Homework 5

Ishita Dutta

5/14/2021

4.1

When joint confidence intervals for B0 and B1 are developed by the Bonferroni method with a family confidence coefficient of 90 percent, does this imply that 10 percent of the time the confidence interval for B0 will be incorrect? That 5 percent of the time the confidence interval for B0 will be incorrect and 5 percent of the time that for B1 will be incorrect? Discuss.

A coefficient of 90% implies that 10% of the time we will have either B0 or B1 incorrect. We are not sure whether the split will be 5 - 5 in error for B0 and B1 though.

4.3

Refer to Copier maintenance Problem 1.20

```
copierdata <- read.table("copier+maintenance.txt")
#columns are time and number
```

4.3a)

Will b0 and b1 tend to err in the same direction or in opposite directions here? Explain.

```
mean(copierdata[,2])
```

```
## [1] 5.111111
```

```
sigma.sq<- -mean(copierdata[,2])*var(copierdata[,1],copierdata[,2])
sigma.sq
```

```
## [1] -594.5926
```

b0 and b1 tend to err in opposite directions, because xbar = 5.111, a positive number, thus the covariance is a negative number which suggests that b0 and b1 will err in opposite directions.

4.3b)

Obtain Bonferroni joint confidence intervals for B0 and B1, using a 95 percent family confidence coefficient.

```
bonf.copier.lm <- lm(copierdata[,1]~copierdata[,2])
bonf.copier.lm
```

```
##
## Call:
## lm(formula = copierdata[, 1] ~ copierdata[, 2])
##
## Coefficients:
##     (Intercept)  copierdata[, 2]
##         -0.5802          15.0352
```

```
bonf.int <- confint(bonf.copier.lm, level = 1-(0.05/2))
bonf.int
```

```
##                     1.25 %   98.75 %
## (Intercept)     -7.092642  5.932329
## copierdata[, 2] 13.913221 16.157275
```

b0 = -0.5802 b1 = 15.0352 -7.093 <= B0 <= 5.932 13.913 <= B1 <= 16.157

4.3c)

A consultant has suggested that B0 should be 0 and B1 should equal 14.0. Do your joint confidence intervals in part (b) support this view? Yes, a B0 of 0 and B1 of 14 are both within the joint confidence intervals found in part (b).

4.10

Refer to Muscle mass Problem 1.27.

```
muscledata <- read.table("muscle+mass.txt")
#columns are muscle mass and age
```

4.10a)

The nutritionist is particularly interested in the mean muscle mass for women aged 45, 55, and 65. Obtain joint confidence intervals for the means of interest using the Working Hotelling procedure and a 95 percent family confidence coefficient.

```
mass= muscledata$V1
age = muscledata$V2
muscleLength = length(mass)
fitmuscle = lm(mass ~ age)
mse = summary(fitmuscle)$sigma^2
b0 = fitmuscle$coefficients[1]
b1 = fitmuscle$coefficients[2]
yhat45 = b0 + b1*45
yhat55 = b0 + b1*55
```

```
yhat65 = b0 + b1*65

se.yhat45 = sqrt(mse*(1/muscleLength + (45 - mean(age))^2/sum((age - mean(age))^2)))
se.yhat55 = sqrt(mse*(1/muscleLength + (55 - mean(age))^2/sum((age - mean(age))^2)))
se.yhat65 = sqrt(mse*(1/muscleLength + (65 - mean(age))^2/sum((age - mean(age))^2)))

fstat = qf(p = 0.95, df1 = 2, df2 = muscleLength - 2)
W = sqrt(fstat * 2)

wh.upper45 <- yhat45 + W * se.yhat45
wh.lower45 <- yhat45 - W * se.yhat45

wh.upper55 <- yhat55 + W * se.yhat55
wh.lower55 <- yhat55 - W * se.yhat55

wh.upper65 <- yhat65 + W * se.yhat65
wh.lower65 <- yhat65 - W * se.yhat65
cat("Age 45 Interval: [", wh.lower45,",",wh.upper45,"]\n")
```

```
## Age 45 Interval: [ 98.48916 , 107.1044 ]
```

```
cat("Age 55 Interval: [", wh.lower55,",",wh.upper55,"]\n")
```

```
## Age 55 Interval: [ 88.0154 , 93.77822 ]
```

```
cat("Age 65 Interval: [", wh.lower65,",",wh.upper65,"]\n")
```

```
## Age 65 Interval: [ 76.11248 , 81.88123 ]
```

4.10b)

Is the Working Hotelling procedure the most efficient one to be employed in part (a)? Explain.

```
B <- 1-qt(.95/(2 * 3), muscleLength - 1)
B < W
```

```
## [1] TRUE
```

Because B is greater than W, we know that the interval from the Bonferroni is a smaller interval, therefore that would be a bit better procedure than the one in part(a).

