Problem 1

(a) We prove (10.12) for an arbitrarily given group C_k. suppose there are n observations, $x_1, x_2 \cdots x_n$ in C_k. Then (10.12) becomes:

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{p}(x_{ij}-x_{kj})^2 = 2\sum_{i=1}^{n}\sum_{j=1}^{p}(x_{ij}-\bar{x}_j)^2$$

where $\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$

$$\text{Left side} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{p}(x_{ij}-\bar{x}_j+\bar{x}_j-x_{kj})^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{p}\left[(x_{ij}-\bar{x}_j)^2+(x_{kj}-\bar{x}_j)^2\right] - \frac{2}{n}\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{p}(x_{ij}-\bar{x}_j)(x_{kj}-\bar{x}_j)$$

$$= \frac{1}{n}\left(\sum_{i=1}^{n}\sum_{j=1}^{p} n(x_{ij}-\bar{x}_j)^2 + \sum_{k=1}^{n}\sum_{j=1}^{p} n(x_{kj}-\bar{x}_j)^2\right] - \frac{2}{n}\sum_{j=1}^{p}\left[\sum_{i=1}^{n} x_{ij} - n\bar{x}_j\right]\left[\sum_{k=1}^{n} x_{kj} - n\bar{x}_j\right]$$

$$= \frac{1}{n}\cdot 2n \cdot \sum_{i=1}^{n}\sum_{j=1}^{p}(x_{ij}-\bar{x}_j)^2 \qquad - \frac{2}{n}\sum_{j=1}^{p}\cdot 0 \cdot 0$$

$$= 2\sum_{i=1}^{n}\sum_{j=1}^{p}(x_{ij}-\bar{x}_j)^2$$

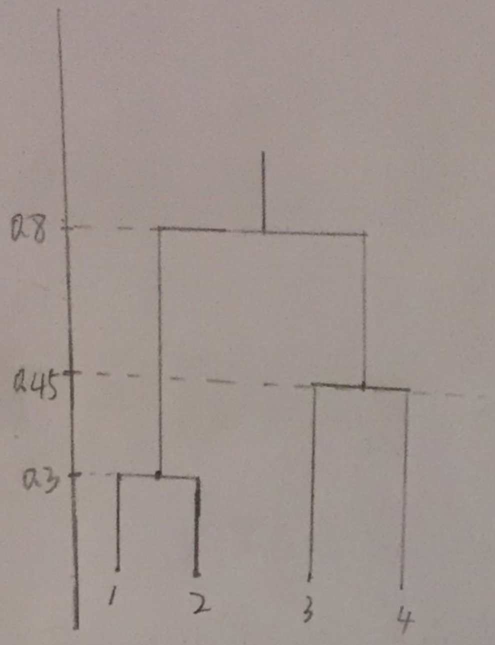
$= \text{right side}$

(b) For any given observation x, suppose it is originally assigned to group i, then in the next iteration, it is assigned to group j, this can only happen when: $\sum_{k=1}^{p}(x_k-\bar{x}_{jk})^2 \leq \sum_{k=1}^{p}(x_k-\bar{x}_{ik})^2$ because j's centroid $\bar{x}_j = (\bar{x}_{j1} \cdots \bar{x}_{jp})'$ is now the cloest to $x = (x_1 \cdots x_p)'$ according to algorithm 10.1 step (b)

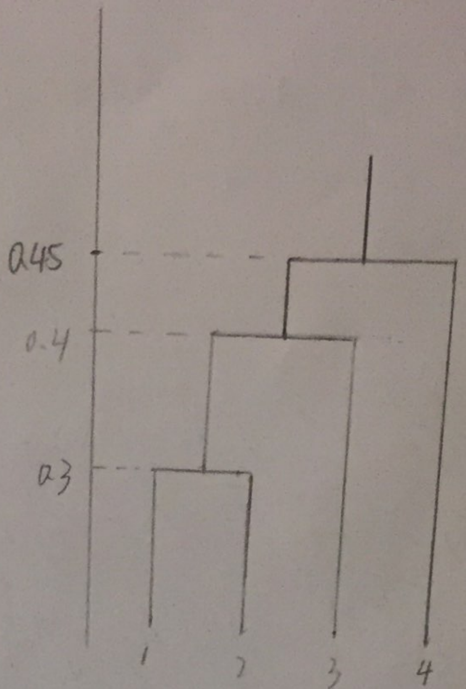
This holds true for all observations. Therefore according to 10.12, algorithm 10.1 decreases 10.11 at each iteration.

Problem 2

(a)



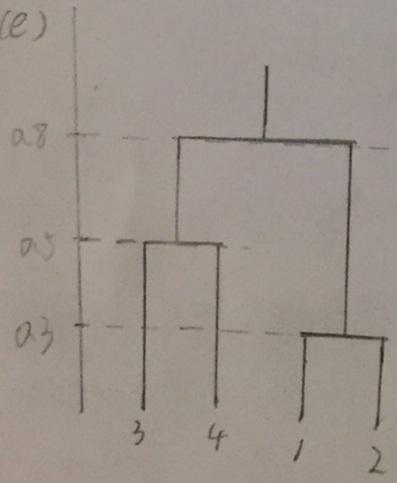
(b)



(c) 1,2 in cluster 1
3.4 in cluster 2

(d) 1,2,3 in cluster 1
4 in cluster 2

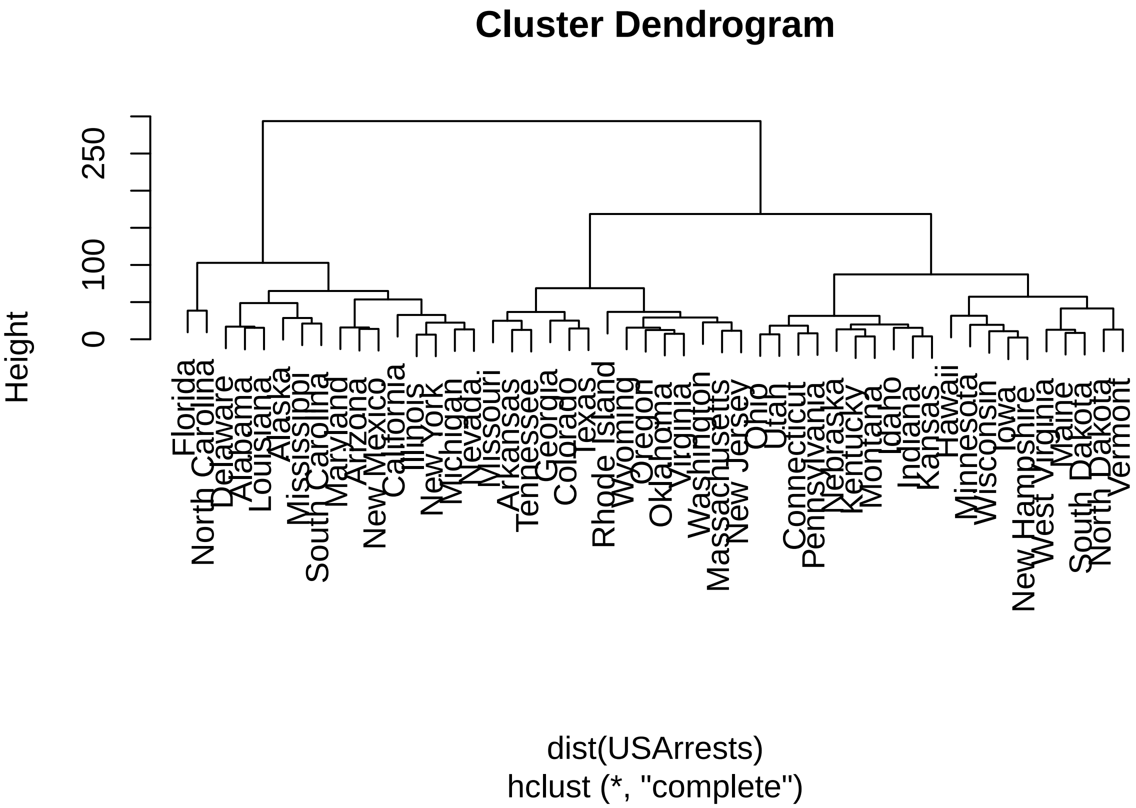
(e)



Problem 3

a)

```
hclust.complete=hclust(dist(USArrests),method='complete')
plot(hclust.complete)
```



