = df.3.1)
# externally Studentized residuals
rsm.3.20.y.TWIab$studres <- rstudent(rsm.3.20.y.TWIab)
summary(rsm.3.20.y.TWIab)

##
## Call:
## rsm(formula = y ~ block + FO(a, b) + TWI(a, b), data = df.3.1)
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25.575      0.872   29.34 3.0e-10 ***
## blocky2      -0.250      1.233   -0.20  0.844
## blocky3      -3.900      1.233   -3.16  0.011 *
## blocky4      -2.825      1.233   -2.29  0.048 *
## a             8.319      0.436   19.08 1.4e-08 ***
## b             3.769      0.436    8.65 1.2e-05 ***
## a:b           4.356      0.436    9.99 3.6e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.984, Adjusted R-squared:  0.973
## F-statistic: 92.2 on 6 and 9 DF,  p-value: 1.44e-07
##
## Analysis of Variance Table
##
## Response: y
##          Df Sum Sq Mean Sq F value Pr(>F)
## block     3    44      15    4.86  0.028
## FO(a, b)   2  1334     667  219.48 2.3e-08
## TWI(a, b)  1   304     304   99.88 3.6e-06
## Residuals  9     27      3
## Lack of fit 0      0     Inf
## Pure error  9     27      3
##
## Stationary point of response surface:
##      a      b
## -0.8651 -1.9096
##
## Eigenanalysis:
## $values
## [1]  2.178 -2.178
##
## $vectors
##      [,1] [,2]
## a 0.7071 -0.7071
## b 0.7071  0.7071
```

The block effect of replicate is significant. The mean between each rep is significant, therefore blocking will reduce the variability of the other parameter estimates. In particular, the interaction

is more significant.

- (c) (5 pts) Suppose that the experimenter's concerns were realized and that as a result of the time effect, the observations in replicate IV were (1) = 34.4, $a = 44.5$, $b = 34.2$, and $ab = 59.9$ instead of the values shown in Exercise 3.1. Reanalyze these new data, assuming that the blocked design from part (a) had been used. How has the time effect influenced your conclusions?

For 3.20 (c), one of the alternative observations for Rep IV is wrong. Indicate what it should be, then answer the question.

Solution: Correction: The a value of 44.4 should be 42.5 (adding 20 to each value in Rep IV).

```
df.3.20 <- df.3.1
df.3.20[(df.3.20$block == "y4"),"y"] <- df.3.20[(df.3.20$block == "y4"),"y"] + 20
library(rsm)
rsm.3.20c.y.TWIab <- rsm(y ~ block + FO(a, b) + TWI(a, b), data = df.3.20)
# externally Studentized residuals
rsm.3.20c.y.TWIab$studres <- rstudent(rsm.3.20c.y.TWIab)
summary(rsm.3.20c.y.TWIab)

##
## Call:
## rsm(formula = y ~ block + FO(a, b) + TWI(a, b), data = df.3.20)
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25.575      0.872   29.34  3.0e-10 ***
## blocky2      -0.250      1.233   -0.20   0.844
## blocky3      -3.900      1.233   -3.16   0.011 *
## blocky4      17.175      1.233   13.93  2.1e-07 ***
## a             8.319      0.436   19.08  1.4e-08 ***
## b             3.769      0.436    8.65  1.2e-05 ***
## a:b           4.356      0.436    9.99  3.6e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.99, Adjusted R-squared:  0.983
## F-statistic: 149 on 6 and 9 DF, p-value: 1.76e-08
##
## Analysis of Variance Table
##
## Response: y
##          Df Sum Sq Mean Sq F value Pr(>F)
## block     3   1071     357   117.5 1.6e-07
## FO(a, b)   2   1334     667   219.5 2.3e-08
## TWI(a, b)  1    304     304    99.9 3.6e-06
## Residuals  9      27      3
## Lack of fit 0      0     Inf
## Pure error  9      27      3
##
## Stationary point of response surface:
##      a      b
## -0.8651 -1.9096
##
## Eigenanalysis:
## $values
## [1]  2.178 -2.178
##
## $vectors
##      [,1] [,2]
## a 0.7071 -0.7071
## b 0.7071  0.7071
```

By blocking the time (rep) effect, which has increased compared to part (b), the results for the other parameters will be exactly the same as in part (b). This is confirmed in the ANOVA table above.

- (25^{pts}) **7. 3.26** Consider the situation described in Exercise 3.13. Set up a design in two blocks for 16 observations each of this problem. Analyze the data that result.

Solution: We confound blocks with the highest-order interaction, ABCDE. Therefore, if we analyze the data and find interaction ABCDE important, it will be unclear whether it is the ABCDE effect, the block effect, or both.

