Homework 8

Yi Chen(yc3356)

December 4, 2017

Homework 8

Problem 9.9

```
## read the data
setwd("C:/Users/cheny/Desktop/study/linear regression model/homework/homework record/homework
 8")
satisfaction <- read.table("6.15.txt",header = FALSE, col.names = c('y','x1','x2','x3'))
```

a

```
# first use the AIC and the method of stepwise regression to find the best subset of the vari
ables.

reg3 <- lm(data = satisfaction,y ~ x1)
reg1 <- lm(data = satisfaction,y ~ x2)
reg2 <- lm(data = satisfaction,y ~ x3)
reg5 <- lm(data = satisfaction,y ~ x1 + x2)
reg6 <- lm(data = satisfaction,y ~ x1 + x3)
reg4 <- lm(data = satisfaction,y ~ x2 + x3)
reg7 <- lm(data = satisfaction,y ~ x1 + x2 + x3)

## AIC
AIC <- AIC(reg1,reg2,reg3,reg4,reg5,reg6,reg7)
AIC
```

```
##      df      AIC
## reg1  3 376.6735
## reg2  3 372.7561
## reg3  3 353.0717
## reg4  4 370.3874
## reg5  4 350.5100
## reg6  4 347.6030
## reg7  5 348.7273
```

```
# actually a quicker way is
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.4.3
```

```
stepAIC(reg7,direction = 'both')
```

```
## Start:  AIC=216.18
## y ~ x1 + x2 + x3
##
##        Df Sum of Sq    RSS    AIC
## - x2    1     81.66 4330.5 215.06
## <none>              4248.8 216.19
## - x3    1    364.16 4613.0 217.97
## - x1    1   2857.55 7106.4 237.84
##
## Step:  AIC=215.06
## y ~ x1 + x3
##
##        Df Sum of Sq    RSS    AIC
## <none>              4330.5 215.06
## + x2    1      81.7 4248.8 216.19
## - x3    1     763.4 5093.9 220.53
## - x1    1    3483.9 7814.4 240.21
```

```
##
## Call:
## lm(formula = y ~ x1 + x3, data = satisfaction)
##
## Coefficients:
## (Intercept)           x1           x3
##      145.94        -1.20       -16.74
```

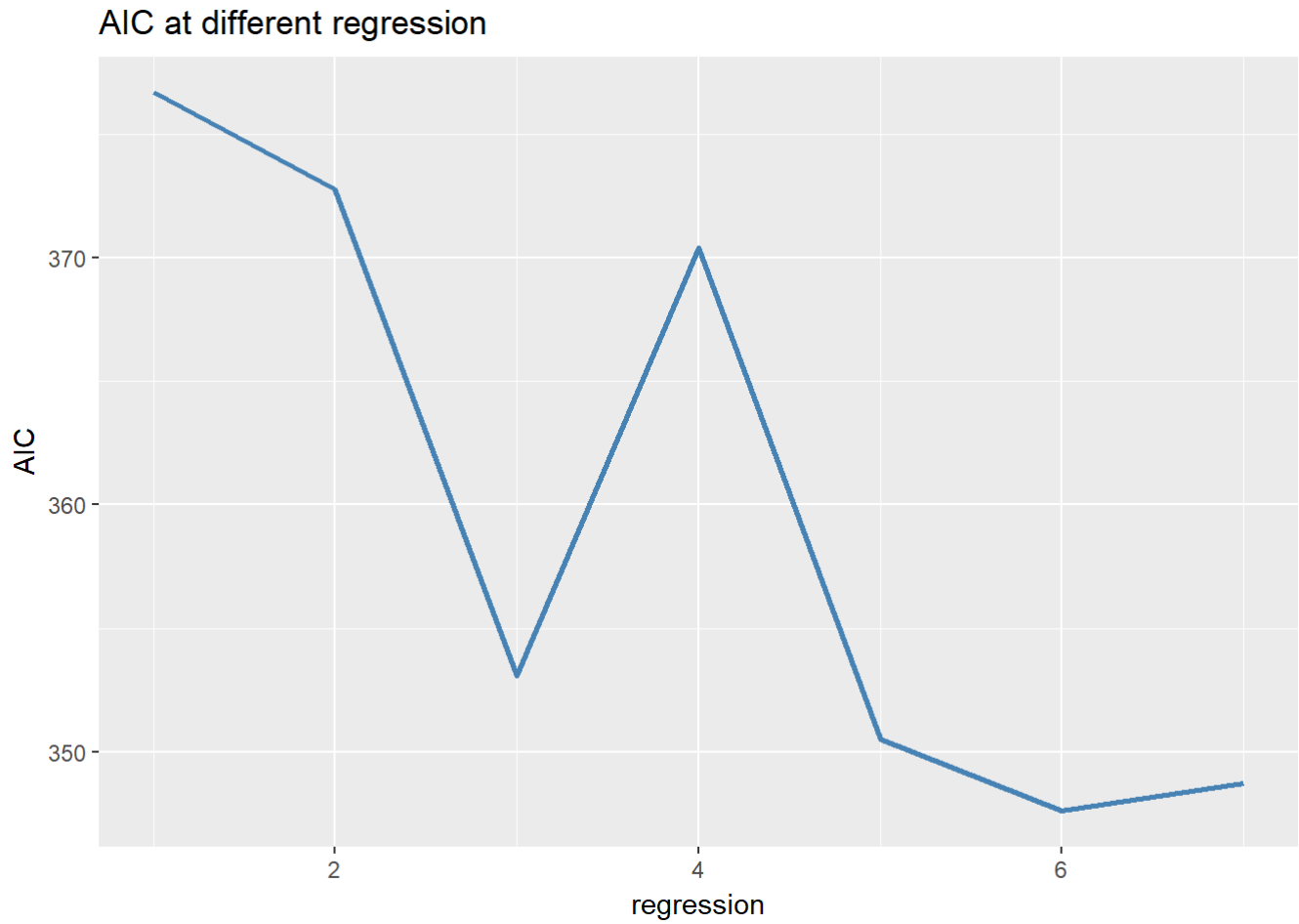
As we can see the best subset is x1 and x3 which has the smallest AIC

