Homework 9

John Kaspers

11/8/2020

##Problem 1 # part b

```
I=rep(1,2^3)
A=rep(c(-1,1),4)
B=rep(c(-1,-1,1,1),2)
C=c(-1,-1,-1,-1,1,1,1,1)
AB=A*B
AC=A*C
BC=B*C
ABC=A*BC
cbind(I=rep(1,8),A,B,AB,C,AC,BC,AB,ABC)
```

```
##      I  A  B AB  C AC BC AB ABC
## [1,] 1 -1 -1  1 -1  1  1  1  -1
## [2,] 1  1 -1 -1 -1 -1  1 -1   1
## [3,] 1 -1  1 -1 -1  1 -1 -1   1
## [4,] 1  1  1  1 -1 -1 -1  1  -1
## [5,] 1 -1 -1  1  1 -1 -1  1   1
## [6,] 1  1 -1 -1  1  1 -1 -1  -1
## [7,] 1 -1  1 -1  1 -1  1 -1  -1
## [8,] 1  1  1  1  1  1  1  1   1
```

```
#design details
k=3#number of factors
n=2#number of replicates

#response
yij=c(22,31,25,32,43,29,35,34,50,55,47,46,44,45,38,40,37,36,60,50,54,39,41,47)

#coded values for treatment levels
x1c=rep(c(rep(-1,3),rep(1,3)),4)
x2c=rep(c(rep(-1,6), rep(1,6)),2)
x3c=c(rep(-1,12), rep(1,12))

#`as.factor` treatment levels
x1f=as.factor(x1c)
x2f=as.factor(x2c)
x3f=as.factor(x3c)
mod.factor=aov(yij~x1f*x2f*x3f)
summary(mod.factor)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x1f          1    0.7     0.7   0.022 0.883680
## x2f          1  770.7   770.7  25.547 0.000117 ***
## x3f          1  280.2   280.2   9.287 0.007679 **
## x1f:x2f      1   16.7    16.7   0.552 0.468078
## x1f:x3f      1  468.2   468.2  15.519 0.001172 **
## x2f:x3f      1   48.2    48.2   1.597 0.224475
## x1f:x2f:x3f  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
```

```
#regression on coded levels
mod.reg=lm(yij~x1c+x2c+x3c+x1c:x2c+x1c:x3c+x2c:x3c+x1c:x2c:x3c)

#effects
l=2^k#l is the number of coefficients in the regression
effects=2*mod.reg$coeff[2:l]
effects
```

```
##         x1c         x2c         x3c     x1c:x2c     x1c:x3c     x2c:x3c
##   0.3333333  11.3333333   6.8333333  -1.6666667  -8.8333333  -2.8333333
## x1c:x2c:x3c
##  -2.1666667
```

Answer The effects for x2c, x3c, and x1c:x3c all appear to be large with x2c and x3c in a positive direction while x1c:x3c is in a negative direction.

#Part c

```
SS=(n*2^(k-2))*effects^2#sums of squares for effects
SST=sum((yij-mean(yij))^2)#Total sums of squares
percentSS=100*SS/SST#percentage contribution to total
cbind(effects, SS, percentSS)
```

```
##                 effects          SS   percentSS
## x1c           0.3333333   0.4444444  0.02121116
## x2c          11.3333333 513.7777778 24.52009757
## x3c           6.8333333 186.7777778  8.91398876
## x1c:x2c      -1.6666667  11.1111111  0.53027893
## x1c:x3c      -8.8333333 312.1111111 14.89553505
## x2c:x3c      -2.8333333  32.1111111  1.53250610
## x1c:x2c:x3c -2.1666667  18.7777778  0.89617139
```

Answer So yes, the results do confirm what was observed in part (b) as x2c, x3c and x1c:x3c all contribute a large portion to percentSS.

