Homework 1

Wentao Zhu

1

P52, 2.4.2

a

Regression. Inference. n = 500 (firms), p = 3 (profit, number of employees, industry).

b

Classification. Prediction. n = 20 (similar products previously launched). p = 13 (price charged, marketing budget, competition price, ten other variables).

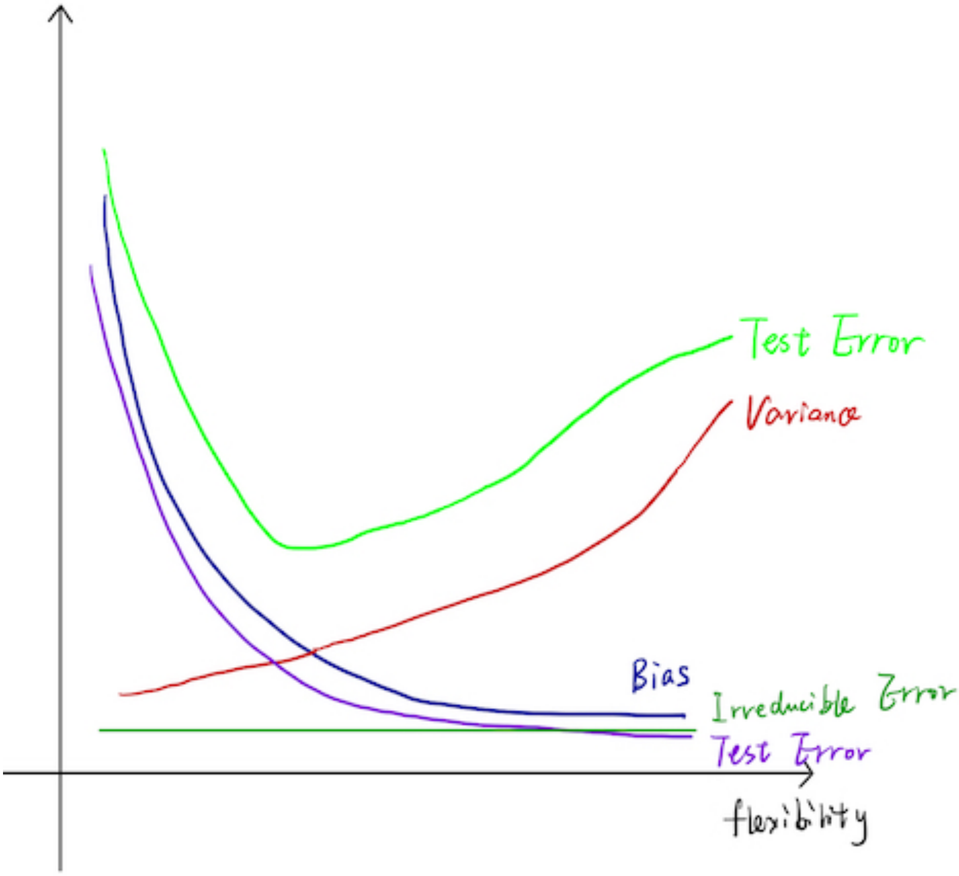
c

Regression. Prediction. n = 52 (weekely change of dollar in 2012). p = 3(% change in US/German/British market).

2

P52, 2.4.3

a



b

- **Bias**: decreases as the method's flexibility increases because of it has less constraints.
- **Variance**: increases as the method's flexibility increases because the model relies on the input data more.
- **Training Error**: decreases as the method's flexibility increases because the more flexible model makes the model fit the training data better.
- **Test Error**: decreases first, and then increases. Increases in flexibility generates a closer fit before overfitting.
- **Irreducible Error**: is the same regardless of the model. It depends on the distribution of ϵ

3

P53, 2.3.7

a

```
Obs = matrix(data=c(0,2,0,0,-1,1,3,0,1,1,0,1,0,0,3,2,1,1), nrow=6, ncol=3)
Pred <- c(0,0,0)
for (i in 1:6) {
   print(sqrt(sum((Obs[i,]-Pred)^2)))
}
```

```
## [1] 3
## [1] 2
## [1] 3.162278
## [1] 2.236068
## [1] 1.414214
## [1] 1.732051
```

b

Green. The nearest neighbor is `Obs[5]`, which is green.

c

Red. The 3-nearest neighbors are `Obs[5]`, `Obs[6]`, `Obs[2]`, which are green, red, red.

d

Small. A small K would be able to capture more local non-linear decision information.

