STAT5044_HW4

zhengzhi lin

2019.10.30

Problem 1

(a) The estimated function is: $\mathrm{Y} = 4.15 + 7.87 * \mathrm{X}_1 - 1.32 * X_2 + 6.24 * X_3$. The coefficients, b1 represents the difference in the predicted value of Y for each one-unit difference in X1, if X2,X3 remains constant. b2 represents the difference in the predicted value of Y for each one-unit difference in X2, if X1,X3 remains constant. b3 represents the difference in the predicted value of Y for each one-unit difference in X3, if X2,X1 remains constant.

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(knitr)
p1 <- read.table("datahw4.txt")
p1 <- p1 %>% as.data.frame() %>% rename( y = V1,
                                        x1 = V2,
                                        x2 = V3,
                                        x3 = V4) %>%
  mutate_if(is.factor, as.character) %>%
  mutate_if(is.character,as.numeric)
```

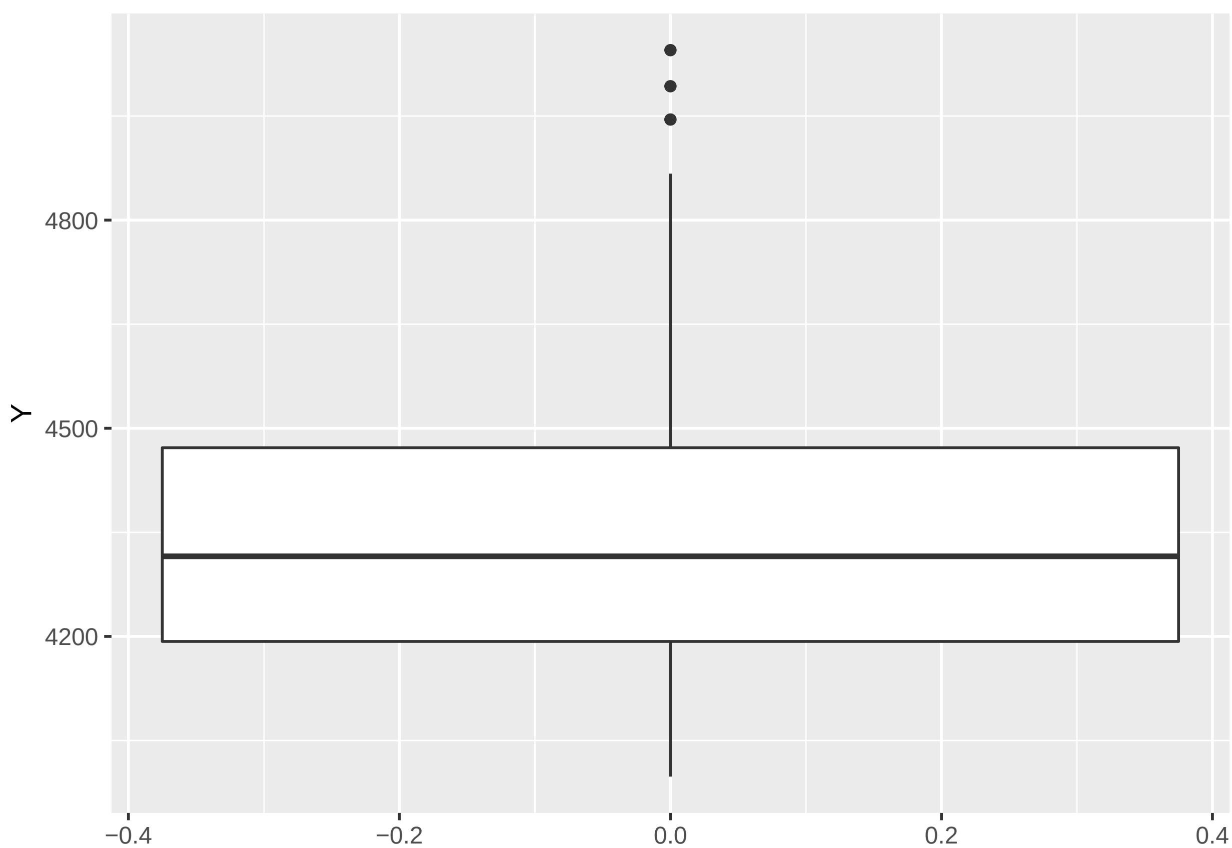
```
## Warning:      NA

## Warning:      NA

## Warning:      NA

## Warning:      NA
```

```
p1 <- p1[-1,]
X <- cbind(rep(1,nrow(p1)),p1[,2:4])
Y <- p1[,1]
X <- as.matrix(X)
Y <- as.matrix(Y)
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% Y    #estimation of beta
Y_hat <- X %*% beta_hat
beta_hat
```

```
##                              [,1]
## rep(1, nrow(p1))  4.149887e+03
## x1                7.870804e-04
## x2               -1.316602e+01
## x3                6.235545e+02
```

(b) The plot shows that the residual is randomly distributed and mean is close to zero. Therefore the error is random.