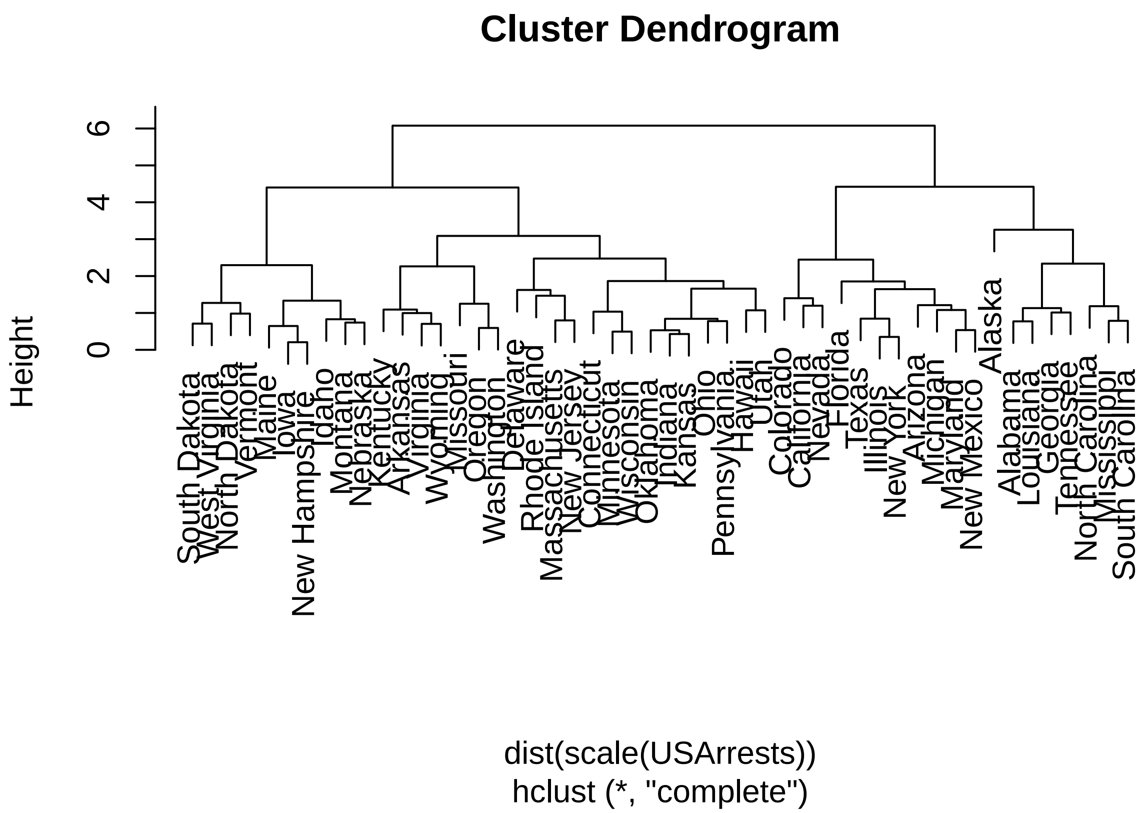
b)

```
re1=cutree(hclust.complete,3)
re1
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              1              2              1
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              2              3              1              1              2
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              3              1              3              3
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              3              3              1              3              1
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              2              1              3              1              2
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              3              3              1              3              2
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              1              1              1              3              3
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              2              2              3              2              1
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              3              2              2              3              3
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              2              2              3              3              2
```

c)

```
hclust.complete.sc=hclust(dist(scale(USArrests)),method='complete')
plot(hclust.complete.sc)
```



```
re2=cutree(hclust.complete.sc,3)
re2
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              2              3              2
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              2              3              3              2              1
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              3              2              3              3
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              3              3              1              3              2
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              3              2              3              1              3
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              3              3              2              3              3
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              2              2              1              3              3
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##               3              3              3              3              1
##    South Dakota      Tennessee          Texas           Utah        Vermont
##               3              1              2              3              3
##        Virginia     Washington  West Virginia      Wisconsin        Wyoming
##               3              3              3              3              3
```

d)

```
table(re1,re2)
```

```
##    re2
## re1  1  2  3
##   1  6  9  1
##   2  2  2 10
##   3  0  0 20
```

Scaling the variables indeed has effect on the output clusters. But the trees are still similar. We should scale the varibles in order to unify the data's measure.

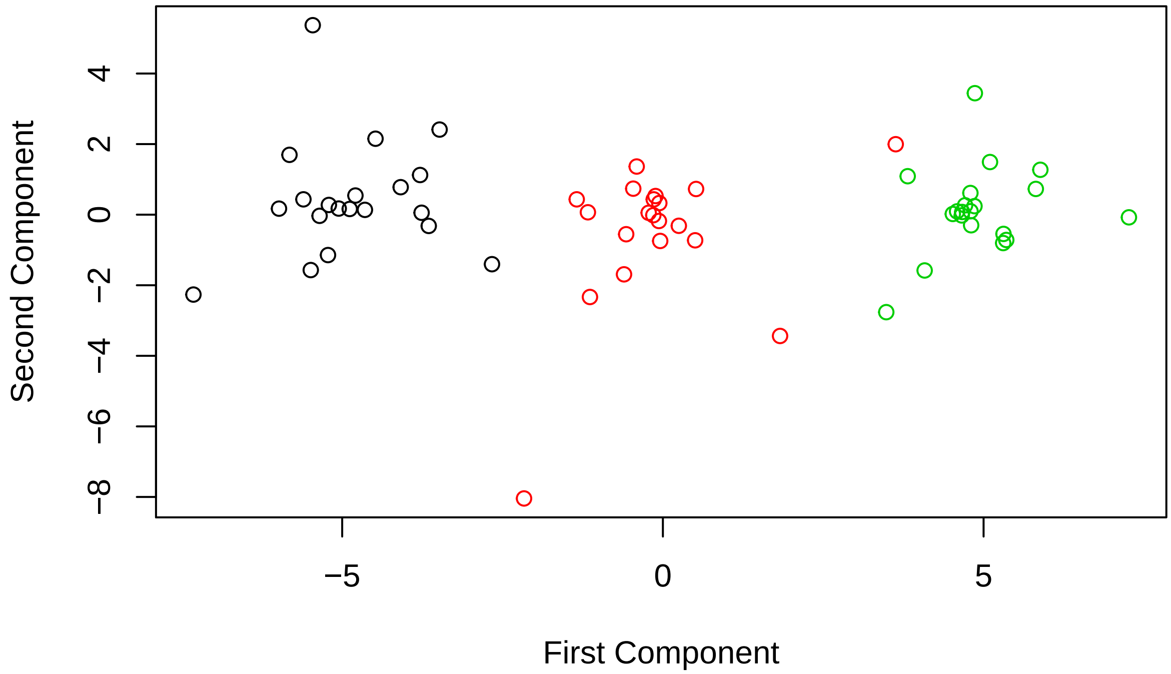
Problem 4

a)

```
set.seed(2)
data=matrix(0,ncol=50,nrow=60)
for(i in 1:20)
{
  data[i,]=rnorm(50,mean=1,sd=i/10)
}
for(i in 21:40)
{
  data[i,]=rnorm(50,mean=2,sd=(i-20)/10)
}
for(i in 41:60)
{
  data[i,]=rnorm(50,mean=3,sd=(i-40)/10 )
}
```

b)

```
pca=prcomp(data,scale=T)
first_comp=(pca$x)[,1]
second_comp=(pca$x)[,2]
plot(first_comp,second_comp,col=c(rep(1,20),rep(2,20),rep(3,20)),xlab='First Component',ylab='Second Co
```



c)

```
set.seed(2)
km=kmeans(data,3,nstart=30)
vec_true_label=c(rep(1,20),rep(2,20),rep(3,20))
table(km$cluster,as.factor(vec_true_label) )
```

```
##    
##      1  2  3
##   1 19  0  0
##   2  0  1 20
##   3  1 19  0
```

K-means is doing a really nice job in clustering the observations with only two wrong label.

d)

```
set.seed(2)
km=kmeans(data,2, nstart=30 )
vec_true_label=c(rep(1,20),rep(2,20),rep(3,20))
table(km$cluster,as.factor(vec_true_label) )
```

```
##    
##      1  2  3
##   1  0  2 20
##   2 20 18  0
```

K-means successfully seperates the true class 3 from the others while failing to seperate class 2 and 3 . That is it forms a cluster that consists of all 20 observations from class 3 and 2 observation from class 2. And all 20 observations in class 3 and 18 observations in class 2 got clustered together.

e)

```
set.seed(2)
km=kmeans(data,4, nstart=30 )
vec_true_label=c(rep(1,20),rep(2,20),rep(3,20))
table(km$cluster,as.factor(vec_true_label) )
```

```
##
##       1  2  3
##    1  1 19  0
##    2  0  1 19
##    3  0  0  1
##    4 19  0  0
```

K-means almost successfully seperates the 3 true class. And it also constructs a cluster with only 1 observation in it.

f)

```
set.seed(2)
km=kmeans(cbind(first_comp,second_comp),3, nstart=30 )
vec_true_label=c(rep(1,20),rep(2,20),rep(3,20))
table(km$cluster,as.factor(vec_true_label) )
```

```
##
##       1  2  3
##    1 19  0  0
##    2  0  1 20
##    3  1 19  0
```

Even with only 2 principle components, K-means is doing a really nice job in clustering the observations with only two wrong label. This shows that the first 2 principle components capture most of the information in the raw data.

g)

```
set.seed(2)
km=kmeans(scale(data),3,nstart=30)
vec_true_label=c(rep(1,20),rep(2,20),rep(3,20))
table(km$cluster,as.factor(vec_true_label) )
```

```
##
##       1  2  3
##    1 20  0  0
##    2  0  1 20
##    3  0 19  0
```

The result is slightly better with only 1 wrong label. This is because scaling the data gives each variable equally impact on the output of the model. This enhances the model's robustness against rare randomly-generated outliers in the observations.

Problem 5

(a) The cubic one's will be smaller
Because for the cubic one we have more predictors, hence we have a more flexible model. This will reduce the training error, namely RSS.

(b) The linear one's is smaller
because the real model is linear and the linear is fitting exactly the right thing while the cubic one fitted too much noise in the training data, it is too flexible in this content. Therefore it will have larger test error than the linear one.