4

P413, 10.7.1

a

$$
\begin{aligned}
\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 &= \frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}((x_{ij}-\bar{x}_{kj})-(x_{i'j}-\bar{x}_{kj}))^2 \\
&= \frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}((x_{ij}-\bar{x}_{kj})^2-2(x_{ij}-\bar{x}_{kj})(x_{i'j}-\bar{x}_{kj})+(x_{i'j}-\bar{x}_{kj})^2) \\
&= \sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2+\sum_{i'\in C_k}\sum_{j=1}^{p}(x_{i'j}-\bar{x}_{kj})^2-\frac{2}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})(x_{i'j}-\bar{x}_{kj}) \\
&= 2\sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2
\end{aligned}
$$

b

From (a), we have:

$$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2$$

To minimize $\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2$, we only need to minimize $\sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2$.

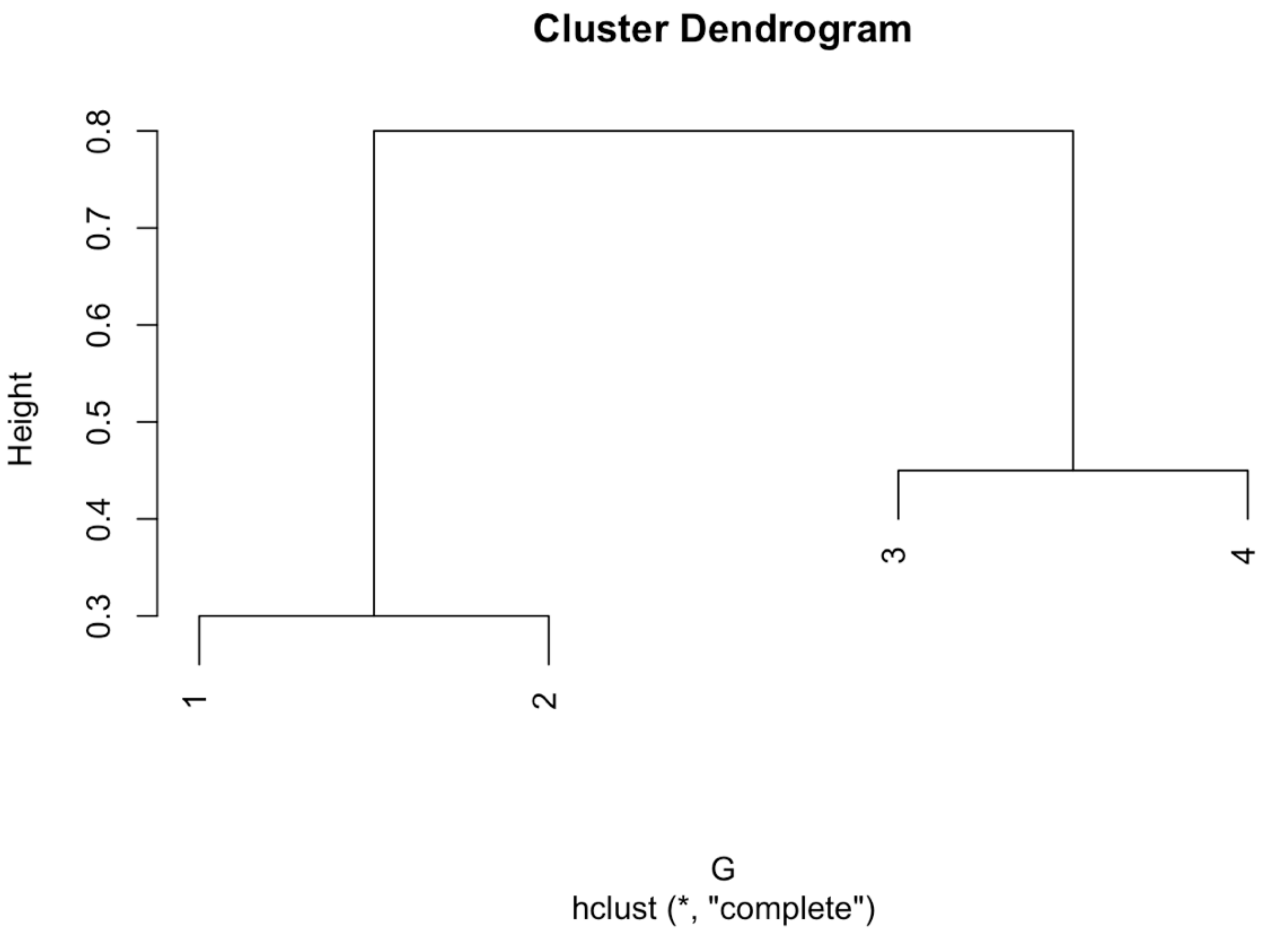
In every round of iteration, $\sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$ is minimized by definition (assigning every point to the closest cluster centroid).