4.10c)

Three additional women aged 48, 59, and 74 have contacted the nutritionist. Predict the muscle mass for each of these three women using the Bonferroni procedure and a 95 percent family confidence coefficient.

```r
yhat48 = b0 + b1*48
yhat59 = b0 + b1*59
yhat74 = b0 + b1*74
pse.yhat_48 = sqrt(mse*(1+1/muscleLength+ (48 - mean(age))^2/sum((age - mean(age))^2)))
pse.yhat_59 = sqrt(mse*(1+1/muscleLength+ (59 - mean(age))^2/sum((age - mean(age))^2)))
pse.yhat_74 = sqrt(mse*(1+1/muscleLength+ (74 - mean(age))^2/sum((age - mean(age))^2)))



bh.upper48 <- yhat48 + B * pse.yhat_48
bh.lower48 <- yhat48 - B * pse.yhat_48

bh.upper59 <- yhat59 + B * pse.yhat_59
bh.lower59 <- yhat59 - B * pse.yhat_59

bh.upper74 <- yhat74 + B * pse.yhat_74
bh.lower74 <- yhat74 - B * pse.yhat_74
cat("Age 45 Interval: [", bh.lower48,",",bh.upper48,"]\n")
```

```
## Age 45 Interval: [ 82.52132 , 115.9322 ]
```

```r
cat("Age 55 Interval: [", bh.lower59,",",bh.upper59,"]\n")
```

```
## Age 55 Interval: [ 69.57227 , 102.7014 ]
```

```r
cat("Age 65 Interval: [", bh.lower74,",",bh.upper74,"]\n")
```

```
## Age 65 Interval: [ 51.52951 , 85.04428 ]
```

4.10d)

Subsequently, the nutritionist wishes to predict the muscle mass for a fourth woman aged-64, with a family confidence coefficient of 95 percent for the four predictions. Will the three prediction intervals in part (c) have to be recalculated? Would this also be true if the Scheffe procedure had been used in constructing the prediction intervals? Yes, we would have to calculate again using 64 now as our x value in each of the three prediction intervals and the Scheffe procedure.

6.1

Set up the X matrix and B vector for each of the following regression models (assume i == 1,…,4): Make sure to switch reading rows and columns when reading the matrix, apparently R does not want to write it down the correct way… The vector is fine though

6.1a)

___Yi = B0 + B1*Xi1 + B2*Xi1*Xi2 + Ei___

```
cat("X = \n")
```

```
## X =
```

```
matrix(c(1, 'X11', 'X11X12', 1, 'X21', 'X21X22', 1, 'X31', 'X31X32', 1, 'X41', 'X41X42'), nrow = 3, ncol
```

```
##      [,1]     [,2]     [,3]     [,4]
## [1,] "1"      "1"      "1"      "1"
## [2,] "X11"    "X21"    "X31"    "X41"
## [3,] "X11X12" "X21X22" "X31X32" "X41X42"
```

```
cat("\n\nB = \n")
```

```
##
##
## B =
```

```
matrix(c('B0', 'B1','B2'), nrow = 3, ncol = 1)
```

```
##      [,1]
## [1,] "B0"
## [2,] "B1"
## [3,] "B2"
```

6.1b)

___logYi = B0 + B1*Xi1 + B2*Xi1*Xi2 + Ei___

```
cat("X = \n")
```

```
## X =
```

```
matrix(c(1, 'X11', 'X12', 1, 'X21', 'X22', 1, 'X31', 'X32', 1, 'X41', 'X42'), nrow = 3, ncol = 4)
```

```
##      [,1]  [,2]  [,3]  [,4]
## [1,] "1"   "1"   "1"   "1"
## [2,] "X11" "X21" "X31" "X41"
## [3,] "X12" "X22" "X32" "X42"
```

```
cat("\n\nB = \n")
```

```
##
##
## B =
```

```
matrix(c('B0', 'B1','B2'), nrow = 3, ncol = 1)
```

```
##      [,1]
## [1,] "B0"
## [2,] "B1"
## [3,] "B2"
```