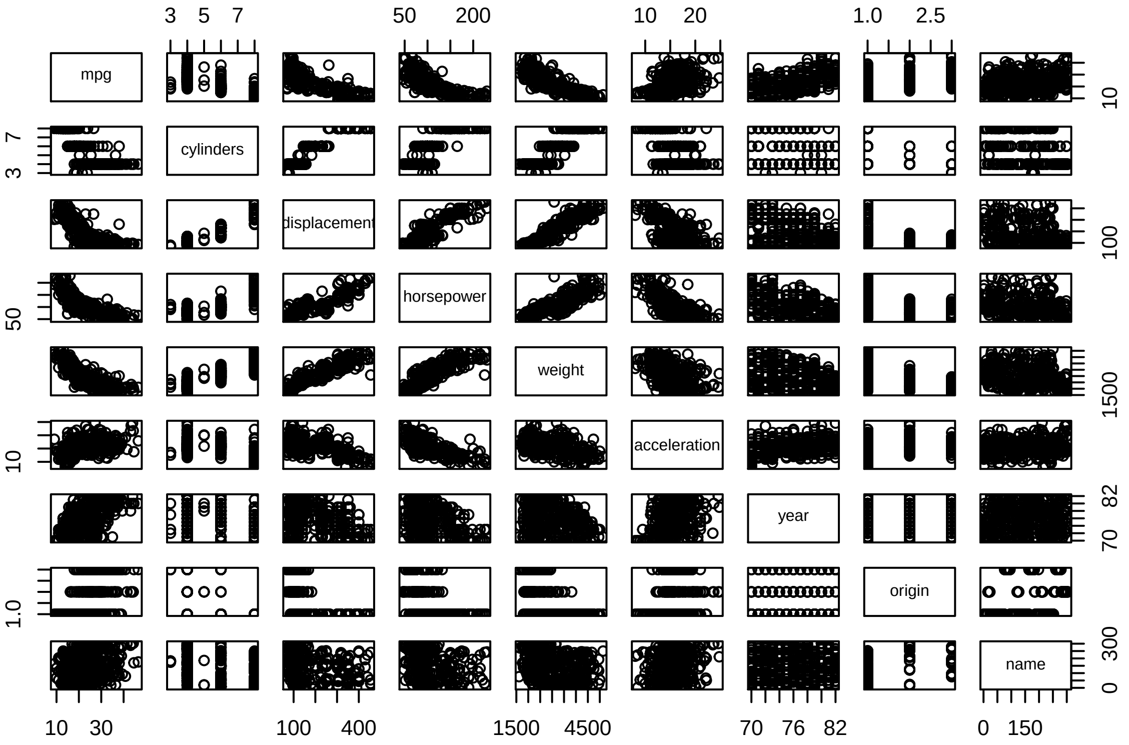
(c) The cubic one's will be smaller.
Also like (a), because the cubic one is more flexible, It has lower training error

(d) There's not enough information to tell.
If the true relationship is more close to linear, then the linear model will have lower test error, otherwise the cubic one's test errow will be lower.

Problem6

a)

```
library(ISLR)
plot(Auto)
```



b)

```
cor(Auto[1:8])
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

c)

```
lm.auto=lm(mpg~.-name,data=Auto)
summary(lm.auto)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

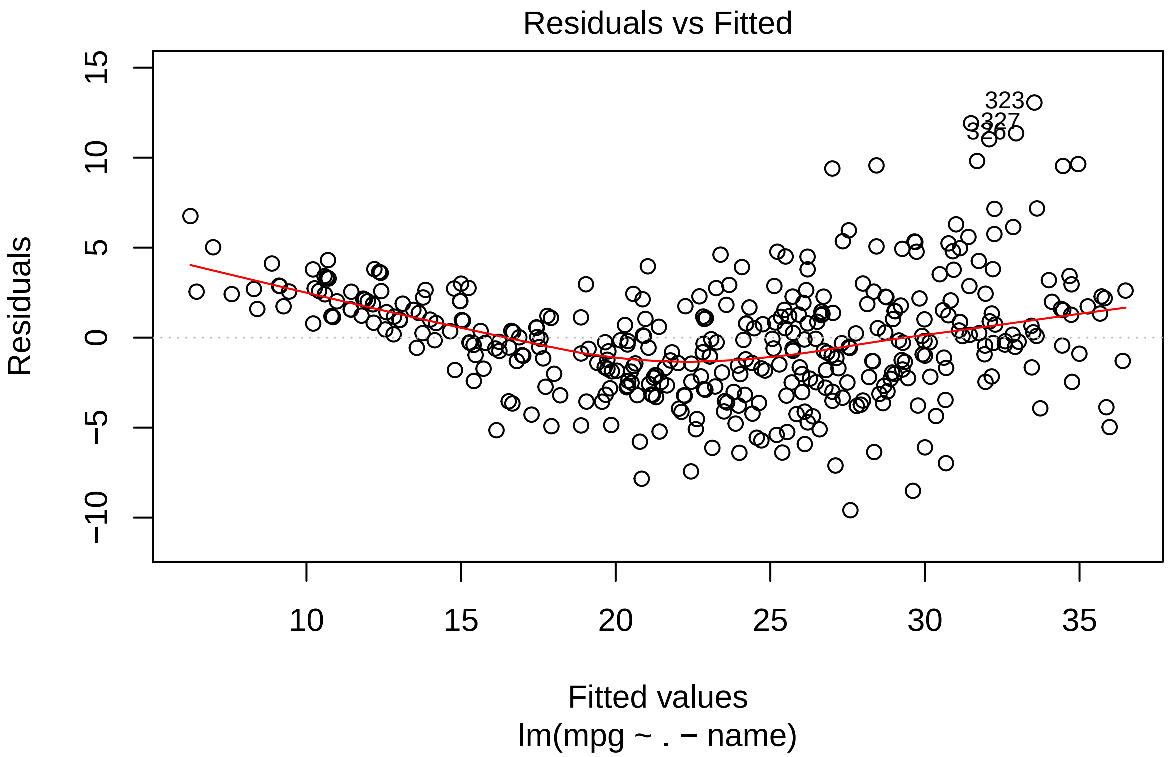
1.The whole model has a p value less than 2.2e-16, therefore there must be some predictors that have relationship with the response.

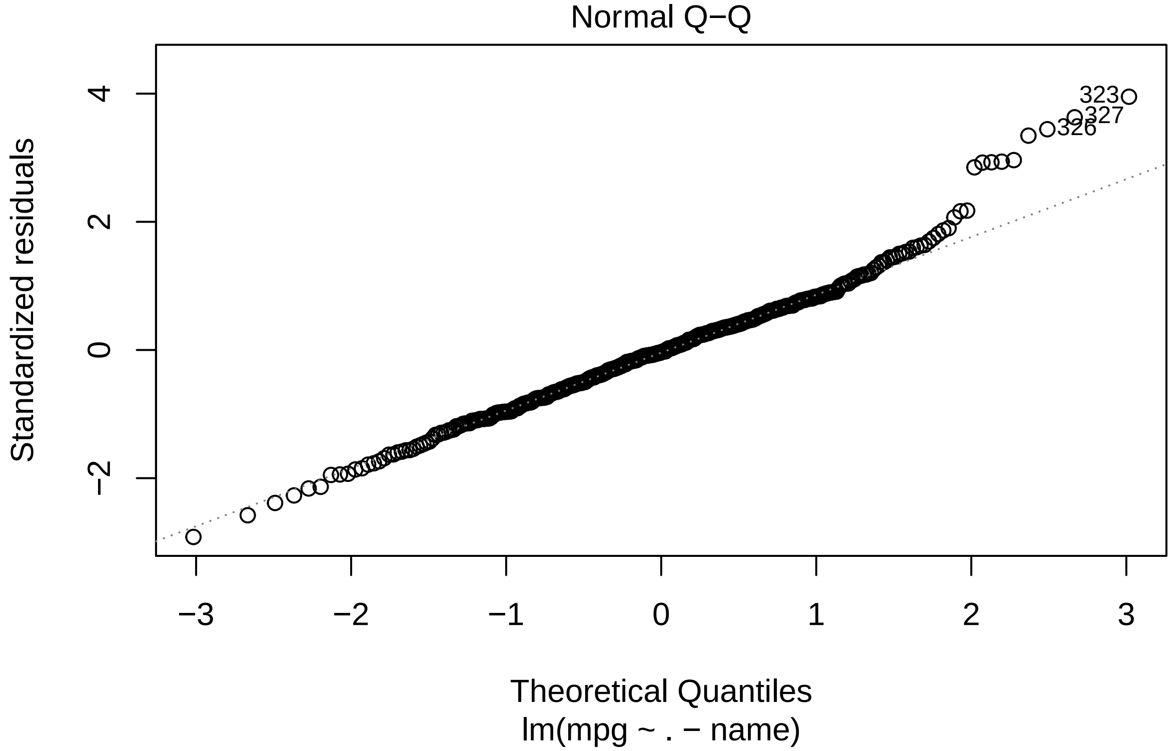
2.displacement,weight, year, origin have statistically significant relationship with the response.

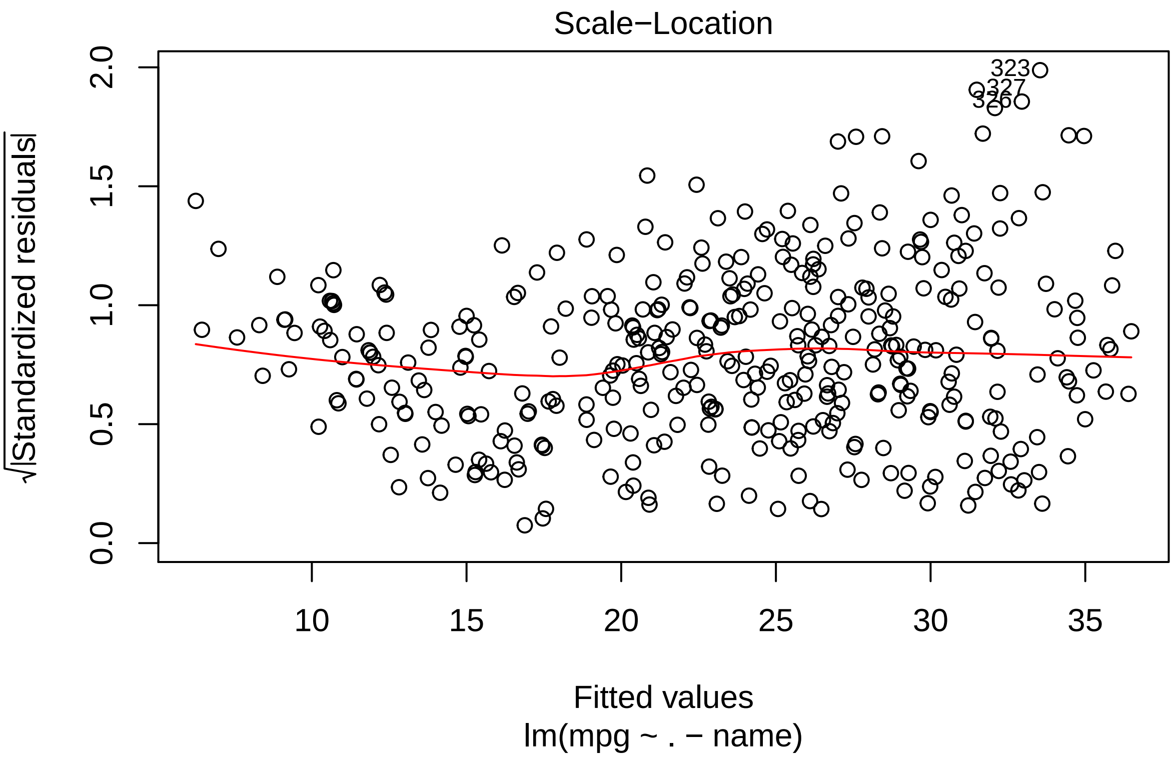
3.The coefficient of year suggests that if the average effect of year goes up 1 is that mpg will go up by 0.75.