Now plot the picture

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
ggplot()+
        geom_line(aes(x=1:7,y=AIC$AIC),col='steelblue',lwd=1)+
        labs(title='AIC at different regression',x='regression',y='AIC')
```

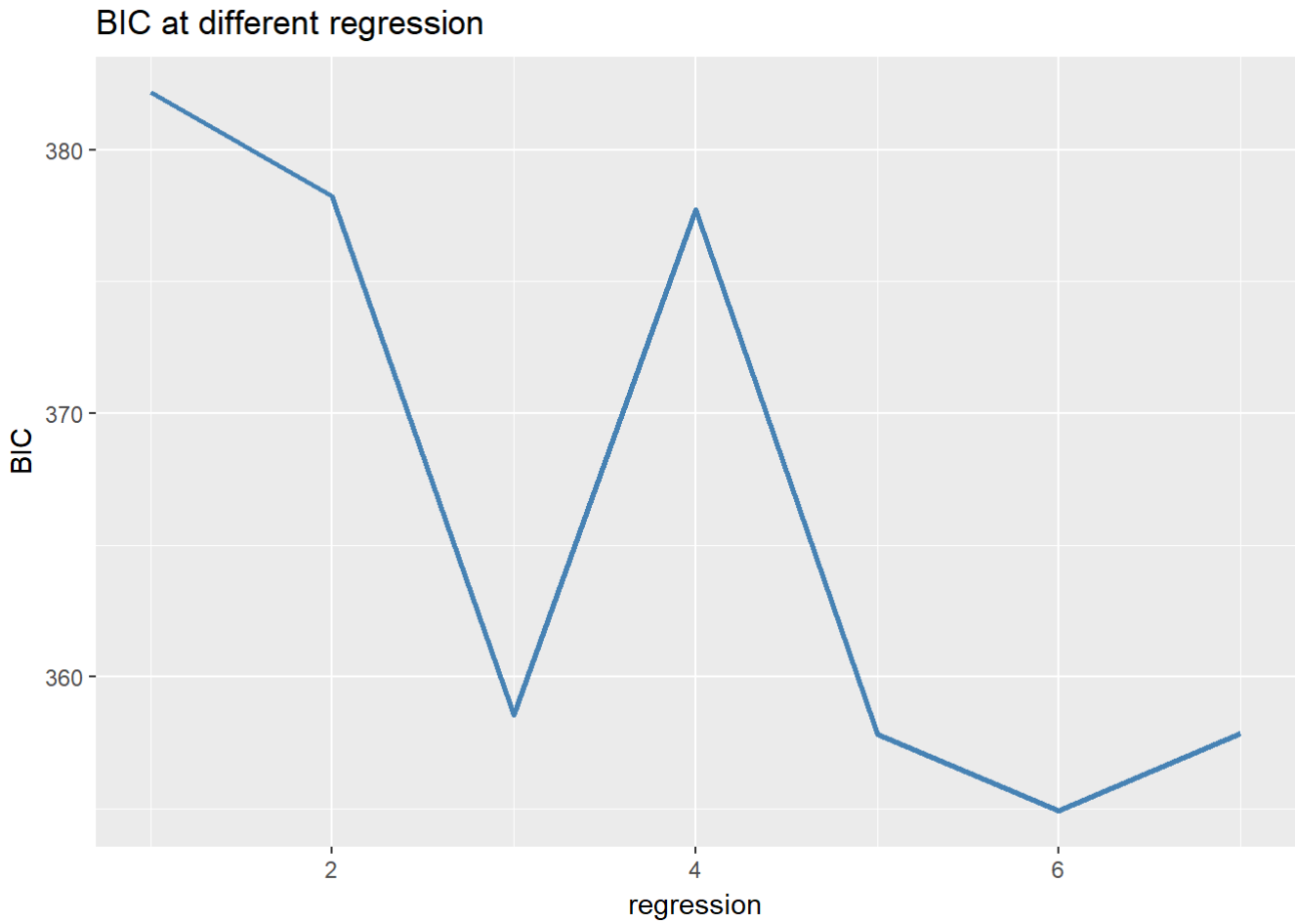


As we can see the regression 6 has the smallest AIC which tell us the best subset is x1 and x3