6.2

Set up the X matrix and B vector for each of the following regression models (assume i == 1,. . . ,5): Make sure to switch reading rows and columns when reading the matrix, apparently R does not want to write it down the correct way. . . The vector is fine though

6.2a)

___Yi = B1*Xi1 + B2*Xi2 + B3*(Xi1ˆ2) + Ei___

```
cat("X = \n")
```

```
## X =
```

```
matrix(c(’X11’, ’X12’, ’X11^2’, ’X21’, ’X22’, ’X21^2’, ’X31’, ’X32’, ’X31^2’, ’X41’, ’X42’,’X41^2’, ’X5
```

```
##      [,1]    [,2]    [,3]    [,4]    [,5]   
## [1,] "X11"   "X21"   "X31"   "X41"   "X51"  
## [2,] "X12"   "X22"   "X32"   "X42"   "X52"  
## [3,] "X11^2" "X21^2" "X31^2" "X41^2" "X51^2"
```

```
cat("\n\nB = \n")
```

```
## 
## 
## B =
```

```
matrix(c(’B1’, ’B2’,’B3’), nrow = 3, ncol = 1)
```

```
##      [,1]
## [1,] "B1"
## [2,] "B2"
## [3,] "B3"
```

6.2b)

sqrt(Yi) B0 + B1*Xi1 + B2*log10(Xi2) + Ei

```
cat("X = \n")
```

```
## X =
```

```
matrix(c(1, ’X11’, ’log10X12’, 1, ’X21’, ’log10X22’, 1, ’X31’, ’log10X32’, 1, ’X41’, ’log10X42’, 1, ’X5
```

```
##      [,1]       [,2]       [,3]       [,4]       [,5]      
## [1,] "1"        "1"        "1"        "1"        "1"       
## [2,] "X11"      "X21"      "X31"      "X41"      "X51"     
## [3,] "log10X12" "log10X22" "log10X32" "log10X42" "log10X52"
```

```
cat("\n\nB = \n")
```

```
##
##
## B =
```

```
matrix(c(’B0’, ’B1’,’B2’), nrow = 3, ncol = 1)
```

```
##      [,1]
## [1,] "B0"
## [2,] "B1"
## [3,] "B2"
```