```
#### 3.26
# generate design by confounding blocks with highest-order interaction, ABCDE
design.3.26 <- cube( ~ a + b + c + d + e, blockgen ~ a * b * c * d * e, n0 = 0, randomize = FALSE)

# read data
fn.data <- "http://statacumen.com/teach/RSM/data/RSM_HW_03-26.txt"
df.3.26 <- read.table(fn.data, header=TRUE)

# join by design variables a, b, c, d, e
library(plyr)
df.3.26.block <- join(df.3.26, as.data.frame(design.3.26), by = c("a", "b", "c", "d", "e"), type = "full")
df.3.26.block$blockgen <- factor(df.3.26.block$blockgen)
str(df.3.26.block)

## 'data.frame': 32 obs. of 9 variables:
## $ a : num -1 1 -1 1 -1 1 -1 1 -1 1 ...
## $ b : num -1 -1 1 1 -1 -1 1 1 -1 -1 ...
## $ c : num -1 -1 -1 -1 1 1 1 1 -1 -1 ...
## $ d : num -1 -1 -1 -1 -1 -1 -1 -1 1 1 ...
## $ e : num -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
## $ y : int 15 20 70 112 30 40 79 125 16 21 ...
## $ run.order: int 1 2 3 4 5 6 7 8 9 10 ...
## $ std.order: int 1 2 3 4 5 6 7 8 9 10 ...
## $ blockgen : Factor w/ 2 levels "-1","1": 1 2 2 1 2 1 1 2 2 1 ...

# Fit first-order with four-way interaction linear model.
#lm.3.26.y.block.4WIabcd <- lm(y ~ (a + b + c + d + e)^5, data = df.3.26.block)
lm.3.26.y.block.4WIabcd <- lm(y ~ blockgen + (a + b + c + d)^4, data = df.3.26.block)
# externally Studentized residuals
lm.3.26.y.block.4WIabcd$studres <- rstudent(lm.3.26.y.block.4WIabcd)
summary(lm.3.26.y.block.4WIabcd)

##
## Call:
## lm(formula = y ~ blockgen + (a + b + c + d)^4, data = df.3.26.block)
##
## Residuals:
## Min 1Q Median 3Q Max
## -5.31 -1.69 0.00 1.69 5.31
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 61.3125 0.8753 70.05 < 2e-16 ***
## blockgen1 0.3750 1.2379 0.30 0.766
## a 12.3125 0.6189 19.89 3.4e-12 ***
## b 34.3125 0.6189 55.44 < 2e-16 ***
## c 9.3750 0.6189 15.15 1.7e-10 ***
## d -1.1875 0.6189 -1.92 0.074 .
## a:b 8.2500 0.6189 13.33 1.0e-09 ***
## a:c 0.5625 0.6189 0.91 0.378
## a:d -0.3750 0.6189 -0.61 0.554
## b:c -0.3125 0.6189 -0.50 0.621
## b:d -1.3750 0.6189 -2.22 0.042 *
## c:d 1.0625 0.6189 1.72 0.107
## a:b:c -0.2500 0.6189 -0.40 0.692
## a:b:d 0.0625 0.6189 0.10 0.921
## a:c:d -0.2500 0.6189 -0.40 0.692
## b:c:d 0.3750 0.6189 0.61 0.554
## a:b:c:d -0.0625 0.6189 -0.10 0.921
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