```
## BIC
BIC <- BIC(reg1,reg2,reg3,reg4,reg5,reg6,reg7)
BIC
```

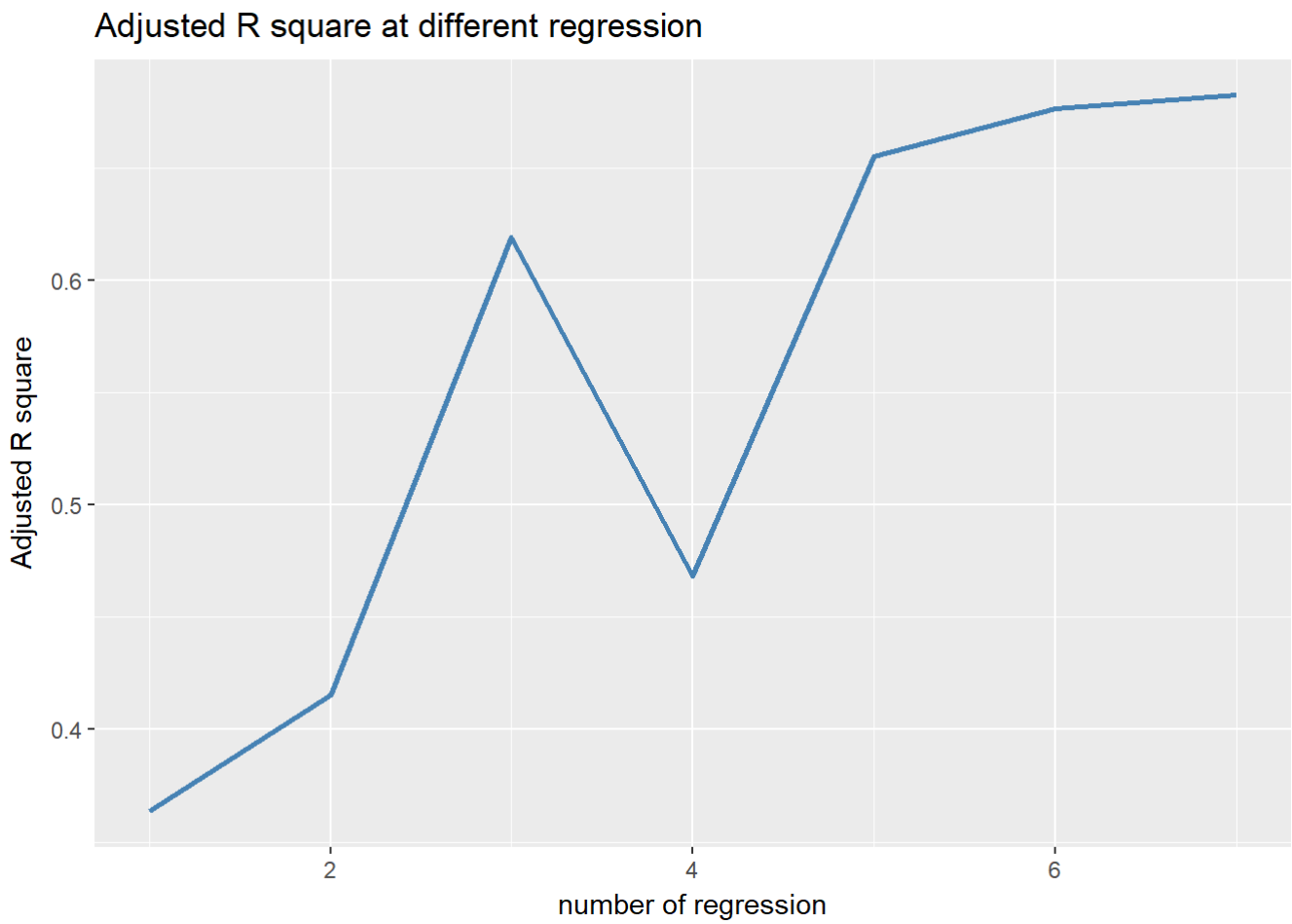
```
##      df      BIC
## reg1  3 382.1595
## reg2  3 378.2420
## reg3  3 358.5577
## reg4  4 377.7019
## reg5  4 357.8246
## reg6  4 354.9176
## reg7  5 357.8705
```