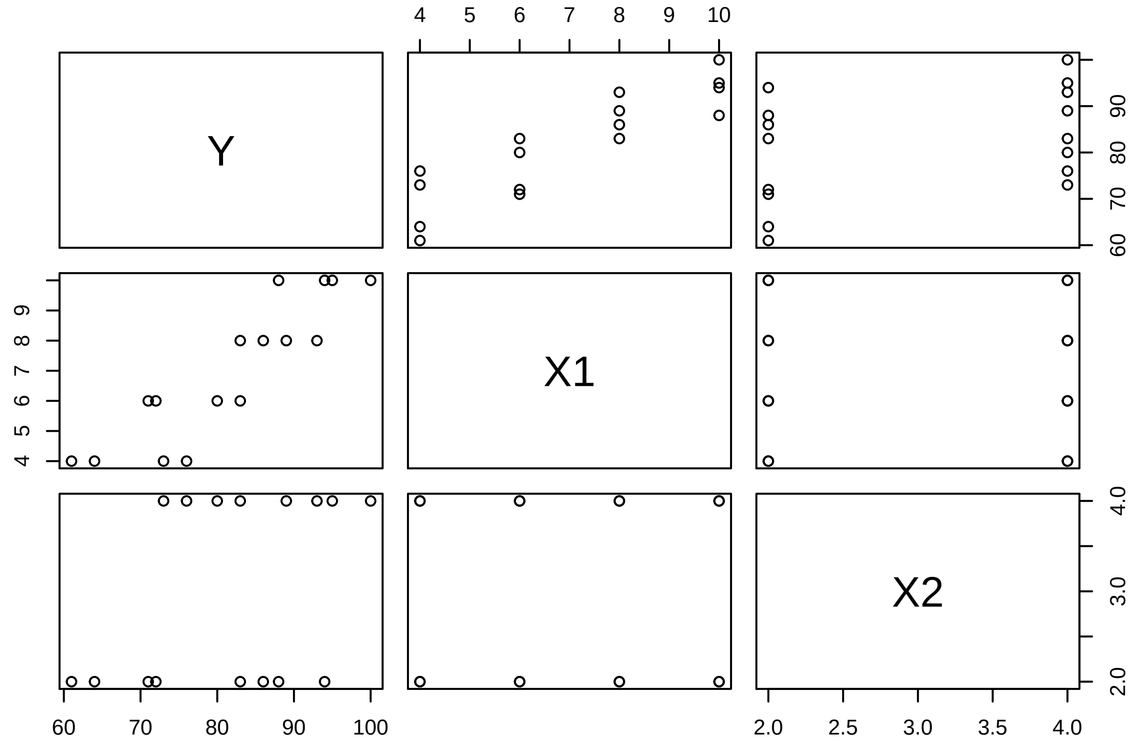
6.5

Brand preference. In a small scale experimental study of the relation between degree of brand liking (Y) and moisture content (X1) and sweetness (X2) of the product, the following results were obtained from the experiment based on a completely randomized design

6.5a)

Obtain the scatter plot matrix and the correlation matrix. What information do these diagnostic aids provide here?

```
brands = read.table("brand+preference.txt")
Y = brands[,1]
X1 = brands[,2]
X2 = brands[,3]
colnames(brands) = c("Y", "X1", "X2")
pairs(brands)
```



```
cor(brands)
```

```
##            Y        X1        X2
## Y  1.0000000 0.8923929 0.3945807
## X1 0.8923929 1.0000000 0.0000000
## X2 0.3945807 0.0000000 1.0000000
```

From the coefficients, we can conclude that Y is positively correlated to X1 and X2. We also know the correlation between Y and X1 is stronger than that of Y and X2, as well as the fact that there is no correlation betwen X1 and X2.

6.5b)

Fit regression model (6.1) to the data. State the estimated regression function. How is b1 interpreted here?

```
fit = lm(Y ~ X1+X2)
fit$coefficients
```

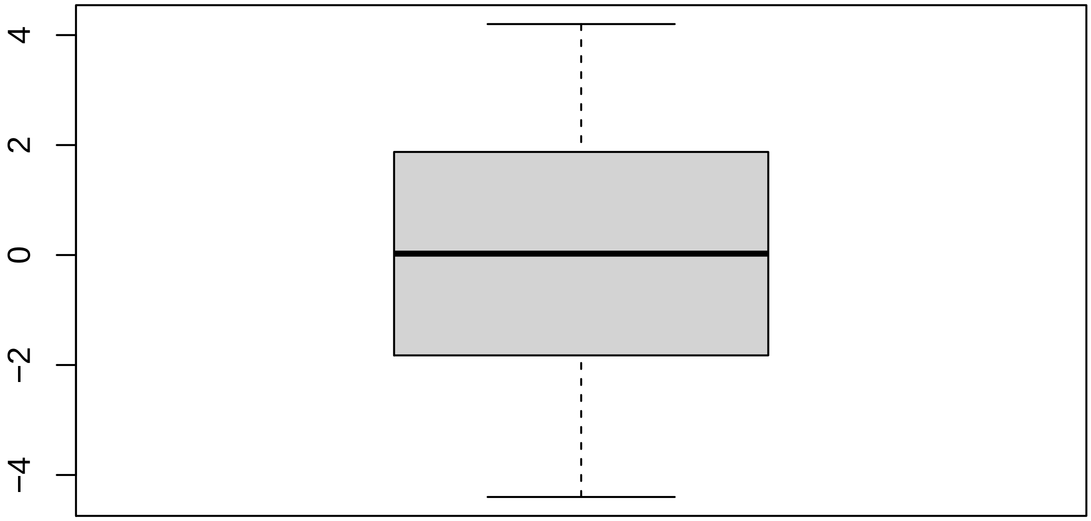
```
## (Intercept)          X1          X2
##      37.650       4.425       4.375
```

b1 shows the change in Y per unit increase/decrease in X1 or moisture content.