```
#residuals
residual <- Y - Y_hat
ggplot(data = as.data.frame(residual),aes(y=Y)) + geom_boxplot()
```



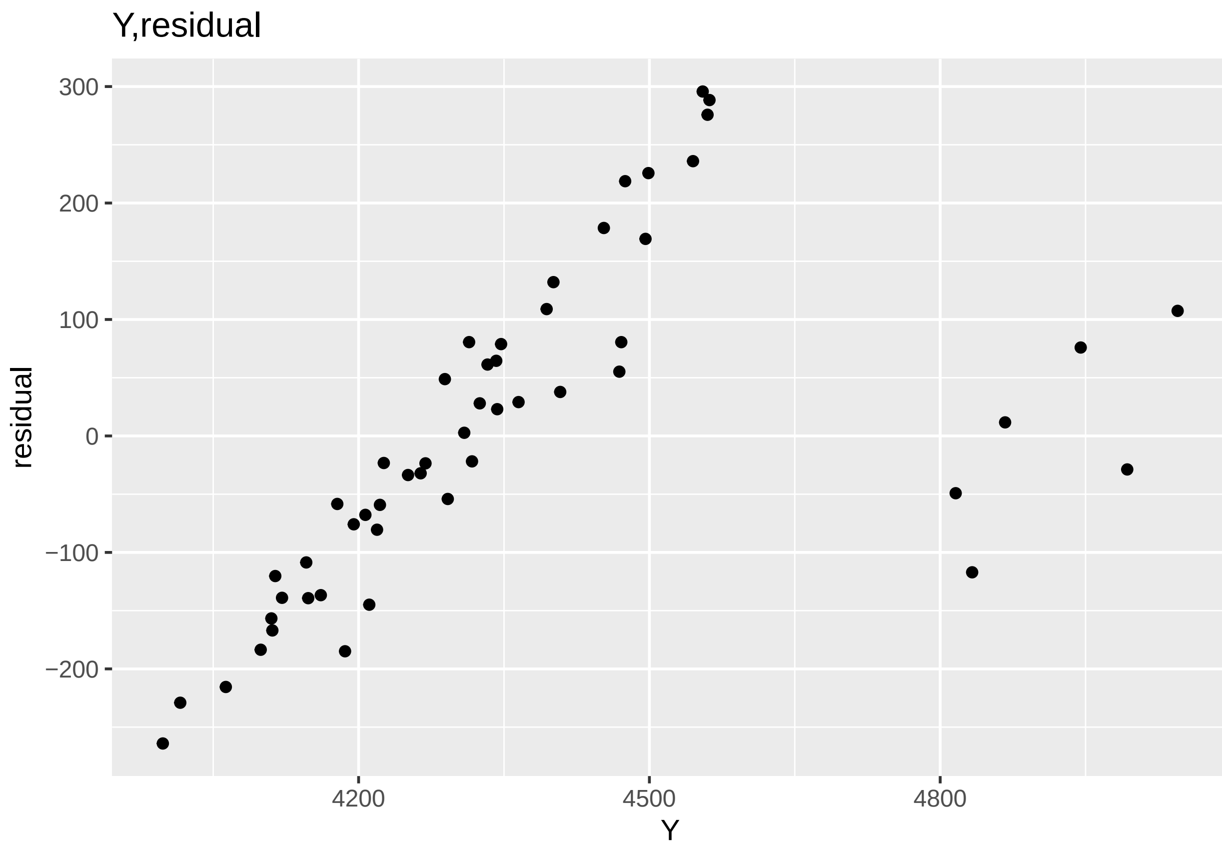
(c)

Plot of residual against Y shows that there is positive linear correlation between and residuals and Y. That is not some particular thing because we can percieve this by calculating the expression of $\mathrm{cov}(\mathrm{Y}, \hat{\mathrm{Y}})$, which will show a positive result. However this plot does tell us some additional information about the some outlier in the data, outliers also have positive linear relation with their residuals.

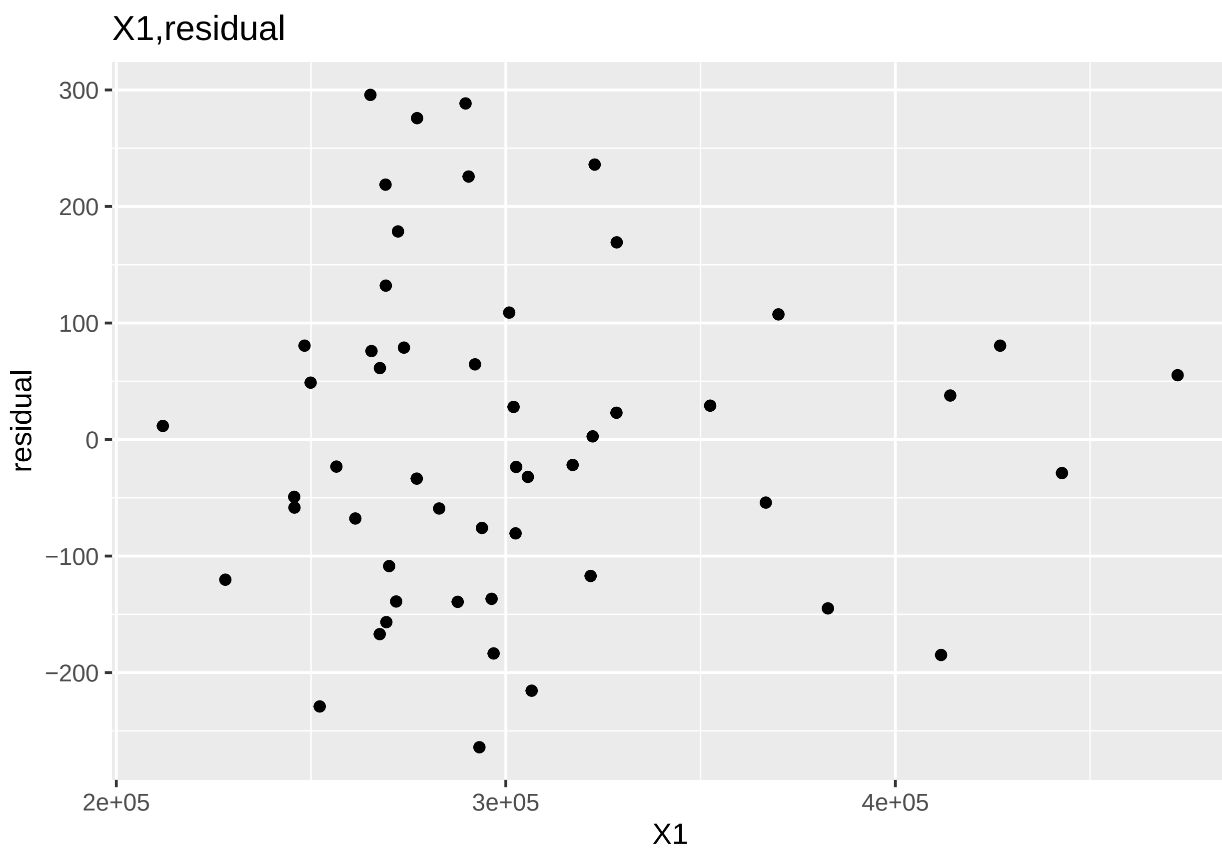
Plot of residual against X1,X2,X1X2 is well-behaved, since the points bounce randomly around residual=0 line. It shows a good fit of the regression line.

Plot of residual against X3 is not what we would like to see. because too many points centered in x=0.

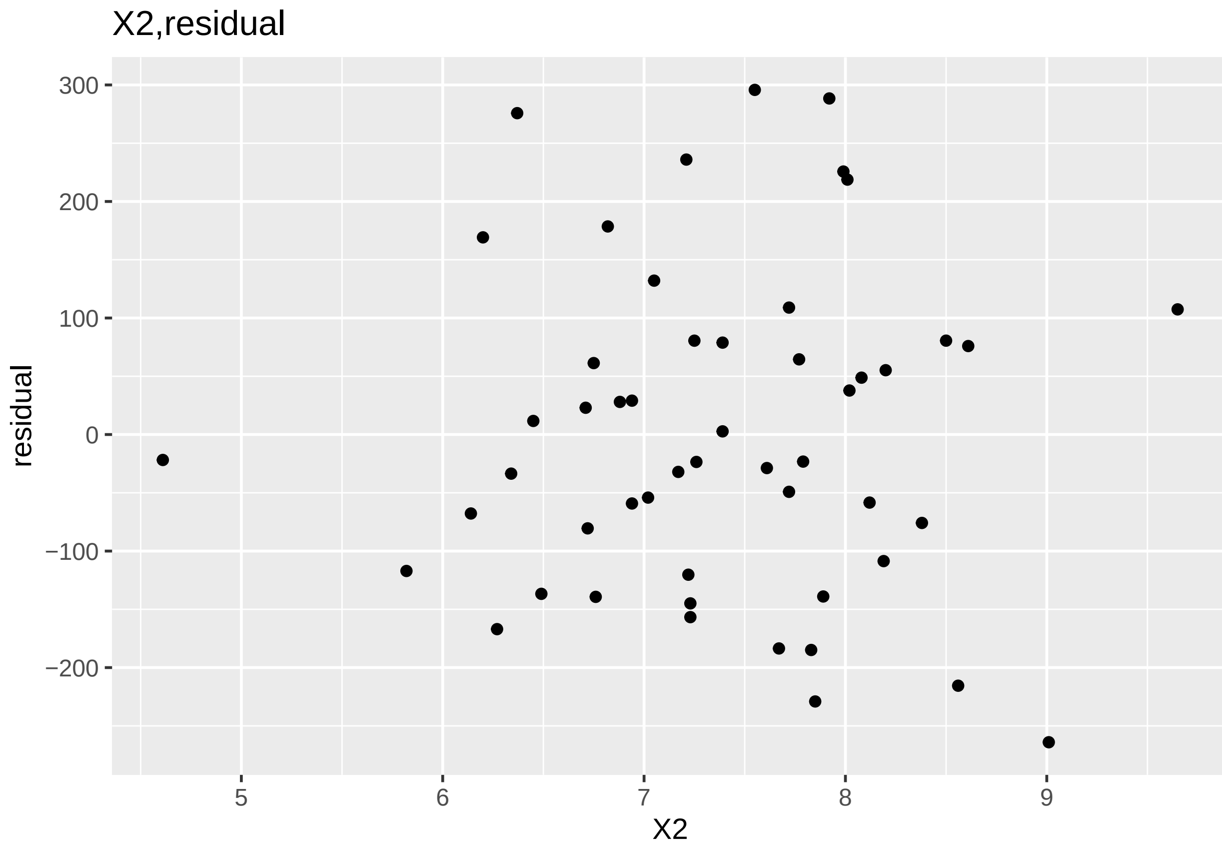
```
dat <- as.data.frame(cbind(residual,Y, X, X[,3]*X[,2]))
colnames(dat) <- c("residual", "Y", "X0" ,"X1", "X2", "X3" , "X1X2")
ggplot(data = dat) + geom_point(aes(x=Y,y=residual)) + ggtitle("Y,residual")
```



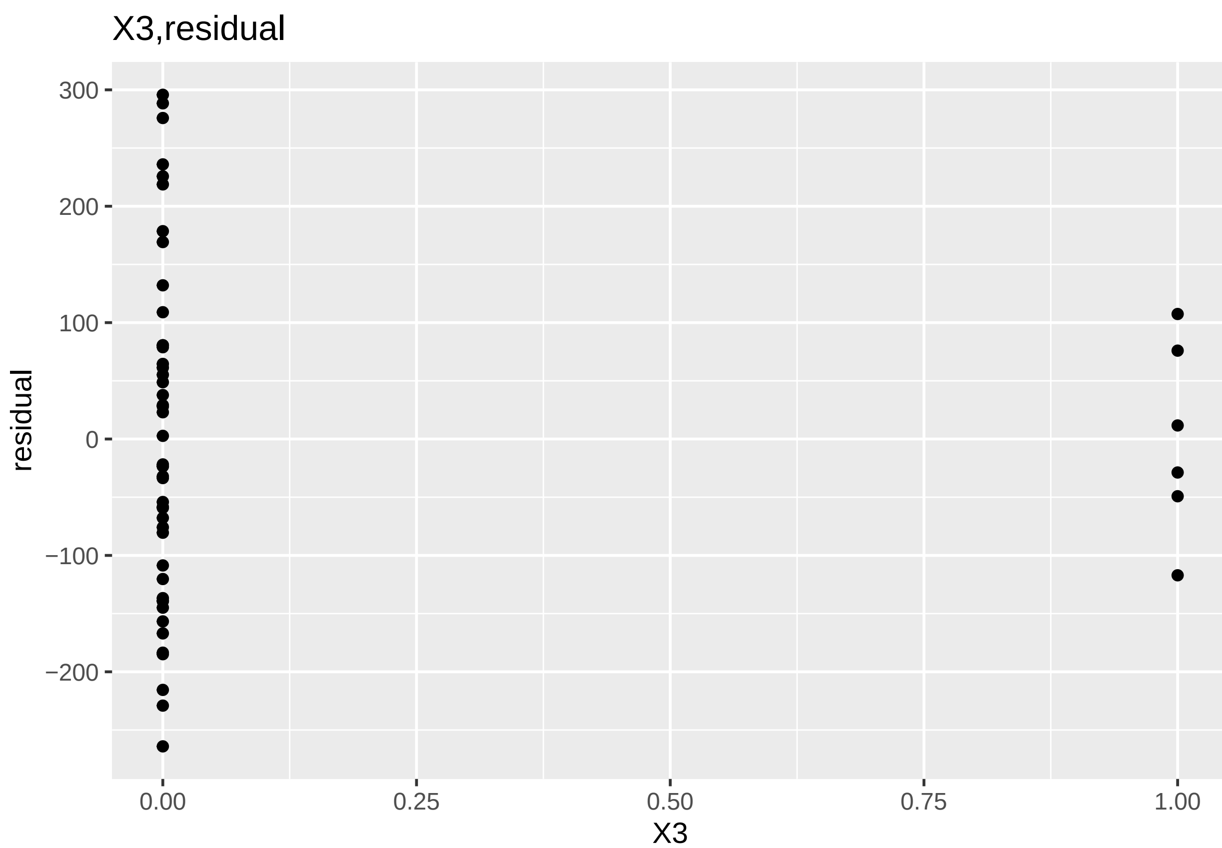