```
library(ggplot2)
ggplot()+
        geom_line(aes(x=1:7,y=BIC$BIC),col='steelblue',lwd=1)+
        labs(title='BIC at different regression',x='regression',y='BIC')
```



As we can see the regression 6 has the smallest AIC which tell us the best subset is x1 and x3 The result is same with AIC

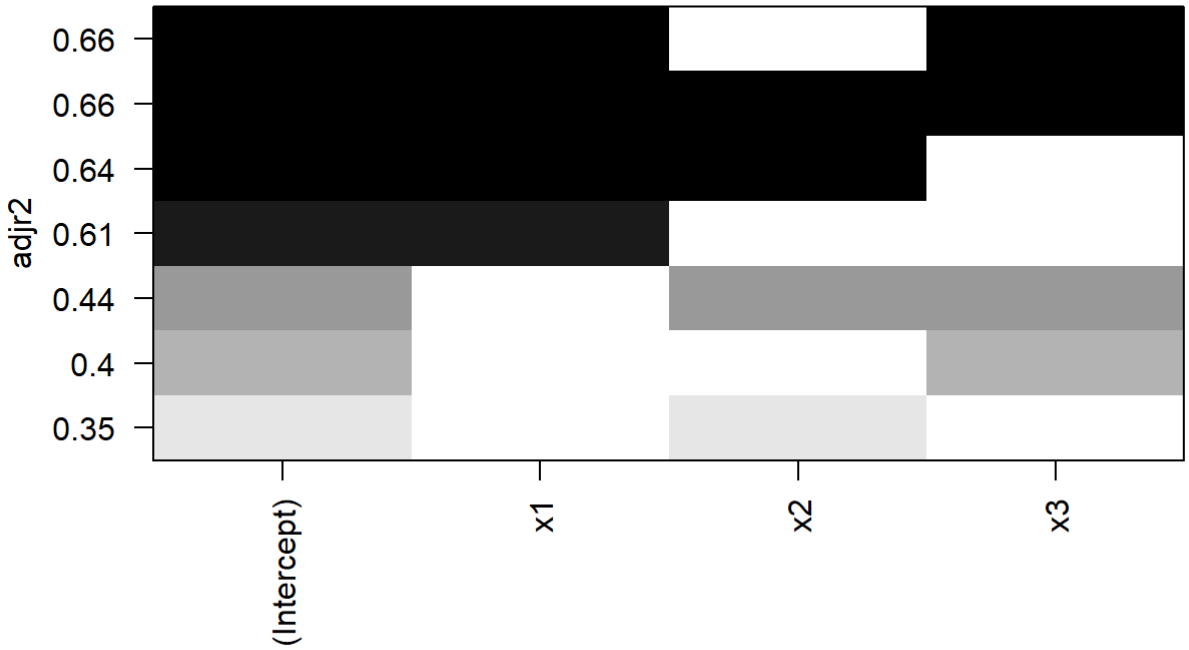
```
## Adjusted R square
R1 <- summary(reg1)$r.squared
R2 <- summary(reg2)$r.squared
R3 <- summary(reg3)$r.squared
R4 <- summary(reg4)$r.squared
R5 <- summary(reg5)$r.squared
R6 <- summary(reg6)$r.squared
R7 <- summary(reg7)$r.squared
adjusted_R <- c(R1,R2,R3,R4,R5,R6,R7)
DF <- c(3,3,3,4,4,4,5)
adjusted_R_summary <- data.frame(regnumber <- 1:7, adjusted_R <- adjusted_R,df=DF)