Part d

```
m1 <- lm(yij ~ x2c + x3c + x1c:x3c); m1
```

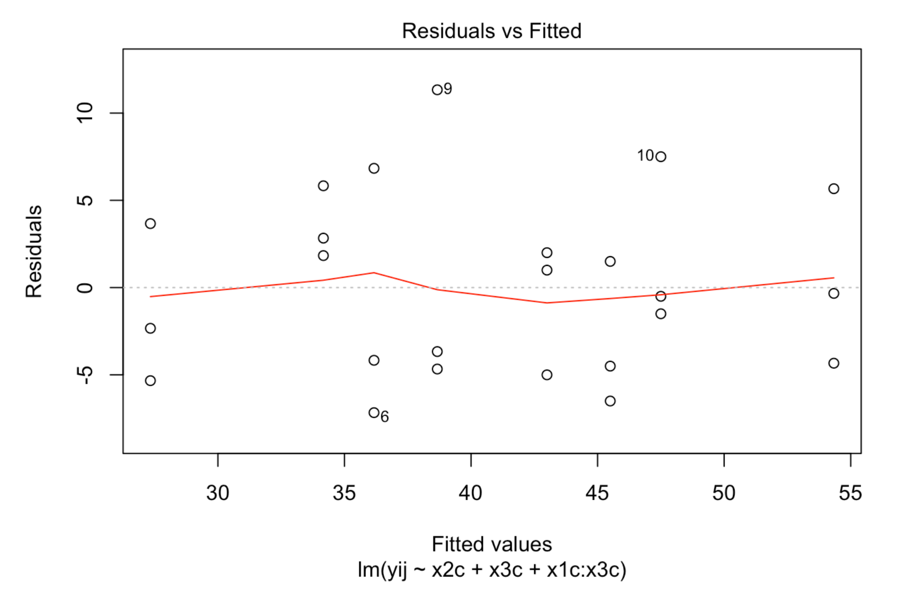
```
##
## Call:
## lm(formula = yij ~ x2c + x3c + x1c:x3c)
##
## Coefficients:
## (Intercept)          x2c          x3c      x3c:x1c
##      40.833        5.667        3.417       -4.417
```

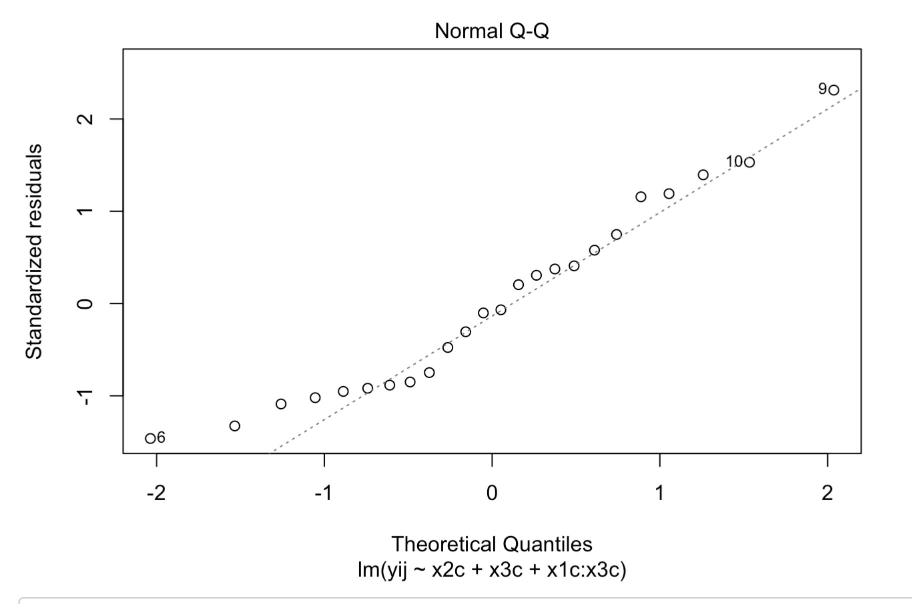
$y = 40.833 + 5.667\beta_{x2} + 3.417\beta_{x3} - 4.417\beta_{x1,x3}$