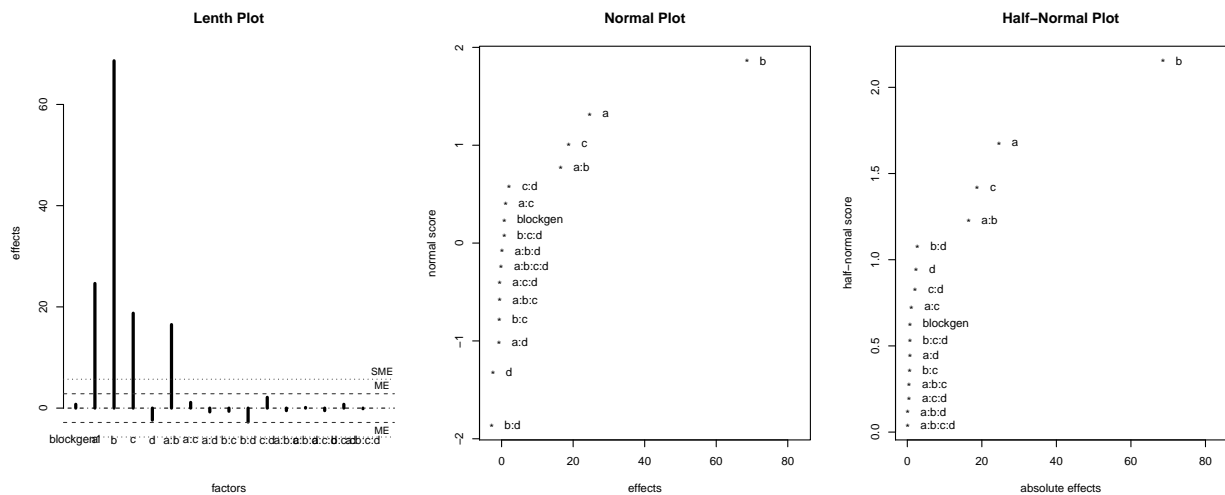
```
## Residual standard error: 3.5 on 15 degrees of freedom
## Multiple R-squared: 0.996, Adjusted R-squared: 0.992
## F-statistic: 243 on 16 and 15 DF, p-value: 3.53e-15
```

The normal plot and Lenth procedure below indicate factors a , b , c , and the ab interaction are important factors, while others are unlikely to be. This is consistent with the parameter estimate table above, which also includes the bd interaction as marginally significant.

```
# BsMD package has unreplicated factorial tests (Daniel plots (aka normal), and Lenth)
library(BsMD)
par(mfrow=c(1,3))
LenthPlot(lm.3.26.y.block.4WIabcd, alpha = 0.05, main = "Lenth Plot") # , adj = 0.2

## alpha PSE ME SME
## 0.050 1.125 2.838 5.711

DanielPlot(lm.3.26.y.block.4WIabcd, main = "Normal Plot")
DanielPlot(lm.3.26.y.block.4WIabcd, half = TRUE, main = "Half-Normal Plot")
```