ggplot(data=adjusted_R_summary)+
        geom_line(aes(x=regnumber,y=adjusted_R),col='steelblue',lwd=1)+
        labs(title='Adjusted R square at different regression',x='number of regression',y='Ad
justed R square')
```




```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 3.4.2
```

```
leaps <- regsubsets(data=satisfaction,y~x1+x2+x3, nbest = 4)
plot(leaps,scale = 'adjr2')
```



Based on the Adjusted R square, the last regression is best which shows that x1,x2,x3 is the best subset of the variables

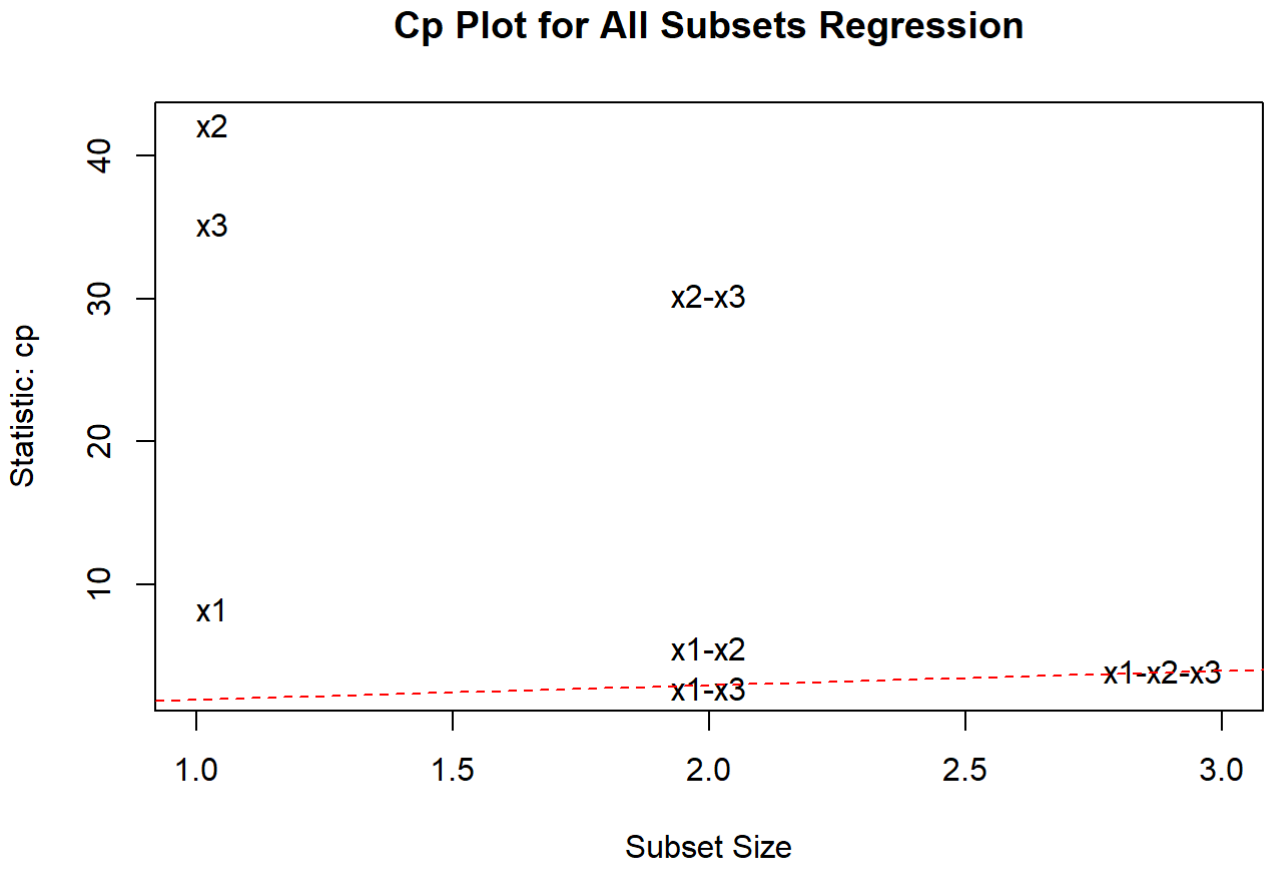
```
library(car)
```