---

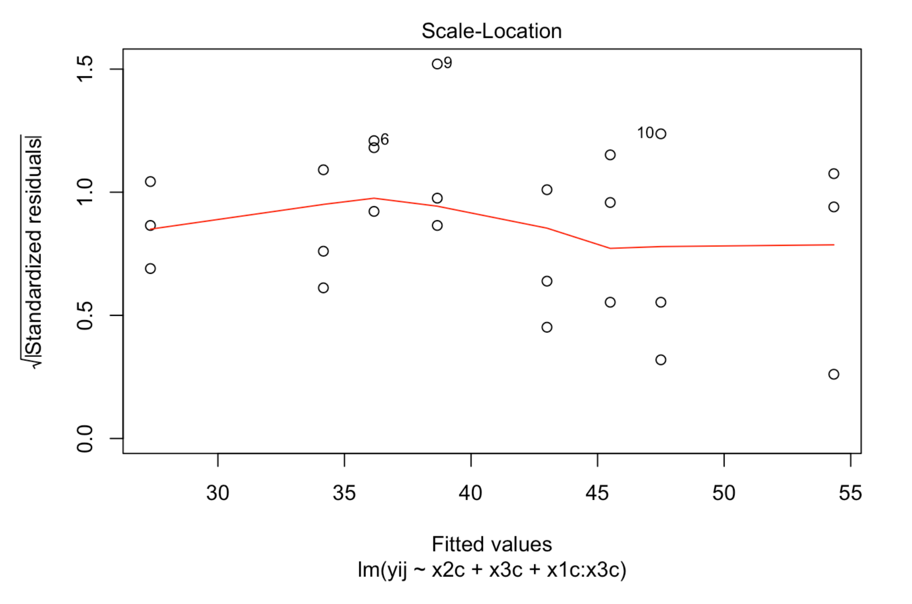
part e

```
plot(m1)
```





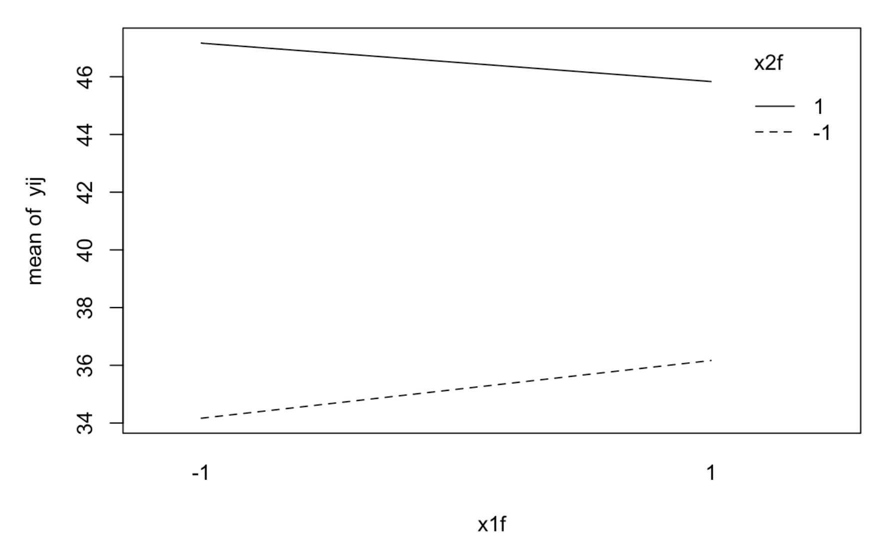

```
## hat values (leverages) are all = 0.1666667
##  and there are no factor predictors; no plot no. 5
```



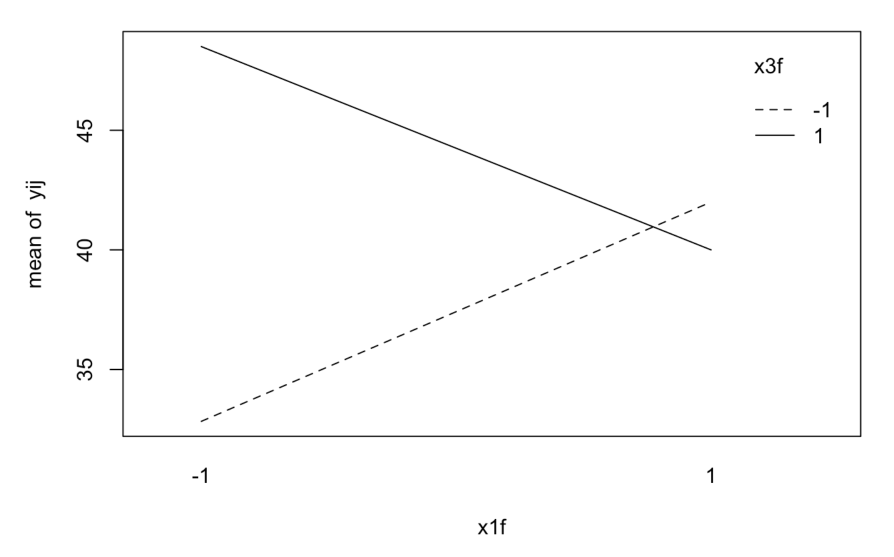
Answer Looking at the residual vs fitted plot, there appears to be a relatively constant variance. Looking at the QQ-plot, the relationship appears fairly linear. That said, leverage points cannot be plotted since there are no factor predictors. There are a few potential outliers in the residual plot and normal QQ-line plot.