We project the design into a 2^3 design in a , b , and c .
Fit first-order with two-way interaction linear model of factors a and d .

```
library(rsm)
rsm.3.26.y.block.TWIabc <- rsm(y ~ blockgen + FO(a, b, c) + TWI(a, b, c), data = df.3.26.block)
# externally Studentized residuals
rsm.3.26.y.block.TWIabc$studres <- rstudent(rsm.3.26.y.block.TWIabc)
summary(rsm.3.26.y.block.TWIabc)

##
## Call:
## rsm(formula = y ~ blockgen + FO(a, b, c) + TWI(a, b, c), data = df.3.26.block)
##
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  61.312     0.939   65.27 < 2e-16 ***
## blockgen1     0.375     1.329    0.28  0.78
## a             12.313     0.664   18.54 1.0e-15 ***
## b             34.313     0.664   51.66 < 2e-16 ***
## c              9.375     0.664   14.11 4.1e-13 ***
## a:b           8.250     0.664   12.42 6.1e-12 ***
## a:c           0.562     0.664    0.85  0.41
## b:c          -0.313     0.664   -0.47  0.64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared: 0.993, Adjusted R-squared: 0.991
## F-statistic: 481 on 7 and 24 DF, p-value: <2e-16
##
## Analysis of Variance Table
```

```
##
## Response: y
##          Df Sum Sq Mean Sq F value Pr(>F)
## blockgen  1     1         1    0.08  0.78
## F0(a, b, c) 3 45339 15113 1070.34 < 2e-16
## TWI(a, b, c) 3  2191    730   51.73 1.3e-10
## Residuals 24   339    14
## Lack of fit  1     2     2    0.14  0.72
## Pure error 23   337    15
##
## Stationary point of response surface:
##          a         b         c
## -10.83  10.51 -176.04
##
## Eigenanalysis:
## $values
## [1]  4.12690  0.02118 -4.14808
##
## $vectors
##          [,1]      [,2]      [,3]
## a -0.70751  0.03742  0.70571
## b -0.70637 -0.06779 -0.70459
## c -0.02147  0.99700 -0.07439
```

The regression confirms that the regression is significant, with an F statistic of 480.9 and a p-value of 0, and the main effects a , b , c and one interaction, ab , are all significant. The block effect is unimportant. In the figure below it appears that residuals do not have homoscedasticity when plotted against predicted values or factor b , having less variability for lower values of the response and lower setting of b . None of the Cook's D values appear large enough to investigate.

```
# plot diagnostics
par(mfrow=c(2,4))

plot(df.3.26.block$a, rsm.3.26.y.block.TWIabc$studres, main="Residuals vs a")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(df.3.26.block$b, rsm.3.26.y.block.TWIabc$studres, main="Residuals vs b")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(df.3.26.block$c, rsm.3.26.y.block.TWIabc$studres, main="Residuals vs c")
# horizontal line at zero
abline(h = 0, col = "gray75")

# residuals vs order of data
plot(rsm.3.26.y.block.TWIabc$studres, main="Residuals vs Order of data")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(rsm.3.26.y.block.TWIabc, which = c(1,4,6))

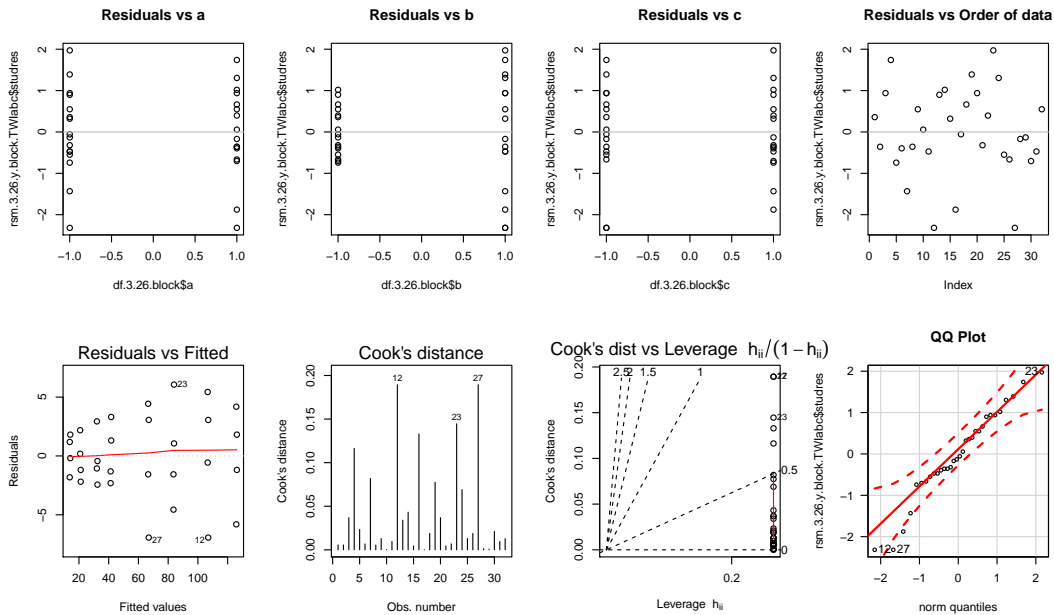
# Normality of Residuals
library(car)
qqPlot(rsm.3.26.y.block.TWIabc$studres, las = 1, id.n = 3, main="QQ Plot")

## 12 27 23
## 1 2 32

cooks.distance(rsm.3.26.y.block.TWIabc)

##          1          2          3          4          5          6          7          8          9
## 0.0055484 0.0055484 0.0369021 0.1163316 0.0233770 0.0067779 0.0819040 0.0055484 0.0129257
##          10         11         12         13         14         15         16         17         18
## 0.0001383 0.0096059 0.1893674 0.0339512 0.0431729 0.0044418 0.1329307 0.0001383 0.0188276
##          19         20         21         22         23         24         25         26         27
## 0.0774776 0.0369021 0.0044418 0.0067779 0.1446115 0.0689936 0.0129257 0.0188276 0.1893674
##          28         29         30         31         32
```

```
## 0.0012449 0.0007531 0.0210408 0.0096059 0.0129257
```