```
## Warning: package 'car' was built under R version 3.4.2
```

```
subsets(leaps,statistic="cp",main="Cp Plot for All Subsets Regression",legend = FALSE)
```

```
##    Abbreviation
## x1           x1
## x2           x2
## x3           x3
```

```
abline(1,1,lty=2,col="red")
```



As we can see from the plot the best subset is also x1-x3

b

As we can see apart from the Adjusted R square the AIC,BIC and CP all have the same best subset. Actually AIC,BIC anc CP will Usually have the same result. Because all these statistics focus both on in sample and out of sample performance. Particularlly, AIC and BIC only have differen penalty term. But it is not alway they will have the same results. Because the penalty term for them is different, thus they would perfer differen value. For example, usually the AIC would have a bigger P value which lead AIC have less probability of under selection.

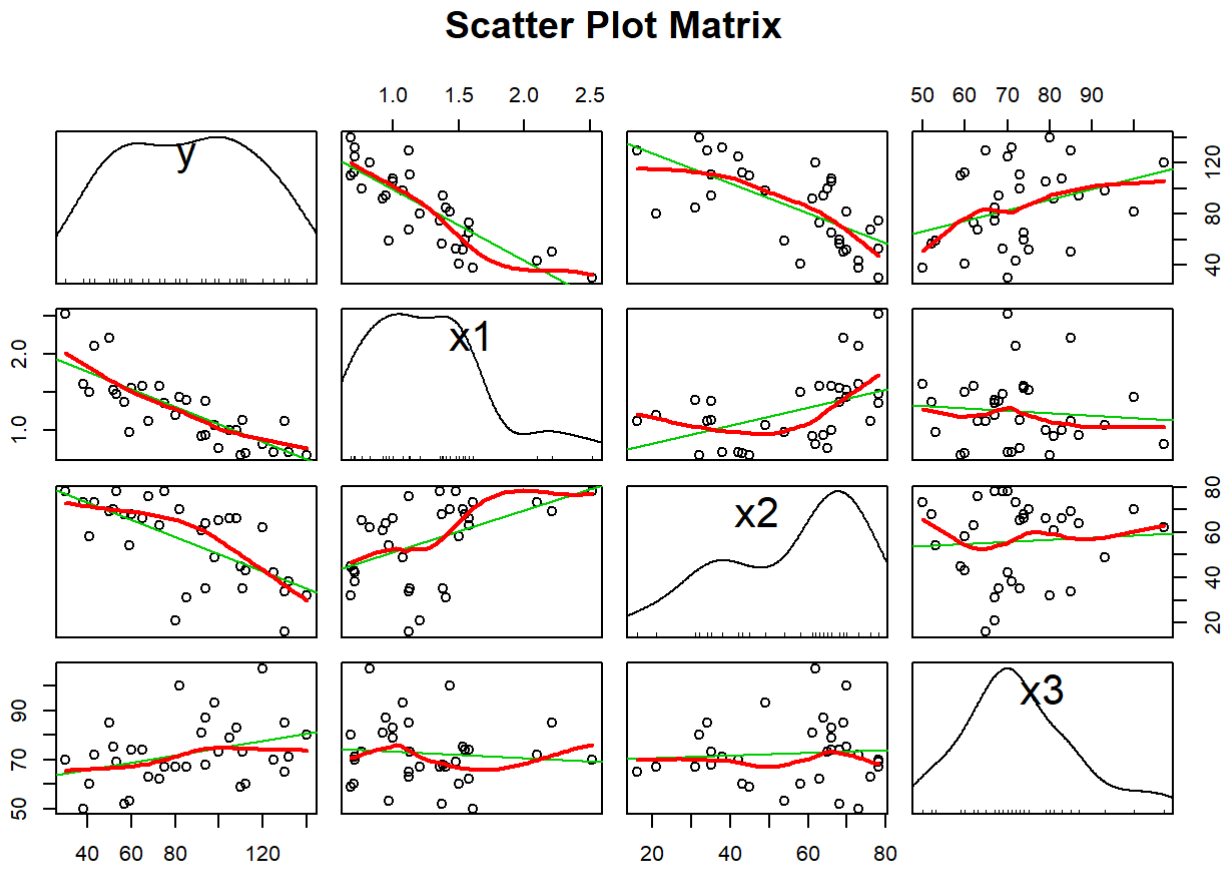
C

Here clearly backward delection would be more efficient. Because it only need to have relative less time of regression to find the best subset.

Problem 9.15

problem b