Part f

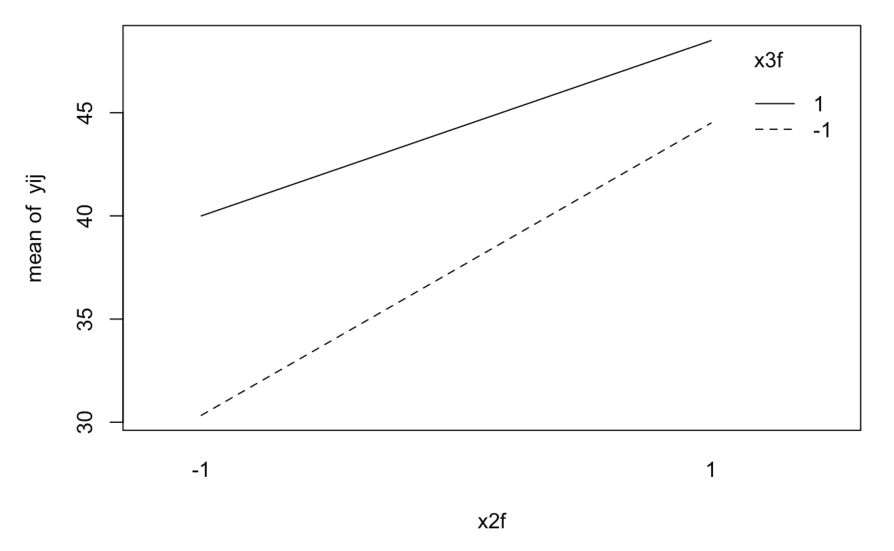
```
interaction.plot(x1f, x2f, yij)
```




```
interaction.plot(x1f, x3f, yij)
```




```
interaction.plot(x2f, x3f, yij)
```



Answer I would recommend a high tool geometry, low cutting angle, and low cutting speed.

##Problem 2

```
#se.effect
se.effect=summary(m1)$sigma/sqrt(n*2^(k-2))
moe=qt(0.975,m1$df.residual)*se.effect
cbind(effects, effects-moe, effects+moe)#confidence intervals
```

```
##                effects
## x1c          0.3333333  -5.265513  5.932180
## x2c         11.3333333   5.734487 16.932180
## x3c          6.8333333   1.234487 12.432180
## x1c:x2c     -1.6666667  -7.265513  3.932180
## x1c:x3c     -8.8333333 -14.432180 -3.234487
## x2c:x3c     -2.8333333  -8.432180  2.765513
## x1c:x2c:x3c -2.1666667  -7.765513  3.432180
```