5

P413, 10.7.2

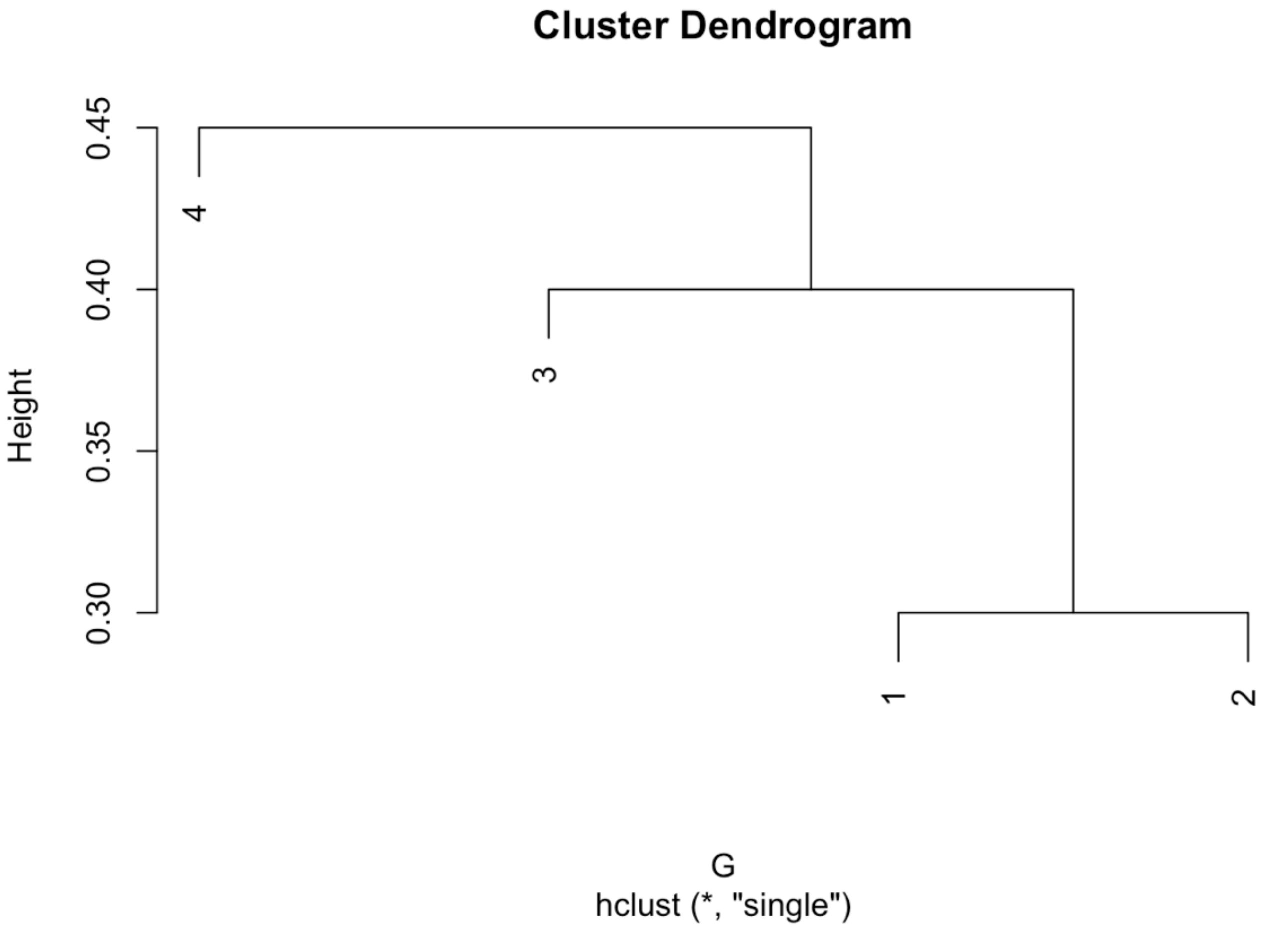
a

```
G = as.dist(matrix(c(0, 0.3, 0.4, 0.7,
                     0.3, 0, 0.5, 0.8,
                     0.4, 0.5, 0.0, 0.45,
                     0.7, 0.8, 0.45, 0.0), nrow=4, ncol=4))
plot(hclust(G, method="complete"))
```