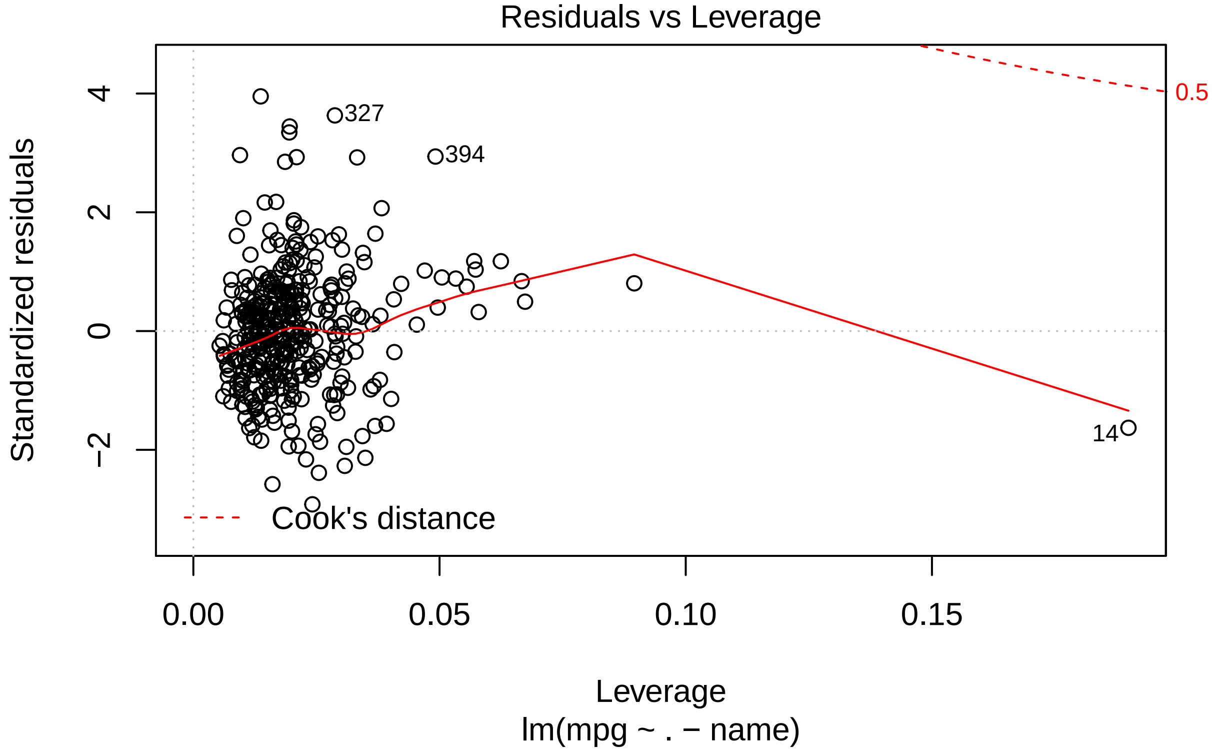
d)

```
plot(lm.auto)
```


Residuals vs Fitted
Residuals
15
10
5
0
−5
−10
323
327
326
10
15
20
25
30
35
Fitted values
lm(mpg ~ . − name)


Normal Q−Q
Standardized residuals
4
2
0
−2
323
327
326
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(mpg ~ . − name)





1.point 323, 326, 327 are unusually large outliers

2.point 14 has unusually large leverage

3.Accoring to the normal q-q plot, the residule is not nicely normal-distributed, it is somewhat right-skewed.

Problem 7

a)

```
set.seed(1)
x1=runif(100)
x2=0.5*x1+rnorm(100)/10
y=2+2*x1+0.3*x2+rnorm(100)
```

The linear model is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

The coefficients are:

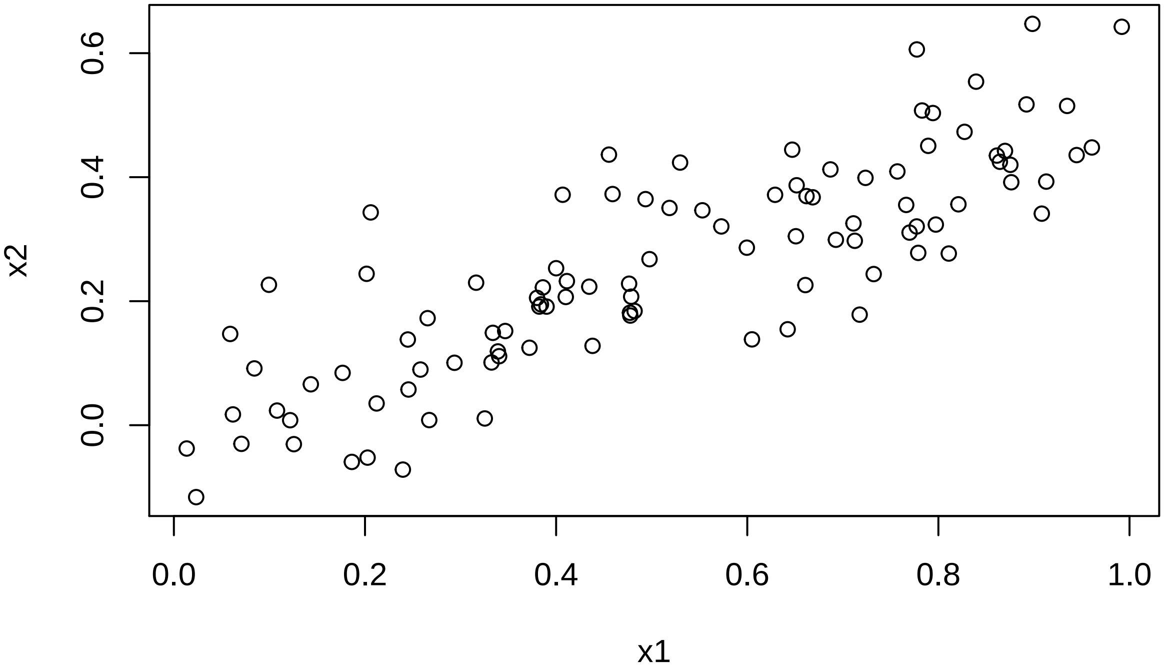
$$\beta_0 = 2, \beta_1 = 2, \beta_2 = 0.3$$

b)

```
cor(x1,x2)
```

```
## [1] 0.8351212
```

```
plot(x1,x2)
```



c)

```
lm.out1<-lm(y~x1+x2)
summary(lm.out1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

The estimators are:

$$\hat{\beta_0} = 2.1305, \hat{\beta_1} = 1.4396, \hat{\beta_2} = 1.0097$$

The β_0 is almost accurate but β_1 and β_2 are not. We can reject the null hypothesis $H_0 : \beta_1 = 0$ but we cannot reject the null hypothesis $H_0 : \beta_2 = 0$

d)

```
lm.out2<-lm(y~x1)
summary(lm.out2)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