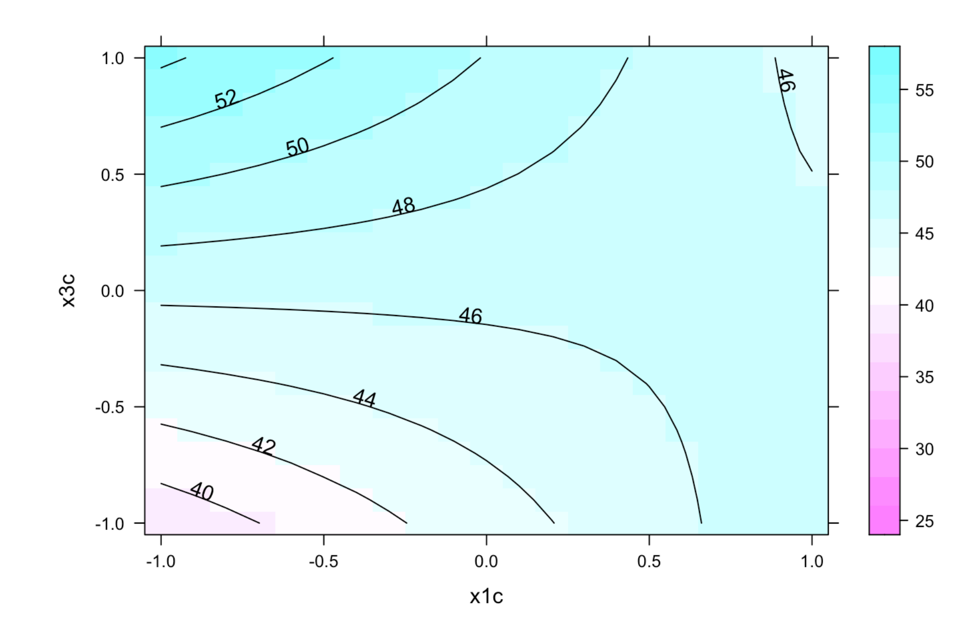
Answer So the intervals in question are x2c, x3c, and x1c:x3c interaction (these intervals do not contain zero), which are all of the ones we used in the m1 model. These intervals are significant.

Problem 3

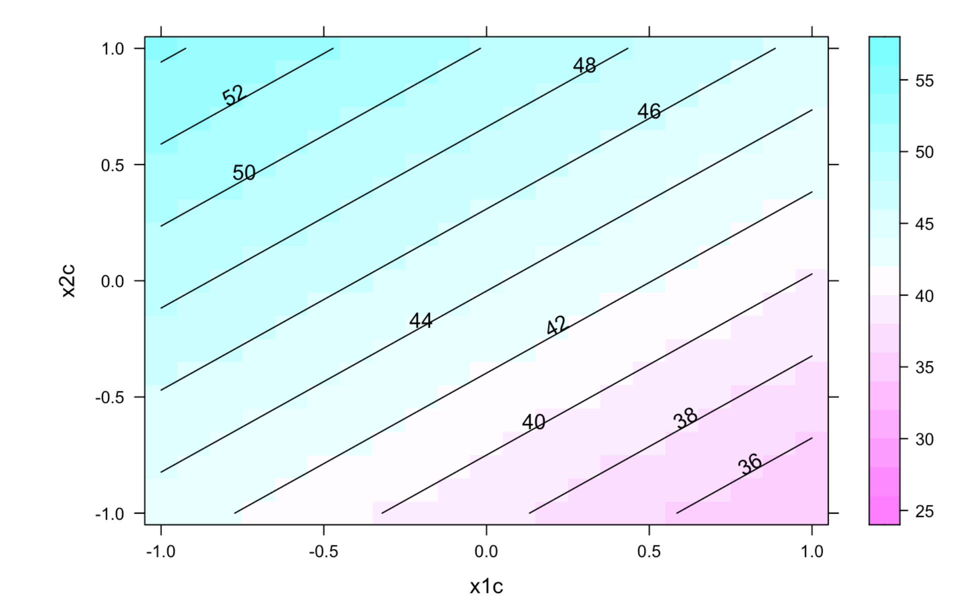
```
library(lattice)
```

```
## Warning: package 'lattice' was built under R version 3.6.2
```

```
x1.new=(rep(seq(-1,1,by=0.1),5))#create a sequence within range of A
x2.new=sort((rep(seq(-1,1,by=0.1),5)))#create a sequence within range of B
x3.new=sort((rep(seq(-1,1,by=0.1),5)))#create a sequence within range of B
tmp=list(x1c=x1.new,x2c=x2.new,x3c=x3.new)
new=expand.grid(tmp)#create grid
fit=c(predict(m1,new))#predict response for grid
contourplot(fit~x1c*x3c,data=new,cuts=16,region=T)
```




```
contourplot(fit~x1c*x2c,data=new,cuts=16,region=T)
```




```
contourplot(fit~x2c*x3c,data=new,cuts=16,region=T)
```