The interaction plots below indicates that when b is at its high level, the number of good chips per wafer increases more when a goes from its low to high level, than when b is at its low level.

```
# Interaction plots, ggplot
library(plyr)
# Calculate the cell means for each (a, b) combination
# create factor version for ggplot categories
df.3.26.factor <- df.3.26.block
df.3.26.factor$a <- factor(df.3.26.factor$a)
df.3.26.factor$b <- factor(df.3.26.factor$b)

#mean(df.3.26.factor[, "y"])
df.3.26.factor.mean <- ddply(df.3.26.factor, .(), summarise, m = mean(y))
#df.3.26.factor.mean
df.3.26.factor.mean.d <- ddply(df.3.26.factor, .(a), summarise, m = mean(y))
#df.3.26.factor.mean.d
df.3.26.factor.mean.i <- ddply(df.3.26.factor, .(b), summarise, m = mean(y))
#df.3.26.factor.mean.i
df.3.26.factor.mean.di <- ddply(df.3.26.factor, .(a,b), summarise, m = mean(y))
#df.3.26.factor.mean.di

p <- ggplot(df.3.26.factor, aes(x = a, y = y, colour = b, shape = b))
p <- p + geom_hline(aes(yintercept = 0), colour = "black",
  , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = df.3.26.factor.mean.di, aes(y = m), size = 4)
p <- p + geom_line(data = df.3.26.factor.mean.di, aes(y = m, group = b), size = 1.5)
p <- p + labs(title = "Beetles interaction plot, b by a")
print(p)

## ymaz not defined: adjusting position using y instead

p <- ggplot(df.3.26.factor, aes(x = b, y = y, colour = a, shape = a))
p <- p + geom_hline(aes(yintercept = 0), colour = "black",
  , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = df.3.26.factor.mean.di, aes(y = m), size = 4)
p <- p + geom_line(data = df.3.26.factor.mean.di, aes(y = m, group = a), size = 1.5)
p <- p + labs(title = "Beetles interaction plot, a by b")
print(p)

## ymaz not defined: adjusting position using y instead
```