Yes, we can reject the null hypothesis $H_0 : \beta_1 = 0$

e)

```r
lm.out3<-lm(y~x2)
summary(lm.out3)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

Yes, we can reject the null hypothesis $H_0 : \beta_2 = 0$

f) No, they don't. This is because in c), the fact that we can not reject $H_0 : \beta_2 = 0$ is in the presence one x1. What it means is that in the presence of x1, x2 provides no statistically significant additional information about y. While d) and e) say that x1 or x2 alone provide statistically information about y.

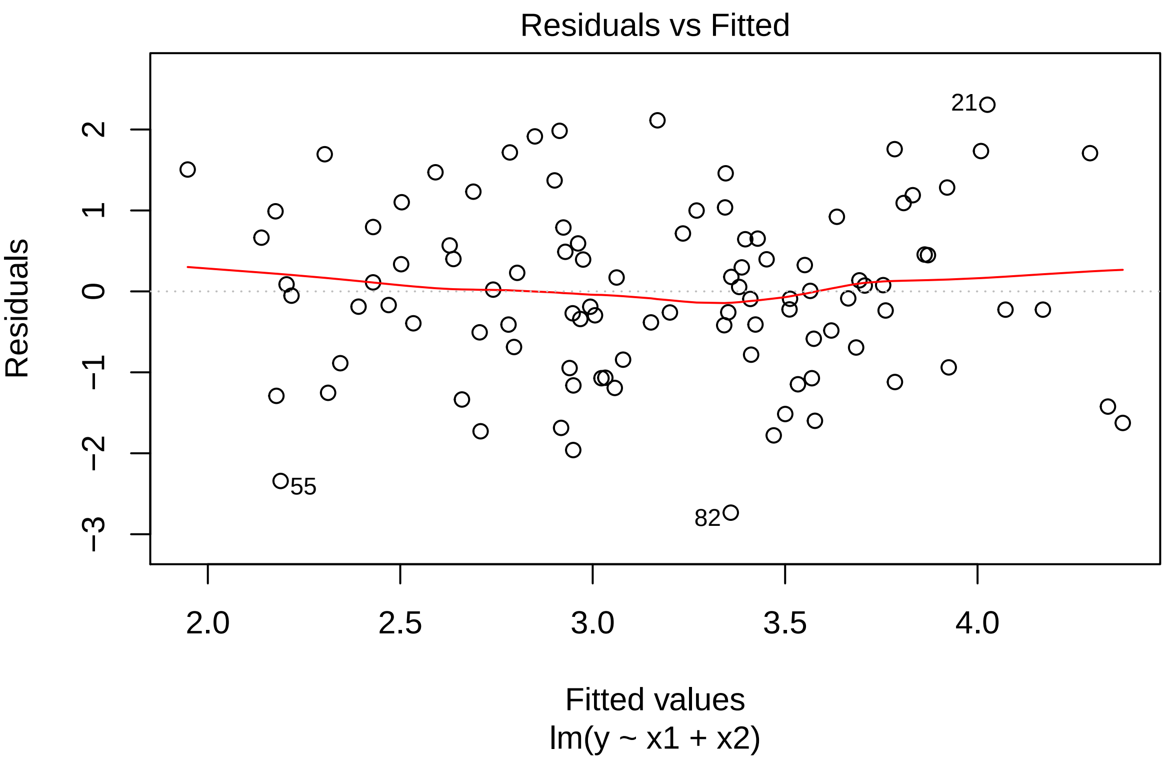
The reason why this is happening is that x1 and x2 are highly correlated. We have collinearity. Collinearity reduces the accuracy of the estimates of the regression coefficients.

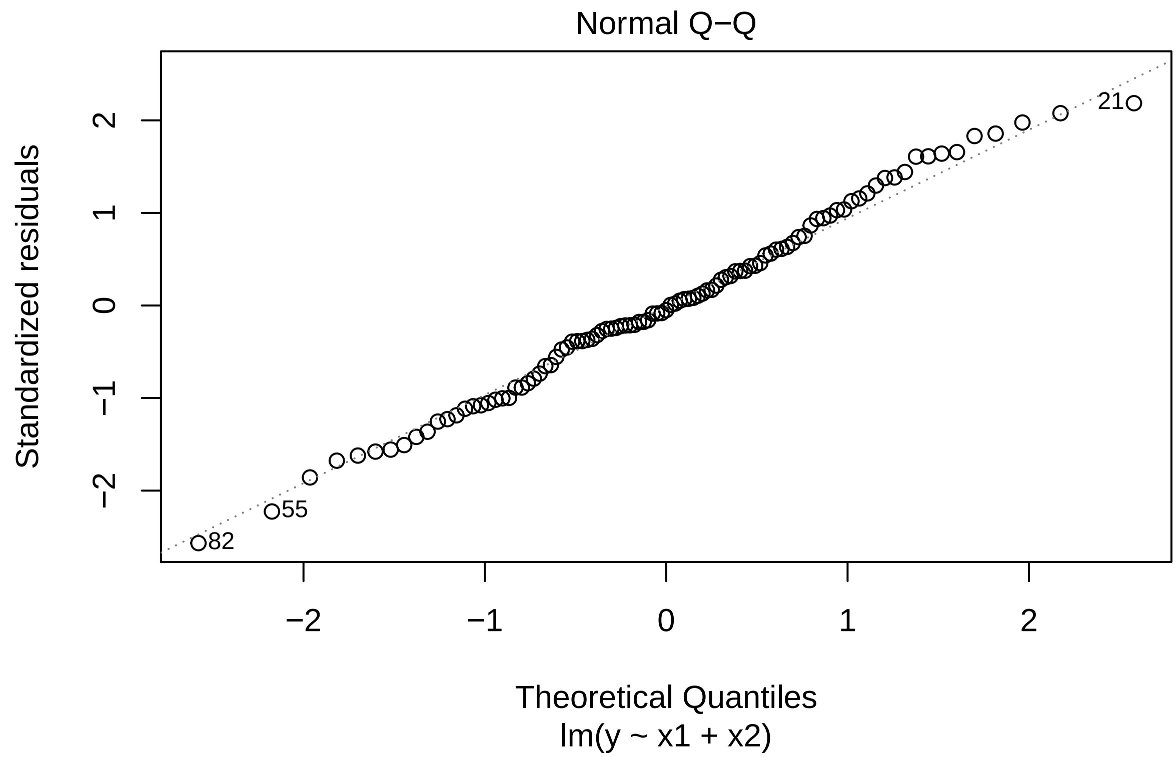
g)

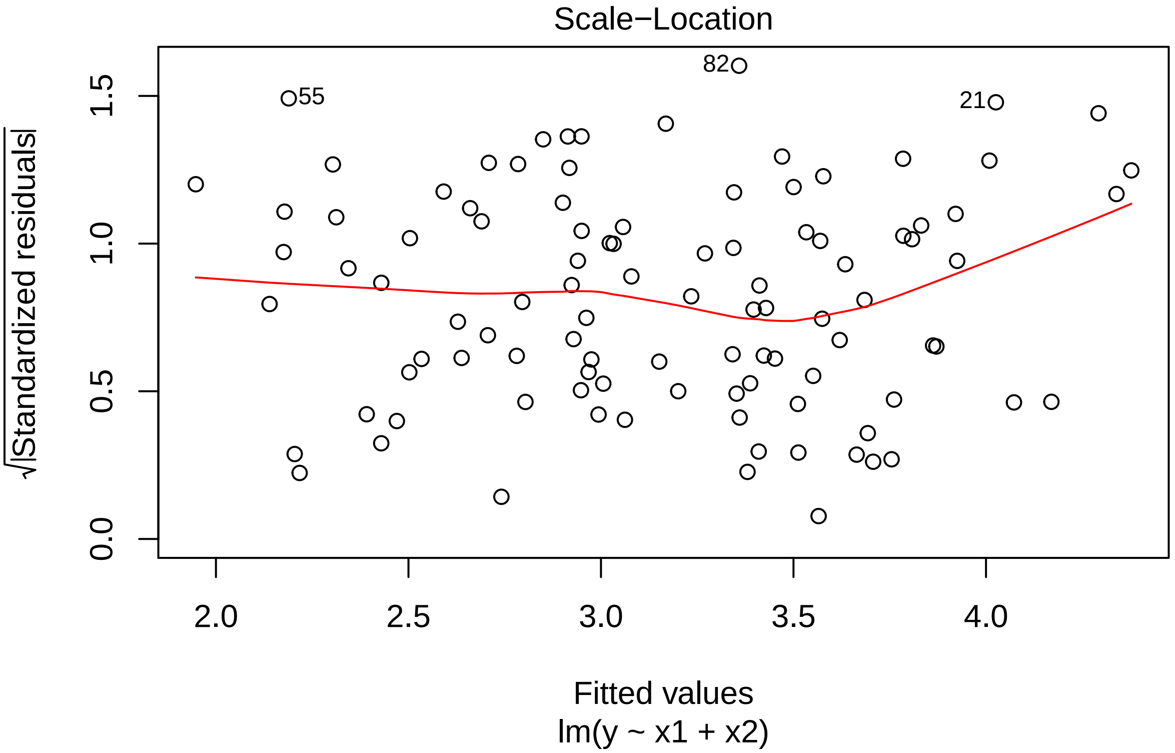
```r
x1=c(x1,0.1)
x2=c(x2,0.8)
y=c(y,6)
lm.out1<-lm(y~x1+x2)
summary(lm.out1)
```

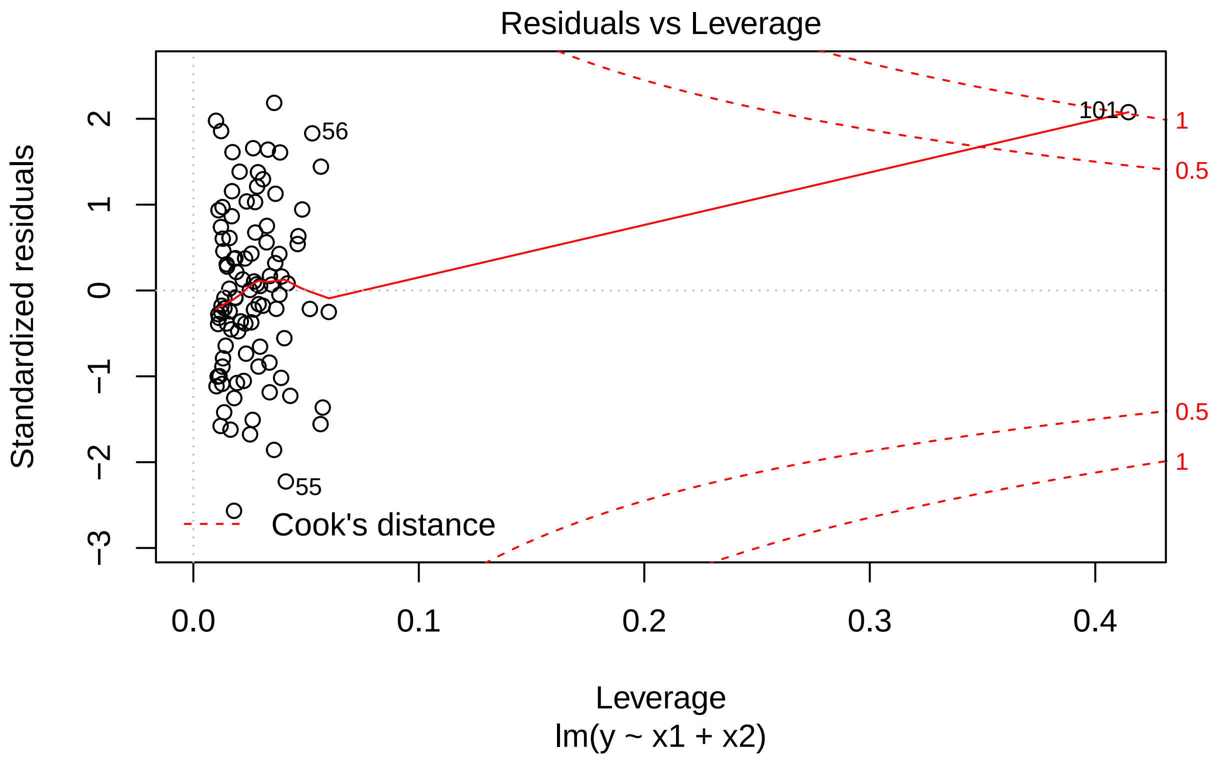
```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 5.564e-06
```