6.5c)

Obtain the residuals and prepare a box plot of the residuals. What information does this plot provide?

```
res = fit$residuals
boxplot(fit$residuals)
```

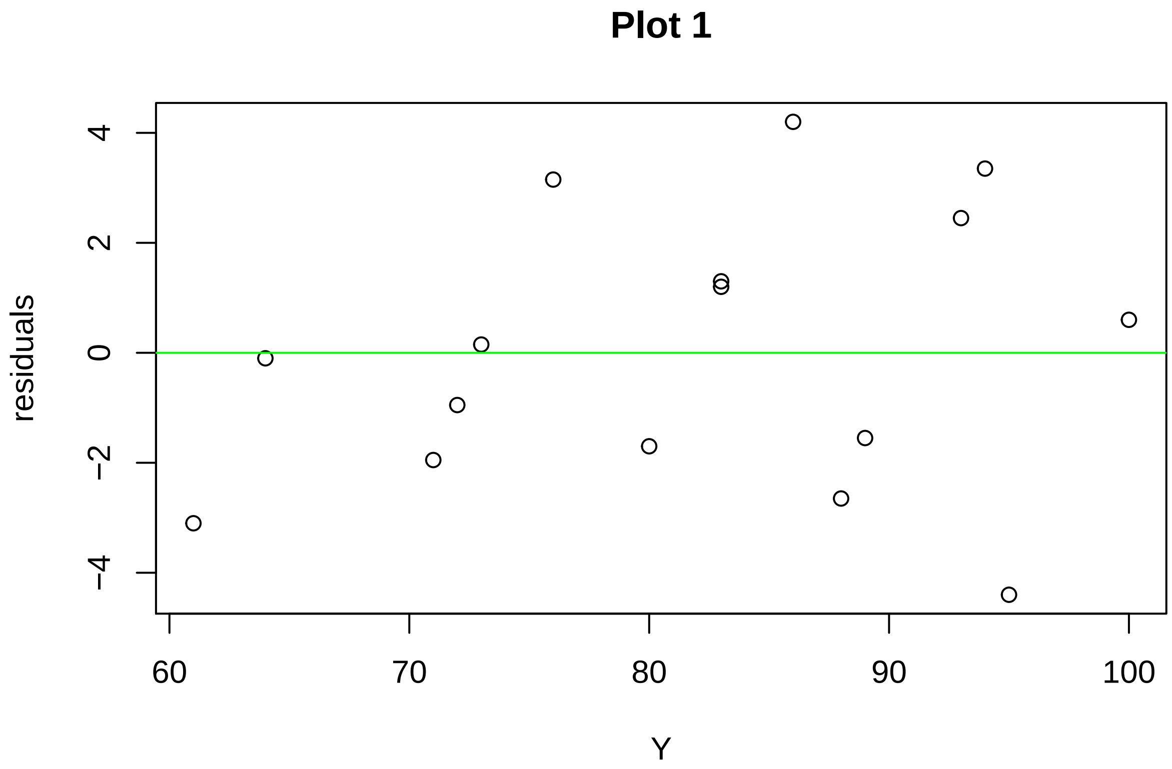


The plot shows there are no outlier residuals, an even distribution of residuals, and a center of 0. We can conclude the regression model fits the data well.