```
ggplot(data = dat) + geom_point(aes(x=X1,y=residual)) + ggtitle("X1,residual")
```



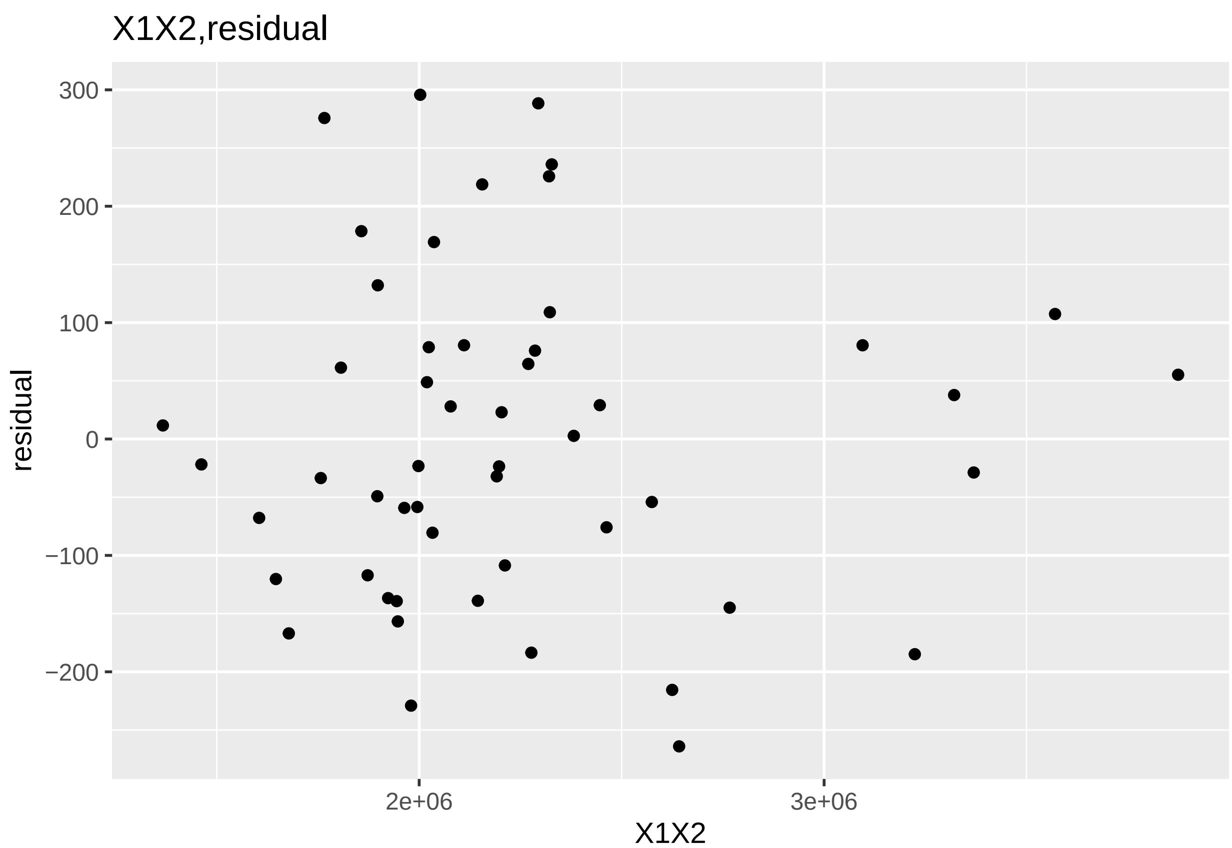