```r
plot(lm.out1)
```




Normal Q−Q
Standardized residuals
2
1
0
−1
−2
21
55
82
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x1 + x2)

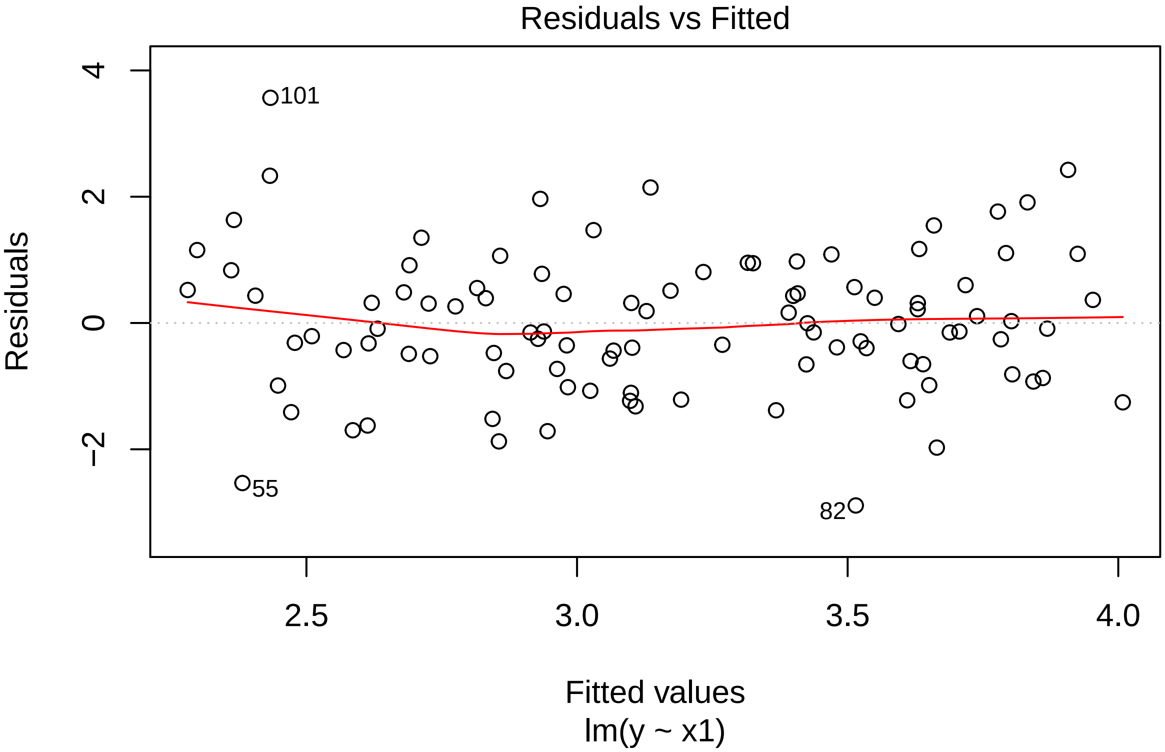

Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
55
82
21
2.0
2.5
3.0
3.5
4.0
Fitted values
lm(y ~ x1 + x2)

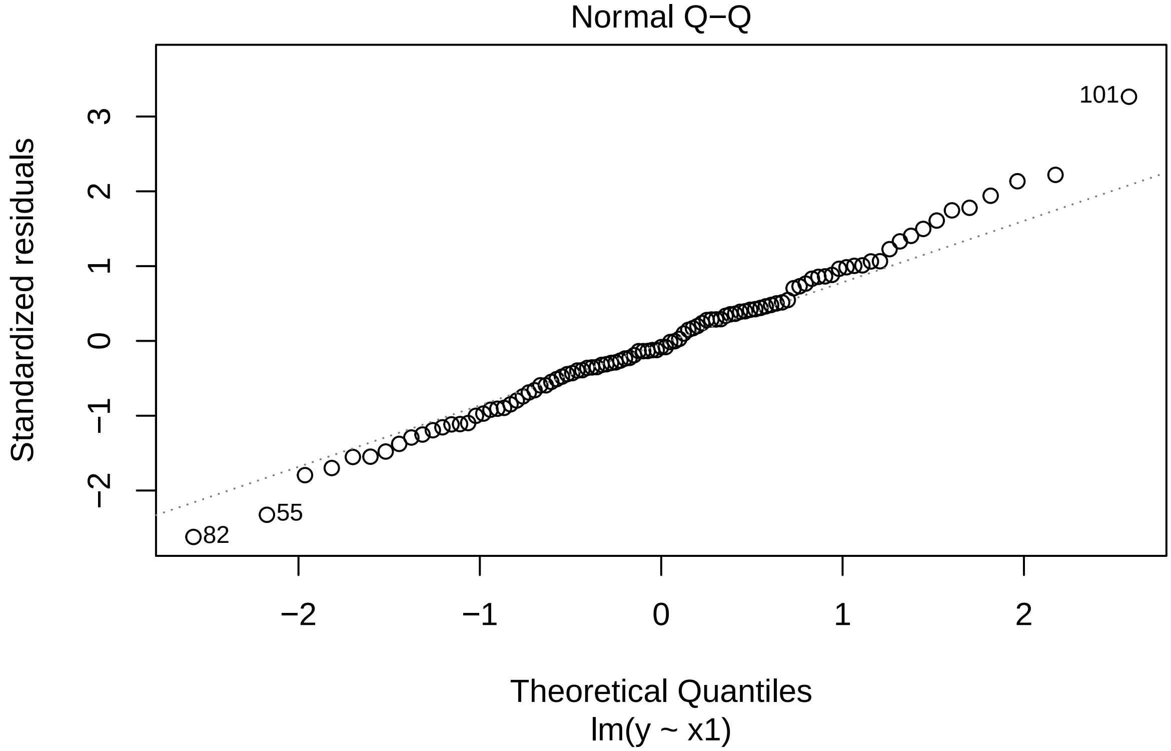


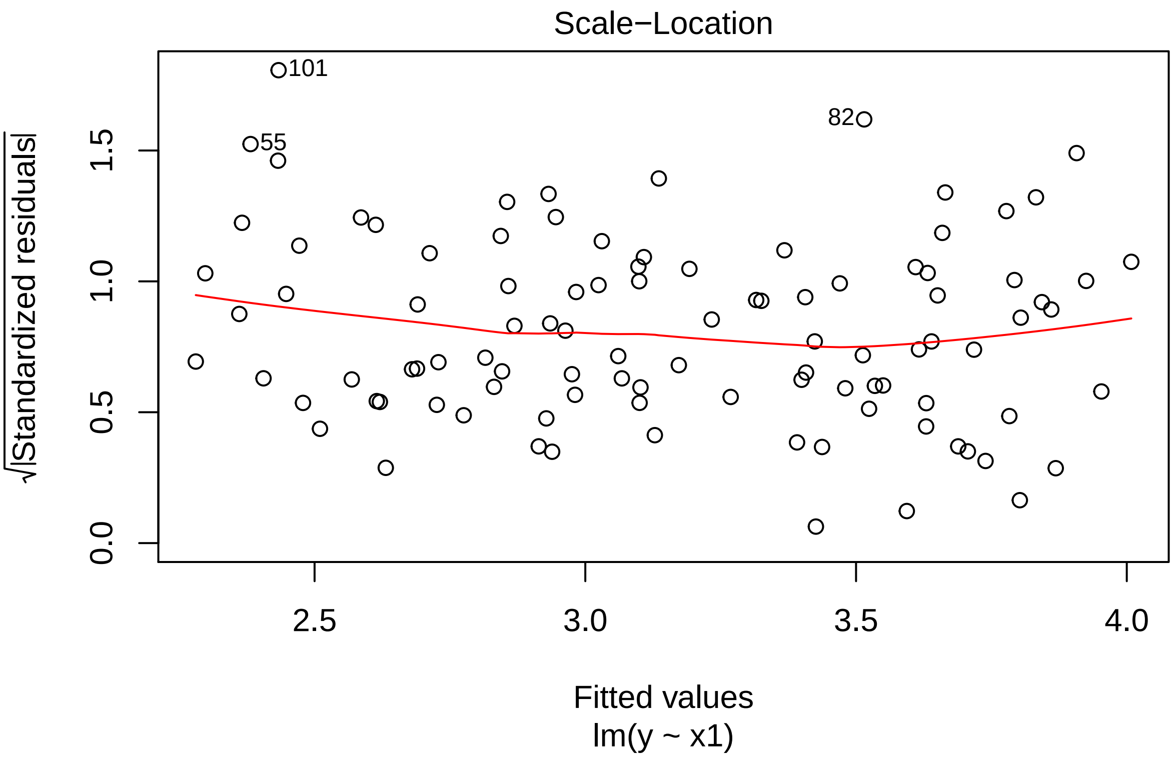
```
lm.out2<-lm(y~x1)
summary(lm.out2)
```

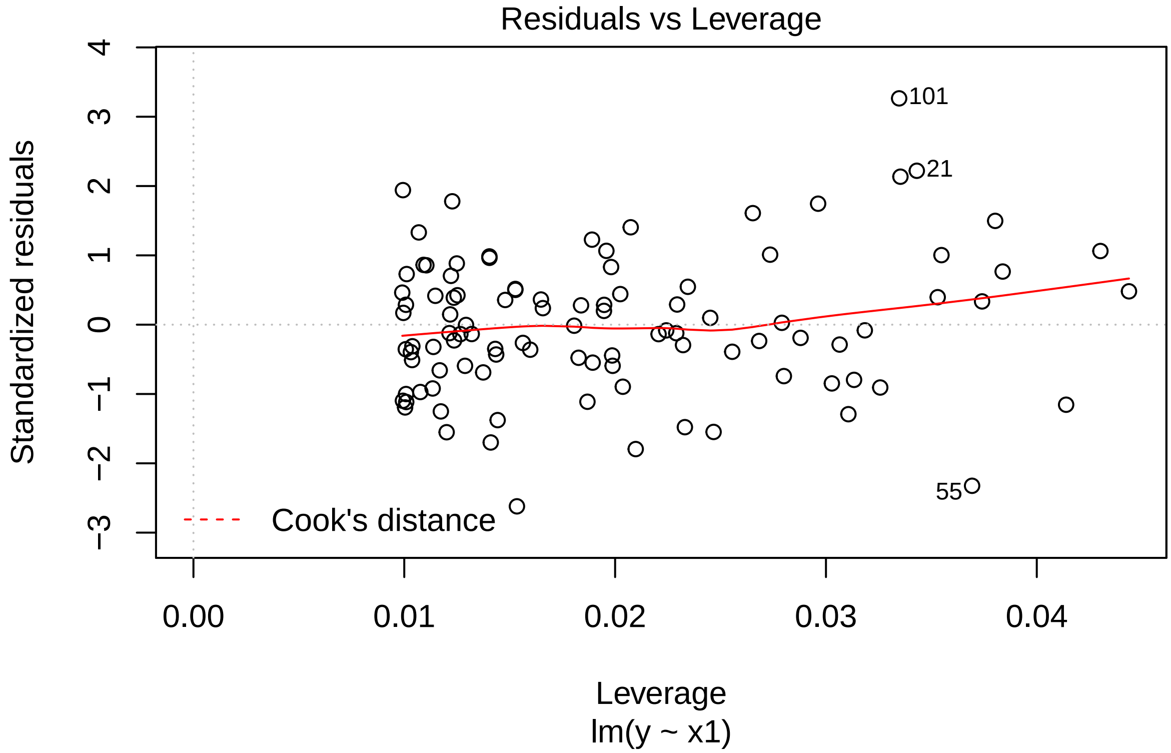
```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1            1.7657     0.4124   4.282 4.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.111 on 99 degrees of freedom
## Multiple R-squared:  0.1562, Adjusted R-squared:  0.1477
## F-statistic: 18.33 on 1 and 99 DF,  p-value: 4.295e-05
```