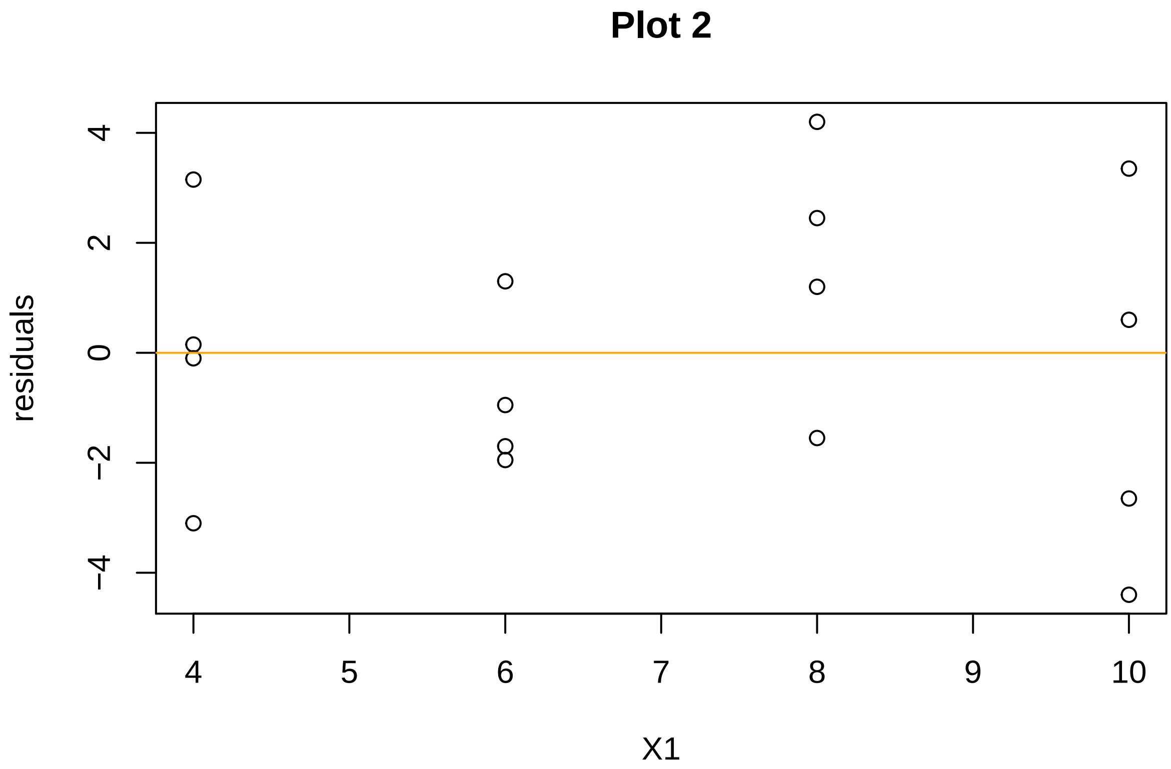
6.5d)

Plot the residuals against Y, XI, X2 , and XI * X2 on separate graphs. Also prepare a normal probability plot. Interpret the plots and summarize your findings.

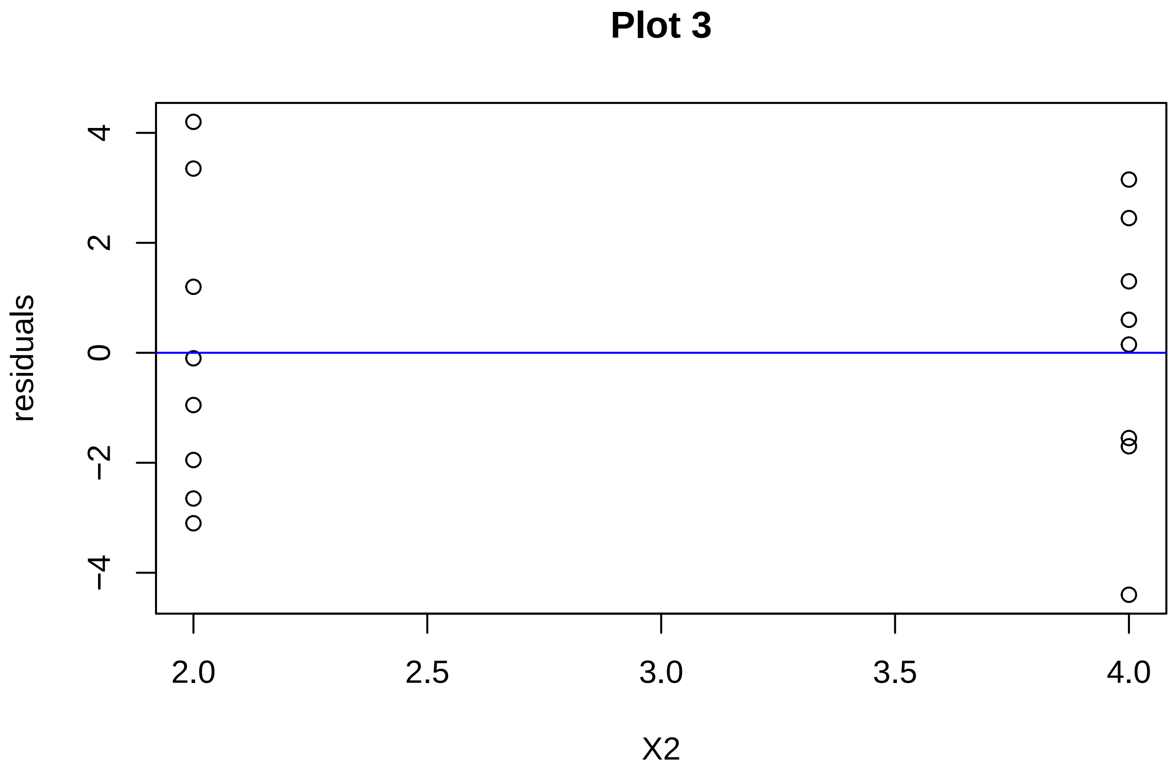
```
plot(Y, y=res, xlab='Y', ylab='residuals', main = "Plot 1")
abline(h=0, col='green')
```