```r
ggplot(data = dat) + geom_point(aes(x=X2,y=residual)) + ggtitle("X2,residual")
```



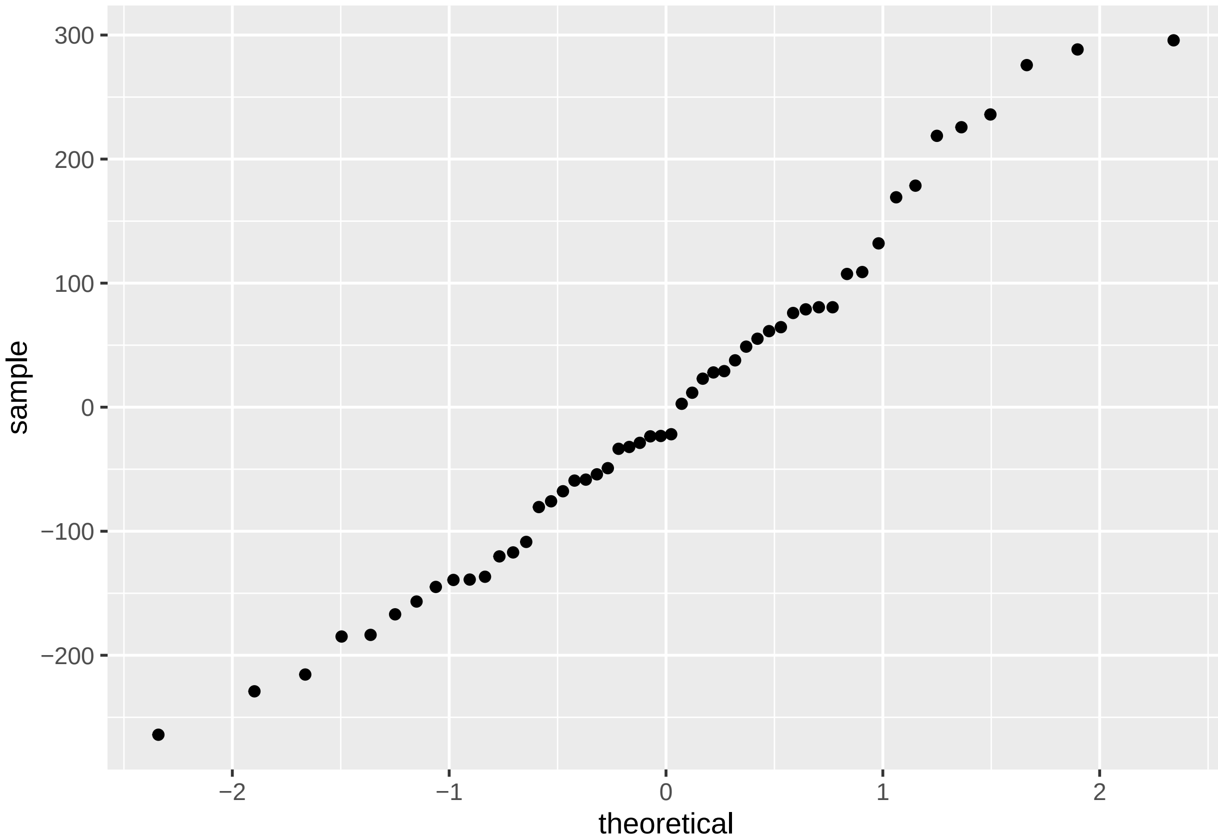