```
plot(lm.out2)
```


Residuals vs Fitted
Residuals
4
2
0
−2
101
55
82
2.5
3.0
3.5
4.0
Fitted values
lm(y ~ x1)


Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
101
55
82
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x1)



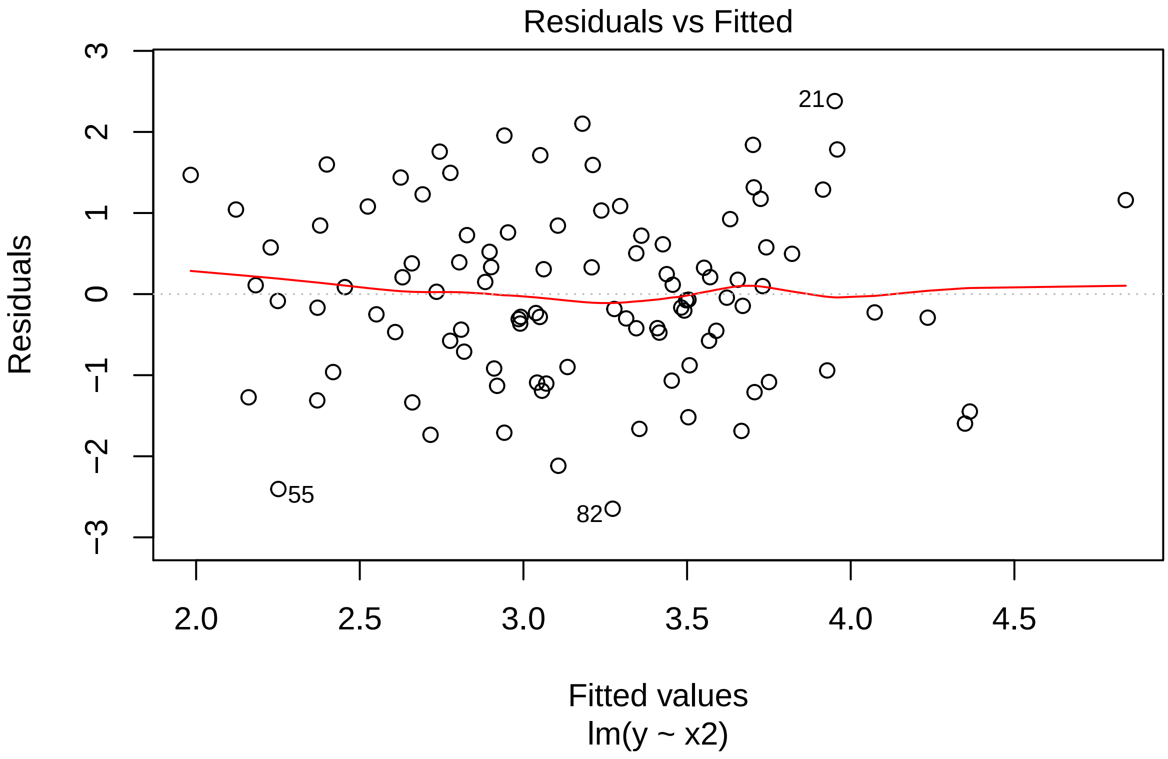


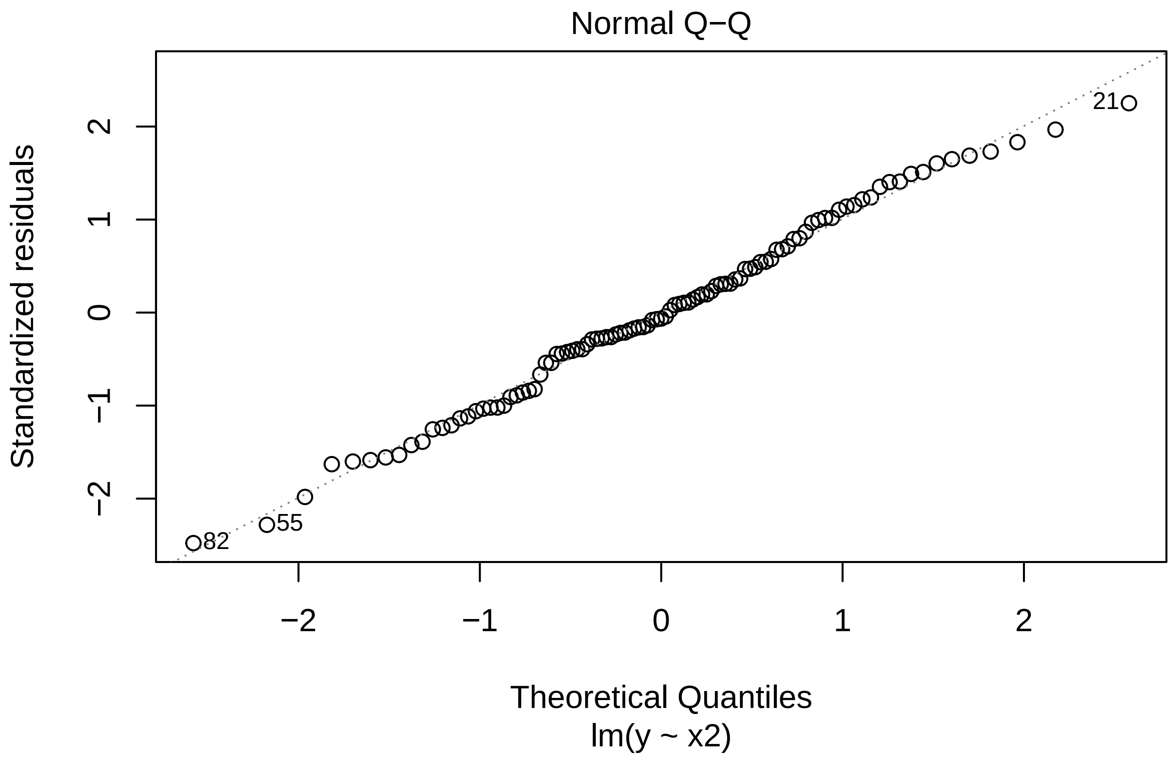
```
lm.out3<-lm(y~x2)
summary(lm.out3)
```

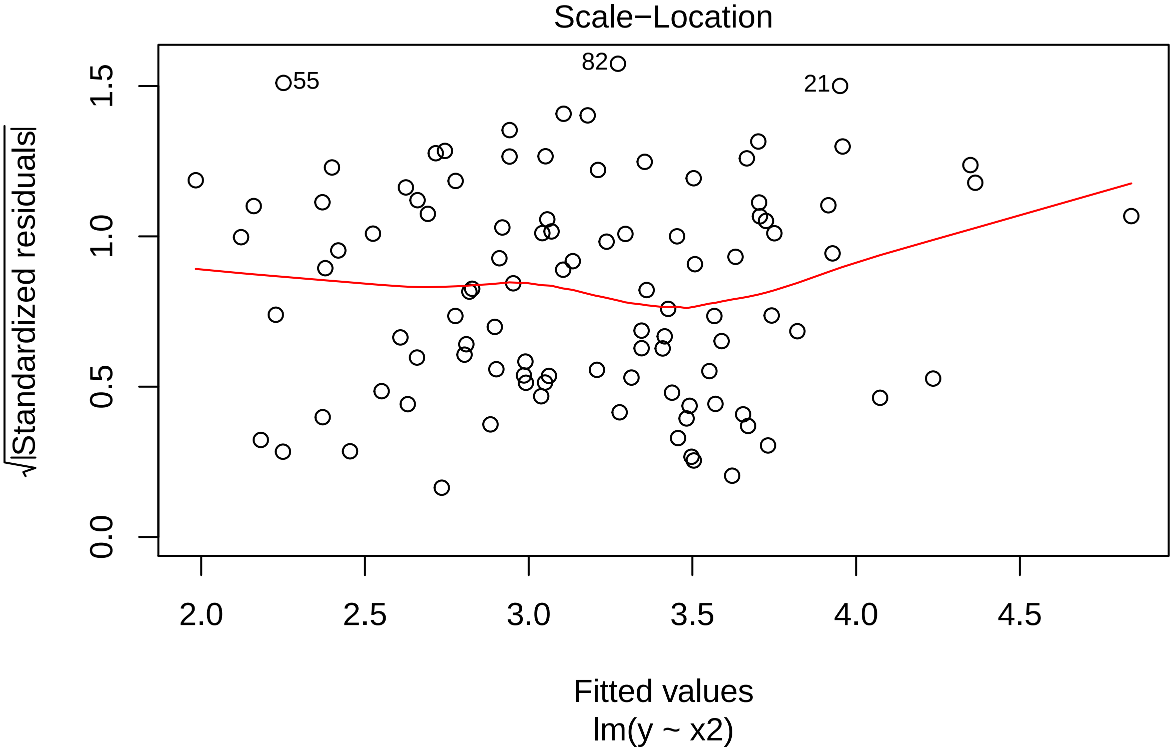
```
##
## Call:
```

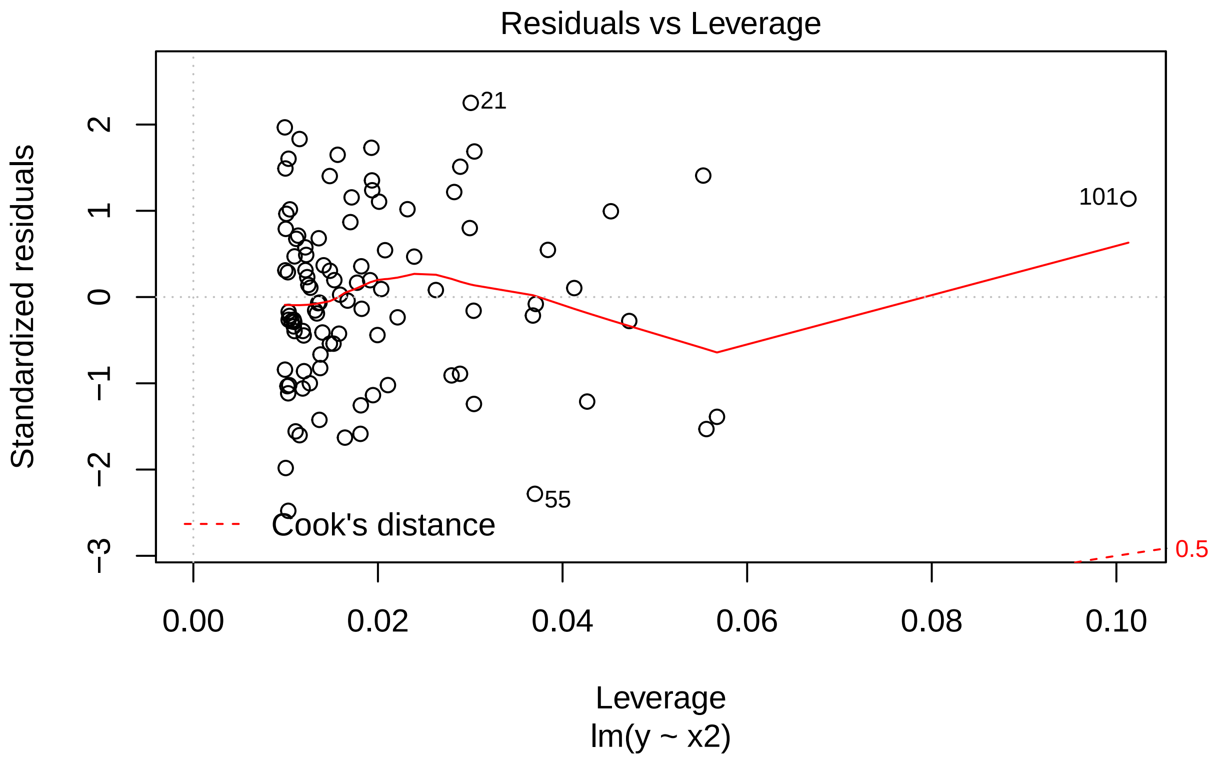
```
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264  < 2e-16 ***
## x2            3.1190     0.6040   5.164 1.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 99 degrees of freedom
## Multiple R-squared:  0.2122, Adjusted R-squared:  0.2042
## F-statistic: 26.66 on 1 and 99 DF,  p-value: 1.253e-06
```