```
## read the data
kidney <- read.table('9.15.txt',header = FALSE, col.names = c('y','x1','x2','x3'))


scatterplotMatrix(kidney,spread=FALSE,main="Scatter Plot Matrix")
```




```
x <- kidney[,2:4]
cor(x)
```

```
##             x1         x2          x3
## x1  1.00000000 0.46773179 -0.08898262
## x2  0.46773179 1.00000000  0.06848147
## x3 -0.08898262 0.06848147  1.00000000
```

As we can see from the plot that: y have a obvious linear relation with x1, x2 and x3. And for y against x1 and x2 , the relations are negative but for y against x3 the relation is positive.

And as we can see from the correlation matrix, x1 and x2 have a relatively high correlation. Other variables tends to have pretty smally correlation. Which indecate that there do exist some kind of multicollinearity problem but the problem is not serious.

problem c

```
reg_1 <- lm(data=kidney,y~.)
summary(reg_1)
```

```
##
## Call:
## lm(formula = y ~ ., data = kidney)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.668  -7.002   1.518   9.905  16.006
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 120.0473    14.7737   8.126 5.84e-09 ***
## x1          -39.9393     5.6000  -7.132 7.55e-08 ***
## x2           -0.7368     0.1414  -5.211 1.41e-05 ***
## x3            0.7764     0.1719   4.517 9.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.46 on 29 degrees of freedom
## Multiple R-squared:  0.8548, Adjusted R-squared:  0.8398
## F-statistic: 56.92 on 3 and 29 DF,  p-value: 2.885e-12
```

```
vif(reg_1)
```

```
##       x1       x2       x3
## 1.304608 1.300377 1.023997
```

As we can see: first, from the summary of the regression we know all the variables pass the t test and the adjusted r square is over 0.8. Which means that the model is good.

By the way, the vif statisics shows that all the varibales have relative small vif which are near 1. So the multicollinearity problem is not serious and there is no need to delete any variables.

proble 9.16

```
# first analysis the first order
## base on the adjusted R square
y <- kidney[,1]
x <- scale(kidney[2:4],scale = TRUE)
x1_2 <- scale(kidney[,2],scale = FALSE)^2
x2_2 <- scale(kidney[,3],scale = FALSE)^2
x3_2 <- scale(kidney[,4],scale = FALSE)^2
x1_x2 <- scale(kidney[,2],scale = FALSE) * scale(kidney[,3],scale = FALSE)
x1_x3 <- scale(kidney[,2],scale = FALSE) * scale(kidney[,4],scale = FALSE)
x2_x3 <- scale(kidney[,3],scale = FALSE) * scale(kidney[,4],scale = FALSE)

x_total <- cbind(x,x1_2,x2_2,x3_2,x1_x2,x1_x3,x2_x3)
colnames(x_total) <- c('x1','x2','x3','x1_2','x2_2','x3_2','x1_x2','x1_x3','x2_x3')

result <- leaps::leaps(x = x_total , y=y ,method = "adjr2")
order_index <- order(result$adjr2,decreasing = TRUE)
result$which[order_index[1:3],]
```

```
##      1    2    3     4     5    6    7     8     9
## 5 TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE FALSE
## 6 TRUE TRUE TRUE FALSE  TRUE TRUE TRUE FALSE FALSE
## 6 TRUE TRUE TRUE FALSE FALSE TRUE TRUE  TRUE FALSE
```

```
result$adjr2[order_index[1:3]]
```

```
## [1] 0.8668497 0.8652362 0.8638250
```

```
## base on the adjusted R square
result2 <- leaps::leaps(x = x_total , y=y ,method = "Cp")
order_index <- order(result2$Cp,decreasing = FALSE)
result2$which[order_index[1:3],]
```

```
##      1    2    3     4     5     6    7     8     9
## 4 TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE FALSE
## 5 TRUE TRUE TRUE FALSE FALSE  TRUE TRUE FALSE FALSE
## 5 TRUE TRUE TRUE FALSE  TRUE FALSE TRUE FALSE FALSE
```