```r
ggplot(data = dat) + geom_point(aes(x=X3,y=residual)) + ggtitle("X3,residual")
```




```
ggplot(data = dat) + geom_point(aes(x=X1X2,y=residual)) + ggtitle("X1X2,residual")
```

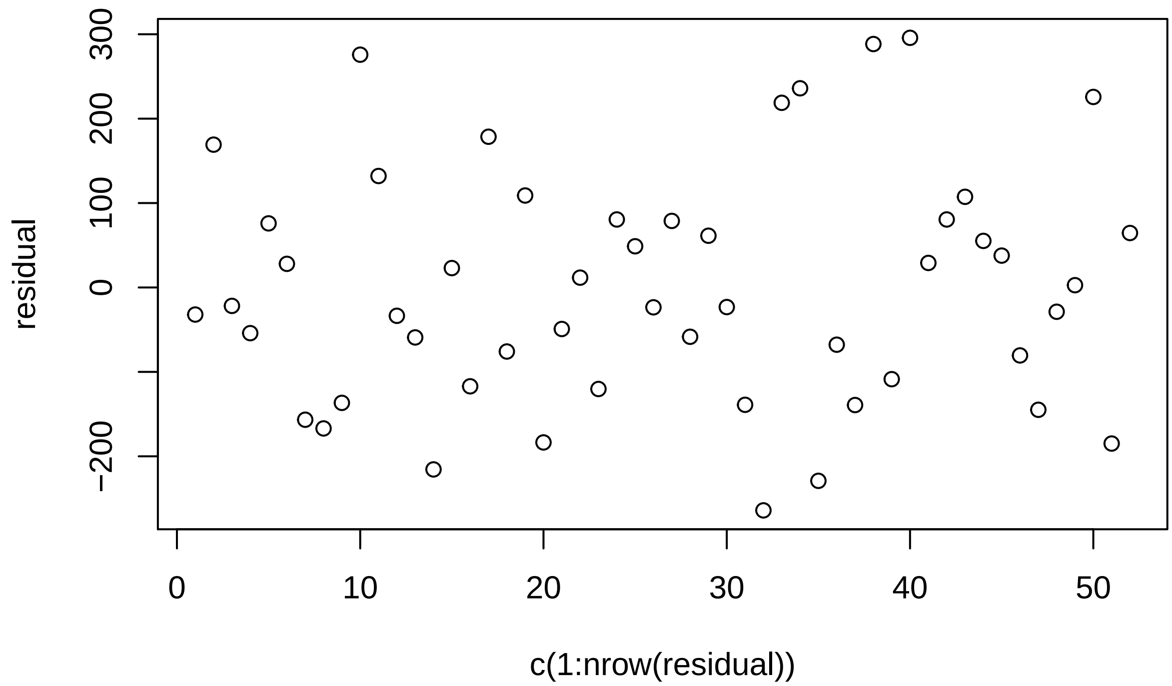



```
ggplot(data = dat) + stat_qq(aes(sample = residual))
```



(d) The erro terms are not correlated because the residuals scatter randomly around line residual=0.

```
plot(x = c(1:nrow(residual)),y = residual)
```



(e) Decision rule: If p-value is higher than α, we accept H_0, otherwise we reject H_0.we use t statistics in the test H0: constant variance vs Ha: not H0. The t statistic follows t distribution which yeilds a p-value = 0.133 is higher than 0.01, therefore we cannot reject H0, so we conclude that our regression model has constant error variance.

```
Y_hat <- cbind(Y_hat,residual)
Y_hat <- as.data.frame(Y_hat)
colnames(Y_hat) <- c("y_hat","residuals")
Y_hat <- Y_hat[order(Y_hat$y_hat),]
r1 <- Y_hat$residuals[1:26]
r2 <- Y_hat$residuals[27:52]
n1 <- 26
n2 <- 26
r1nod <- median(r1)
r2nod <- median(r2)
d1 <- abs(r1 - r1nod)
d2 <- abs(r2 - r2nod)
s <-(sum((d1 - mean(d1))^2) + sum((d2 - mean(d2))^2))/(10-2)
t_bf <- (mean(d1) - mean(d2))/sqrt((1/n1 + 1/n2) * s) # t distribution with df 5+5-2=8

pt(t_bf, 8, lower.tail = FALSE, log.p = FALSE) # P-value is .133, fail to reject constant variance
```

```
## [1] 0.1332698
```

(f) Decision rule:p-value smaller than α, reject H_0, otherwise, accept it. H_{00} : $\beta_1 = \beta_2 = \beta_3 = 0$, H_a :

not H_{00} The p-value is 3.315708e-12, reject H0. Conclusion: $\beta_1, \beta_2, \beta_3$ not all equals to 0, there is a regression relation between Y and predictors. Implication: $\beta_1, \beta_2, \beta_3$ at least one of them is not 0.