```
plot(lm.out3)
```




Normal Q−Q
Standardized residuals
−2
−1
0
1
2
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x2)
21
55
82


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
Fitted values
lm(y ~ x2)
55
82
21



This point is a high-leverage point in the model with both x1 and x2; it is both an outlier and a high-leverage point in the model with only x1; it is a high-leverage point in the model with only x2.

Problem 8

a)

```
library(MASS)
attach(Boston)
name=names(Boston)
single_coef=rep(0,13)
for(i in 2:14)
{
  print(paste('result for ',name[i],sep=''))
  lm.fit=lm(crim~Boston[,i],data=Boston)
  print(summary(lm.fit))
  single_coef[i-1]=lm.fit$coefficients[2]
}
```

zn, indus, nox, rm, age, dis, rad, tax, ptratio, black, lstat, medv have statistically significant association with crim.

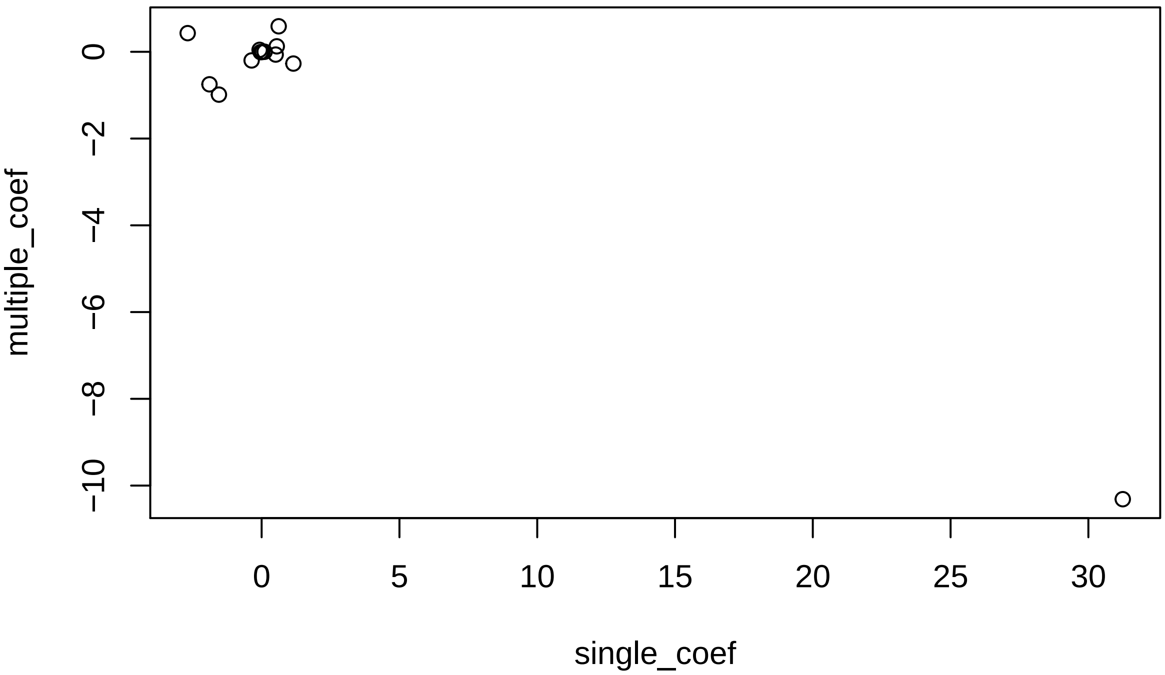
b)

```
lm.fit=lm(crim~.,data=Boston)
summary(lm.fit)
multiple_coef=lm.fit$coefficients[2:14]
```

For zn, dis, rad,black and medv, we can reject the null hypothesis $H_0 : \beta_j = 0$

c) Some of the predictors that are previously significant in a) are no-longer significant in b)

```
plot(x=single_coef,y=multiple_coef)
```



d)

```
for(i in c(2:14))
{
  print(paste('result for ',name[i],sep=''))
  lm.fit=lm(crim~poly(Boston[,i],3,raw=T),data=Boston)
  print(summary(lm.fit))
}
```

For indus, nox, age, dis, ptratio, medv, they have non-linear association with the response crim.