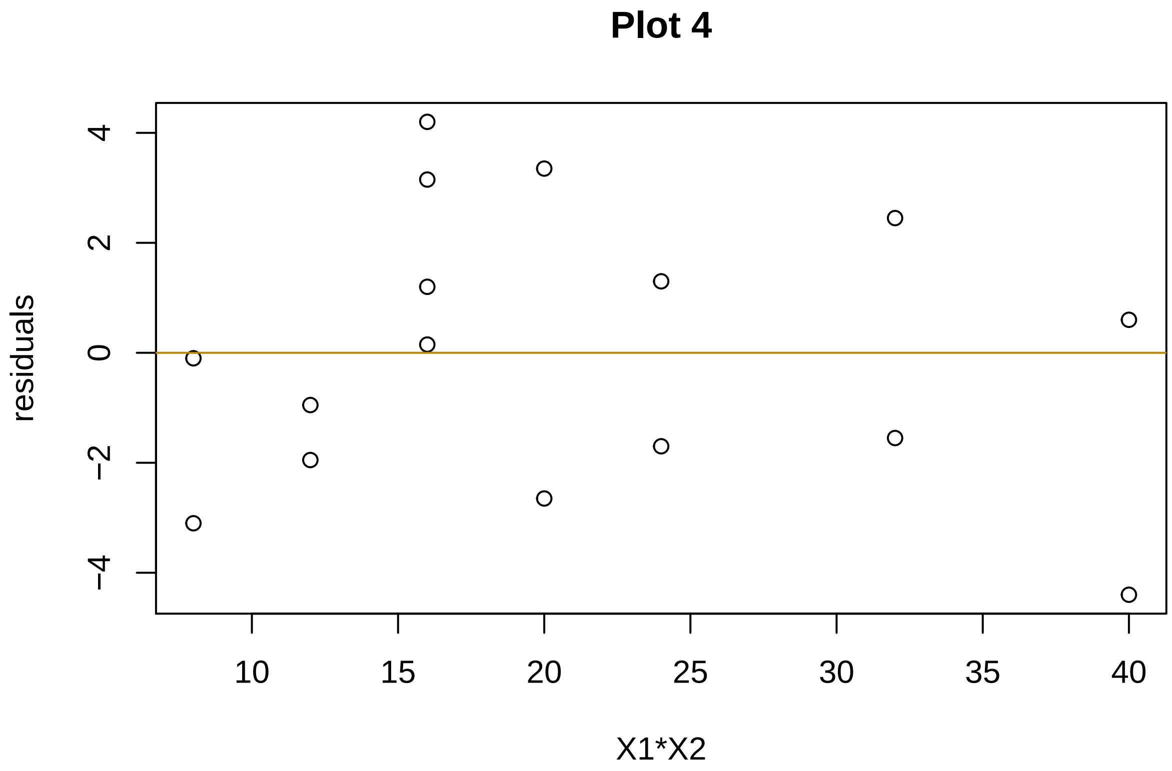
```r
plot(X1, y=res, xlab='X1', ylab='residuals', main = "Plot 2")
abline(h=0, col='orange')
```