```
sse <- sum((X%*%solve(t(X)%*%X)%*%t(X)%*%Y - Y)^2)
ssr <- sum((X%*%solve(t(X)%*%X)%*%t(X)%*%Y - mean(Y))^2)
f <- (ssr)/3 / (sse/(52-4))              #f statistic
pf(f,3,48,lower.tail = FALSE)
```

```
## [1] 3.315708e-12
```

(g)

```
sse <- sum(residual^2)
s <- sqrt(sse/(52-4))
bon_t <- qt(0.05/6, 48,lower.tail = F)
upper <- beta_hat + bon_t * s * sqrt(diag(solve(t(X)%*%X)))
lower <- beta_hat - bon_t * s * sqrt(diag(solve(t(X)%*%X)))
t <- as.data.frame(cbind(lower,beta_hat,upper))
colnames(t) <- c("lower","beta","upper")
kable(t)
```

	lower	beta	upper
rep(1, nrow(p1))	3664.7317854	4149.8872120	4635.0426385
x1	-0.0001173	0.0007871	0.0016915
x2	-70.4516052	-13.1660192	44.1195668
x3	468.1558555	623.5544807	778.9531059

(h)

```
ssr_x1 <- sum((mean(Y) - X[,1:2] %*% solve(t(X[,1:2]) %*% X[,1:2]) %*% t(X[,1:2]) %*% Y)^2)
ssr_x3x1 <- sum((mean(Y) - X[,c(1,2,4)] %*% solve(t(X[,c(1,2,4)]) %*% X[,c(1,2,4)]) %*% t(X[,c(1,2,4)])
ssr_x3x1-ssr_x1    #ssr with x3 given x1
```

```
## [1] 2033565
```

```
ssr_x1x2x3 <- sum((mean(Y) - X %*% solve(t(X) %*% X) %*% t(X) %*% Y)^2)
ssr_x2 <- sum((mean(Y) - X[,c(1,3)] %*% solve(t(X[,c(1,3)]) %*% X[,c(1,3)]) %*% t(X[,c(1,3)]) %*% Y)^2)
ssr_x1x2x3 - ssr_x3x1 #ssr with x2 given x1 and x3
```

```
## [1] 6674.588
```

```
#obtain the ANOVA table by using package
anova(lm(y~x1 + x3 + x2,data = p1))
```

```
## Analysis of Variance Table
##
## Response: y
##           Df   Sum Sq Mean Sq F value    Pr(>F)
```

```
## x1          1  136366  136366  6.6417   0.01309 *
## x3          1 2033565 2033565 99.0443 2.963e-13 ***
## x2          1    6675    6675  0.3251   0.57123
## Residuals 48  985530   20532
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(i) $H_0 : \beta_2 = 0$, H_a : not H_{20} Decision rule: If p-value is smaller than α, we reject H0, otherwise accept null hypothesis. The p-value is 0.57, fail to reject H0. Conclusion:X2 can be dropped from the model. Implication: $\beta_2 = 0$

```
f <- (ssr_x1x2x3 - ssr_x3x1)/1/(sse/48)
pf(f,1,48,lower.tail = FALSE)
```

```
## [1] 0.5712274
```

(k)

```
#SSR of X1 / SST
sum((mean(Y) - X[,1:2] %*% solve(t(X[,1:2]) %*% X[,1:2]) %*% t(X[,1:2]) %*% Y)^2) / sum((Y - mean(Y))^2
```

```
## [1] 0.04312473
```

```
#SSR of X2 / SST
sum((mean(Y) - X[,c(1,3)] %*% solve(t(X[,c(1,3)]) %*% X[,c(1,3)]) %*% t(X[,c(1,3)]) %*% Y)^2) / sum((Y
```

```
## [1] 0.003603553
```

```
#SSR of X1+X2 / SST
sum((mean(Y) - X[,1:3] %*% solve(t(X[,1:3]) %*% X[,1:3]) %*% t(X[,1:3]) %*% Y)^2) / sum((Y - mean(Y))^2
```

```
## [1] 0.0449355
```

```
#SSR of X1+X2 - SSR of X2 / SSE of X2

(sum((mean(Y) - X[,1:3] %*% solve(t(X[,1:3]) %*% X[,1:3]) %*% t(X[,1:3]) %*% Y)^2) -
  sum((mean(Y) - X[,c(1,3)] %*% solve(t(X[,c(1,3)]) %*% X[,c(1,3)]) %*% t(X[,c(1,3)]) %*% Y)^2))/ sum((
```

```
## [1] 0.04133195
```

```
#SSR of X1+X2 - SSR of X1 / SSE of X1

(sum((mean(Y) - X[,1:3] %*% solve(t(X[,1:3]) %*% X[,1:3]) %*% t(X[,1:3]) %*% Y)^2) -
  sum((mean(Y) - X[,c(1,2)] %*% solve(t(X[,c(1,2)]) %*% X[,c(1,2)]) %*% t(X[,c(1,2)]) %*% Y)^2))/ sum((
```

```
## [1] 0.001810777
```

```r
# R squared = SSR/SST

sum((mean(Y) - X %*% solve(t(X) %*% X) %*% t(X) %*% Y)^2) / sum((Y - mean(Y))^2)
```

```
## [1] 0.6883342
```

(l)

```r
sdx <- X %>% as.data.frame() %>% mutate( x1 = (x1 - mean(x1))/sd(x1),
                      x2 = (x2 - mean(x2))/sd(x2),
                      x3 = (x3 - mean(x3))/sd(x3)) %>% as.matrix()
sdy <- (Y - mean(Y))/sd(Y)
beta_sd <- solve(t(sdx) %*% sdx) %*% t(sdx) %*% sdy
beta_sd
```

```
##                            [,1]
## rep(1, nrow(p1)) -1.407945e-15
## x1                1.747189e-01
## x2               -4.639130e-02
## x3                8.078617e-01
```

(m)

As we see in the summary, there is few correlation between different predictors.