```
result2$Cp[order_index[1:3]]
```

```
## [1] 3.302215 3.384990 4.447976
```

```
kidney[2:4,] <- scale(kidney[2,4],scale = FALSE)
fit <- lm(y~1,data = kidney)
step(fit,direction = "forward",scope = ~x1+x2+x3+I(x1^2)+I(x2^2)+I(x3^2)+x1:x2+x1:x3+x2:x3)
```

```
## Start:  AIC=243.31
## y ~ 1
##
##           Df Sum of Sq   RSS    AIC
## + x3       1   26462.3 22992 220.03
## + I(x3^2)  1   22134.0 27320 225.72
## + I(x1^2)  1    4170.8 45283 242.40
## <none>                 49454 243.31
## + I(x2^2)  1     875.5 48578 244.72
## + x2       1     360.8 49093 245.06
## + x1       1      72.5 49381 245.26
##
## Step:  AIC=220.03
## y ~ x3
##
##           Df Sum of Sq   RSS    AIC
## + I(x1^2)  1   11118.8 11873 200.22
## + I(x2^2)  1   11085.5 11906 200.31
## + x1       1    9930.5 13061 203.37
## + x2       1    9333.7 13658 204.84
## <none>                 22992 220.03
## + I(x3^2)  1     107.2 22885 221.88
##
## Step:  AIC=200.22
## y ~ x3 + I(x1^2)
##
##           Df Sum of Sq     RSS    AIC
## + I(x2^2)  1    3174.7  8698.1 191.95
## + I(x3^2)  1    2804.2  9068.6 193.33
## + x2       1    2343.5  9529.3 194.97
## <none>                 11872.8 200.22
## + x1       1      28.9 11843.9 202.14
##
## Step:  AIC=191.95
## y ~ x3 + I(x1^2) + I(x2^2)
##
##           Df Sum of Sq    RSS    AIC
## + I(x3^2)  1   3162.03 5536.0 179.04
## + x2       1    752.07 7946.0 190.97
## <none>                 8698.1 191.95
## + x1       1    194.61 8503.5 193.21
##
## Step:  AIC=179.04
## y ~ x3 + I(x1^2) + I(x2^2) + I(x3^2)
##
##        Df Sum of Sq    RSS    AIC
## + x1    1   1474.51 4061.5 170.82
## <none>              5536.0 179.04
## + x2    1    305.27 5230.8 179.17
##
## Step:  AIC=170.82
## y ~ x3 + I(x1^2) + I(x2^2) + I(x3^2) + x1
##
##         Df Sum of Sq    RSS    AIC
## + x2     1    291.04 3770.5 170.37
## <none>               4061.5 170.82
## + x1:x3  1    228.56 3833.0 170.91
```

```
##
## Step:  AIC=170.37
## y ~ x3 + I(x1^2) + I(x2^2) + I(x3^2) + x1 + x2
##
##         Df Sum of Sq    RSS    AIC
## + x2:x3  1    540.28 3230.2 167.26
## + x1:x3  1    449.42 3321.1 168.18
## <none>               3770.5 170.37
## + x1:x2  1     43.34 3727.1 171.99
##
## Step:  AIC=167.27
## y ~ x3 + I(x1^2) + I(x2^2) + I(x3^2) + x1 + x2 + x3:x2
##
##         Df Sum of Sq    RSS    AIC
## <none>               3230.2 167.26
## + x1:x3  1    34.150 3196.0 168.91
## + x1:x2  1    20.106 3210.1 169.06
```

```
##
## Call:
## lm(formula = y ~ x3 + I(x1^2) + I(x2^2) + I(x3^2) + x1 + x2 +
##     x3:x2, data = kidney)
##
## Coefficients:
## (Intercept)           x3      I(x1^2)      I(x2^2)      I(x3^2)
##     0.11782      4.39664     15.56527      0.01176     -0.01520
##          x1           x2        x3:x2
##   -91.23842     -0.22790     -0.02271
```

As we can see the best subset of variables are x3, x1_2 ,x2_2, x3_2, x1, x2, x3_x2

b

```
# record the result in the last problem
result$which[60,]
```

```
##     1     2     3     4     5     6     7     8     9
##  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE
```

```
result$which[30,]
```

```
##     1     2     3     4     5     6     7     8     9
##  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
```

```
result$adjr2[30]
```

```
## [1] 0.8615103
```

```
result$adjr2[60]
```

```
## [1] 0.8623357
```