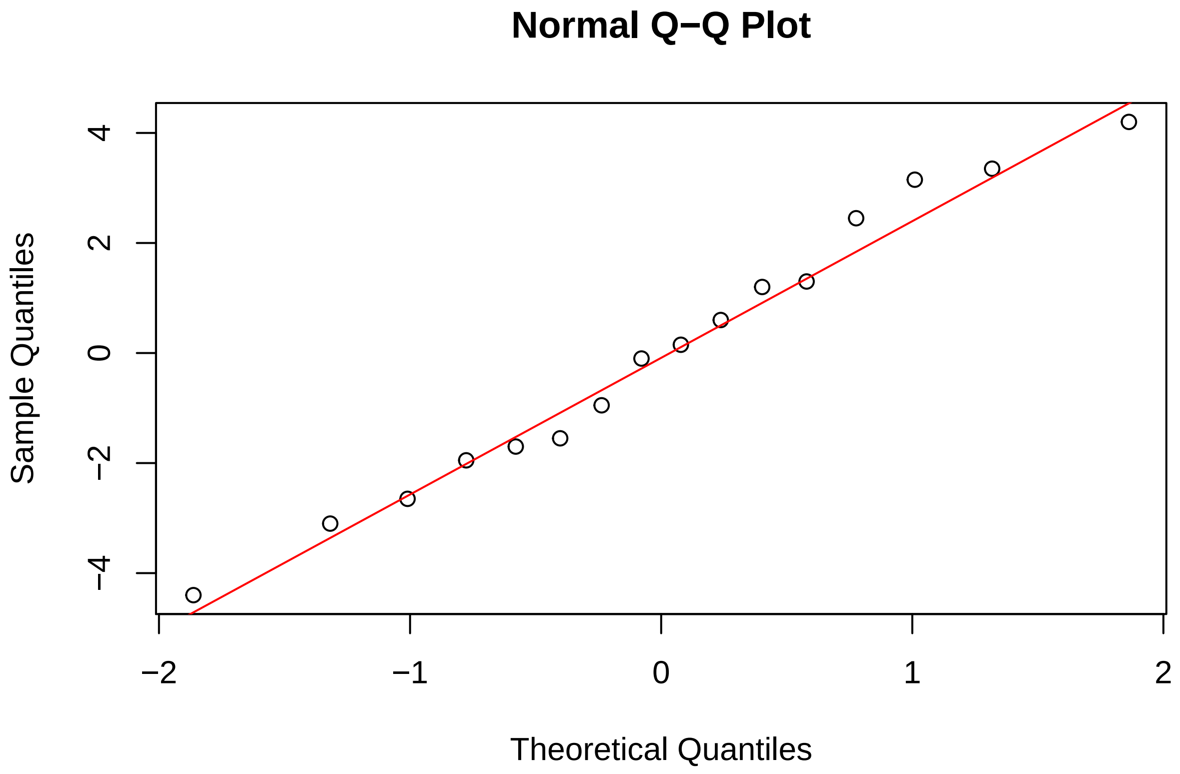
```r
plot(X2, y=res, xlab='X2', ylab='residuals', main = "Plot 3")
abline(h=0, col='blue')
```



```
plot(X1*X2, y=res, xlab='X1*X2', ylab='residuals', main = "Plot 4")
abline(h=0, col='darkgoldenrod')
```



```
qqnorm(res)
qqline(res, col='red')
```



Plots 1 and 4 are showing a variance in the data in that there are multiple different values for Y or X1 * X2, as opposed to plots 2 and 3, which only show that there are some values, like plot 2 has 4 distinct values for X1 and plot 3 has 2 distinct values for X3. From this, we can see that plots 1 and 4, rather what variables they depict might be more connected to each other than any of the plots are to plot 1. The normal probability plot tells us that the data is very normally distributed as the points are relatively on/very near to the line

6.5f)

Conduct a formal test for lack of fit of the first-order regression function; use alpha = .01. State the alternatives, decision rule, and conclusion. Ho –> E(Y) = B0 + B1X1 + B2X2 Ha –> E(Y) != B0 + B1X1 + B2X2 Decision –> If p-value is less than alpha, conclude Ha

```
fit.full = lm(Y~0 + as.factor(X1):as.factor(X2))
kable(anova(fit, fit.full))
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
13	94.3	NA	NA	NA	NA
8	57.0	5	37.3	1.047017	0.4530065

Conclusion: p-value of 0.45 > alpha of 0.01. Fail to reject Ha.