It is a good way to fit a standardized regression model, because each predictors have very different scale, so it is a good way to make them standardized in order to compare which one would have larger effect on response variable while keep others constant.

```r
summary(lm(x1~x2,data = p1))
```

```
##
## Call:
## lm(formula = x1 ~ x2, data = p1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -85825 -33900 -14882  22577 165347
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   263272      65884   3.996 0.000212 ***
## x2              5348       8877   0.602 0.549575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55620 on 50 degrees of freedom
## Multiple R-squared:  0.007207,   Adjusted R-squared:  -0.01265
## F-statistic: 0.363 on 1 and 50 DF,  p-value: 0.5496
```

```r
summary(lm(x1~x3,data = p1))
```

```
##
## Call:
## lm(formula = x1 ~ x3, data = p1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -97669 -33031 -10519  16658 170686
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   301791       8222  36.704   <2e-16 ***
## x3              7823      24206   0.323    0.748
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55770 on 50 degrees of freedom
## Multiple R-squared:  0.002085,   Adjusted R-squared:  -0.01787
## F-statistic: 0.1044 on 1 and 50 DF,  p-value: 0.7479
```

```r
summary(lm(x3~x2,data = p1))
```

```
##
## Call:
## lm(formula = x3 ~ x2, data = p1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18372 -0.13579 -0.10932 -0.08556  0.94925
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.19186    0.38343  -0.500    0.619
## x2           0.04169    0.05166   0.807    0.424
##
## Residual standard error: 0.3237 on 50 degrees of freedom
## Multiple R-squared:  0.01285,    Adjusted R-squared:  -0.00689
## F-statistic: 0.651 on 1 and 50 DF,  p-value: 0.4236
```

```r
result <- c(0.007207,0.002085,0.01285)
result <- as.matrix(result)
result <- as.data.frame(result)
rownames(result) <- c("X1X2 R^2","X1X3 R^2","X2X3 R^2")
kable(result)
```

	V1
X1X2 R^2	0.007207
X1X3 R^2	0.002085
X2X3 R^2	0.012850

(n)

```r
beta_sd[2] - beta_hat[2]*sd(X[,2])/(sd(Y)) #beta for X1
```

```
## [1] -7.46625e-15
```

```r
beta_sd[3] - beta_hat[3]*sd(X[,3])/(sd(Y)) #beta for X2
```

```
## [1] -7.476658e-14
```

```r
beta_sd[4] - beta_hat[4]*sd(X[,4])/(sd(Y)) #beta for X3
```

```
## [1] 6.883383e-15
```

(o) $SSR(X_1)$ = 2.199 $SSR(X_{1}|X_{2}) = SSR(X_{1},X_{2}) - SSR(X_{2})) = 0.092 + 2.199 - 0.184 = 2.107 $ There is difference between them, but the difference is not substantial.

```r
anova(lm(scale(y)~scale(x1)+scale(x2),data = p1))[1,2] + anova(lm(scale(y)~scale(x1)+scale(x2),data = p
```

```
## [1] 2.107929
```

```r
anova(lm(scale(y)~scale(x1)+scale(x2),data = p1))[1,2] + anova(lm(scale(y)~scale(x1)+scale(x2),data = p
```

```
## [1] -0.09143157
```

```r
0.092 + 2.199 - 0.184
```

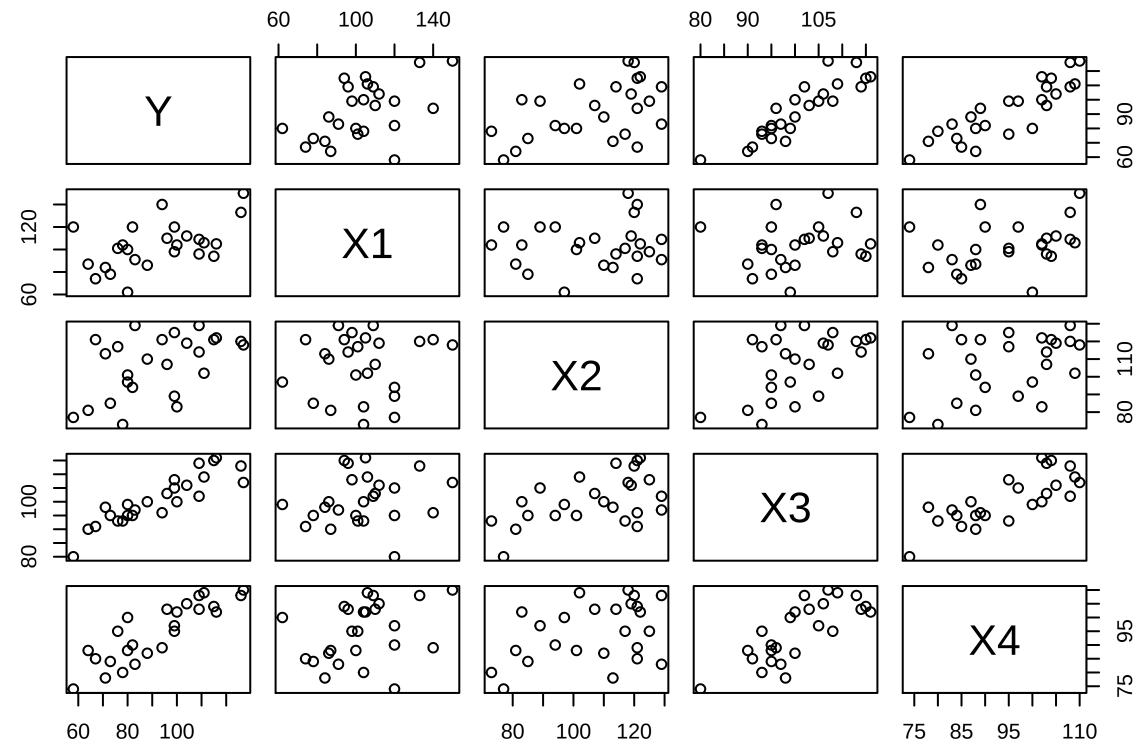
```
## [1] 2.107
```

Problem 2

(a)

The scatter plots suggests there are linear relationships between (Y,X1), (Y,X2), (Y,X3), (Y,X4), (X4,X3). The plots suggest the linear relation between (Y,X1) and (Y,X2) are weak, but strong for the other two. There is colinearity between X4 and X3 by taking a look at the plot of X3 and X4. X3 and X2 also has linear relation. Therefore there exists multicolinearity problem.

```r
p2 <- read.table("jobhw4.txt",header = T)
pairs(p2)
```




```r
cor(p2)
```

```
##            Y        X1        X2        X3        X4
## Y  1.0000000 0.5144107 0.4970057 0.8970645 0.8693865
## X1 0.5144107 1.0000000 0.1022689 0.1807692 0.3266632
## X2 0.4970057 0.1022689 1.0000000 0.5190448 0.3967101
## X3 0.8970645 0.1807692 0.5190448 1.0000000 0.7820385
## X4 0.8693865 0.3266632 0.3967101 0.7820385 1.0000000
```

(b) The p-value of X2 is not significant, but for the others are significant. Thus we could retain the others and drop X2.

```r
model <- lm(Y ~ ., data = p2)
summary(model)
```

```
##
## Call:
## lm(formula = Y ~ ., data = p2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9779 -3.4506  0.0941  2.4749  5.9959
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -124.38182    9.94106 -12.512 6.48e-11 ***
## X1             0.29573    0.04397   6.725 1.52e-06 ***
## X2             0.04829    0.05662   0.853  0.40383
## X3             1.30601    0.16409   7.959 1.26e-07 ***
## X4             0.51982    0.13194   3.940  0.00081 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.099 on 20 degrees of freedom
## Multiple R-squared:  0.9629, Adjusted R-squared:  0.9555
## F-statistic: 129.7 on 4 and 20 DF,  p-value: 5.262e-14
```

(c) The four best R^2 is 0.9628918,0.9615422,0.9340931,0.9329956. {X1,X2,X3,X4},{X1,X3,X4},{X1,X2,X3},{X1,X3}

```r
library(leaps)
result2 <- leaps(y=p2[,1],x=p2[,2:5],method = "r2")
result2$which
```

```
##       1     2     3     4
## 1 FALSE FALSE  TRUE FALSE
## 1 FALSE FALSE FALSE  TRUE
## 1  TRUE FALSE FALSE FALSE
## 1 FALSE  TRUE FALSE FALSE
## 2  TRUE FALSE  TRUE FALSE
## 2 FALSE FALSE  TRUE  TRUE
## 2  TRUE FALSE FALSE  TRUE
## 2 FALSE  TRUE  TRUE FALSE
## 2 FALSE  TRUE FALSE  TRUE
## 2  TRUE  TRUE FALSE FALSE
## 3  TRUE FALSE  TRUE  TRUE
## 3  TRUE  TRUE  TRUE FALSE
## 3 FALSE  TRUE  TRUE  TRUE
## 3  TRUE  TRUE FALSE  TRUE
## 4  TRUE  TRUE  TRUE  TRUE
```

```r
sort(result2$r2)
```

```
##  [1] 0.2470147 0.2646184 0.4641948 0.7558329 0.7832923 0.8047247 0.8060733
##  [8] 0.8152656 0.8453581 0.8772573 0.8789698 0.9329956 0.9340931 0.9615422
## [15] 0.9628918
```

```r
result2$r2
```

```
##  [1] 0.8047247 0.7558329 0.2646184 0.2470147 0.9329956 0.8772573 0.8152656
##  [8] 0.8060733 0.7832923 0.4641948 0.9615422 0.9340931 0.8789698 0.8453581
## [15] 0.9628918
```

(d) I would use AIC. I conclude the first model has lower AIC thus better than other three.

```r
AIC(lm(Y ~ ., data = p2))
```

```
## [1] 147.9011
```

```r
AIC(lm(Y ~ X1+X3+X4, data = p2))
```

```
## [1] 146.7942
```

```r
AIC(lm(Y ~ X1+X2+X3, data = p2))
```

```
## [1] 160.2613
```

```r
AIC(lm(Y ~ X1+X3, data = p2))
```

```
## [1] 158.6741
```

(e) The best subset should be X_1, X_3, X_4 because it has a very small SSE(close to full model), small BIC, and a big R^2.

```r
bmodel <- regsubsets(Y~.,data = p2, method = "backward")
b <- summary(bmodel)
b
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = p2, method = "backward")
## 4 Variables  (and intercept)
##    Forced in Forced out
## X1     FALSE      FALSE
## X2     FALSE      FALSE
## X3     FALSE      FALSE
## X4     FALSE      FALSE
## 1 subsets of each size up to 4
## Selection Algorithm: backward
##          X1  X2  X3  X4
## 1  ( 1 ) " " " " "*" " "
## 2  ( 1 ) "*" " " "*" " "
## 3  ( 1 ) "*" " " "*" "*"
## 4  ( 1 ) "*" "*" "*" "*"
```

```r
b$rss
```

```
## [1] 1768.0228  606.6574  348.1970  335.9775
```

```r
b$bic
```

```
## [1] -34.39587 -57.91831 -68.57933 -66.25356
```

```r
b$rsq
```

```
## [1] 0.8047247 0.9329956 0.9615422 0.9628918
```

(f) The best subset should be X_1, X_3, X_4 because it has a very small SSE(close to full model), small BIC, and a big R^2.

```r
fmodel <- regsubsets(Y~.,data = p2,method = "forward")
f <- summary(fmodel)
f
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = p2, method = "forward")
## 4 Variables  (and intercept)
##    Forced in Forced out
## X1     FALSE      FALSE
## X2     FALSE      FALSE
## X3     FALSE      FALSE
## X4     FALSE      FALSE
## 1 subsets of each size up to 4
## Selection Algorithm: forward
##          X1  X2  X3  X4
## 1  ( 1 ) " " " " "*" " "
## 2  ( 1 ) "*" " " "*" " "
## 3  ( 1 ) "*" " " "*" "*"
## 4  ( 1 ) "*" "*" "*" "*"
```

```r
f$rss
```

```
## [1] 1768.0228  606.6574  348.1970  335.9775
```

```r
f$bic
```

```
## [1] -34.39587 -57.91831 -68.57933 -66.25356
```

```r
f$rsq
```

```
## [1] 0.8047247 0.9329956 0.9615422 0.9628918
```

(g) The best subset is also X_1, X_3, X_4

```r
g <- lm(Y~.,data=p2)
step(g)
```

```
## Start:  AIC=74.95
## Y ~ X1 + X2 + X3 + X4
##
##        Df Sum of Sq     RSS     AIC
## - X2    1     12.22  348.20  73.847
```

```
## <none>                335.98  74.954
## - X4    1    260.74  596.72  87.314
## - X1    1    759.83 1095.81 102.509
## - X3    1   1064.15 1400.13 108.636
##
## Step:  AIC=73.85
## Y ~ X1 + X3 + X4
##
##        Df Sum of Sq     RSS     AIC
## <none>                348.20  73.847
## - X4    1    258.46  606.66  85.727
## - X1    1    763.12 1111.31 100.861
## - X3    1   1324.39 1672.59 111.081
```

```
##
## Call:
## lm(formula = Y ~ X1 + X3 + X4, data = p2)
##
## Coefficients:
## (Intercept)           X1           X3           X4
##   -124.2000       0.2963       1.3570       0.5174
```

(h) They are same. Because we only have 4 variables, and all of them are continuous, thus it is very likely that all methods give out same result.