b

```
plot(hclust(G, method="single"))
```



c

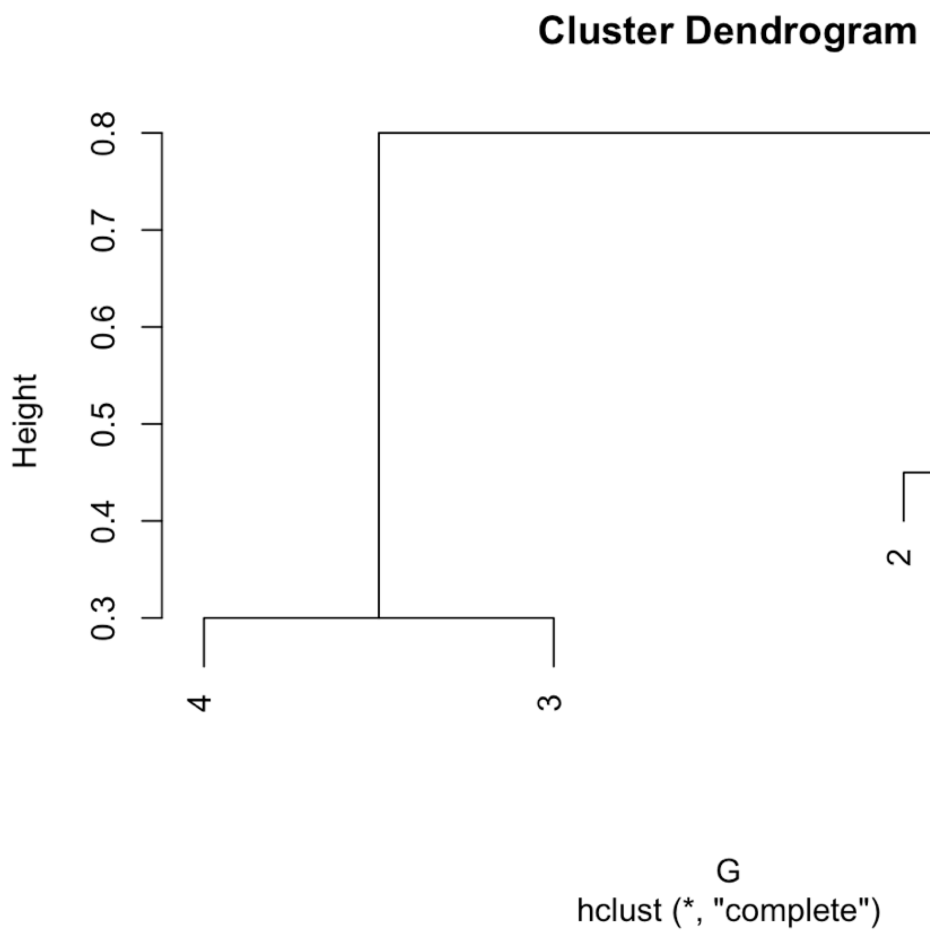
- 1,2
- 3,4

d

- 1,2,3
- 4

e

```
plot(hclust(G, method="complete"), labels=c(4,3,2,1))
```



6

P414, 10.7.4

a

Not enough information to tell. It depends on the exact average distance and minimum distance of two clusters. If the two distances are equal, they would fuse at the same height. Else the single linkage dendogram would fuse at a lower height.

b

Same. Height of fusions of leaf nodes are not influenced by the linkage method.

7

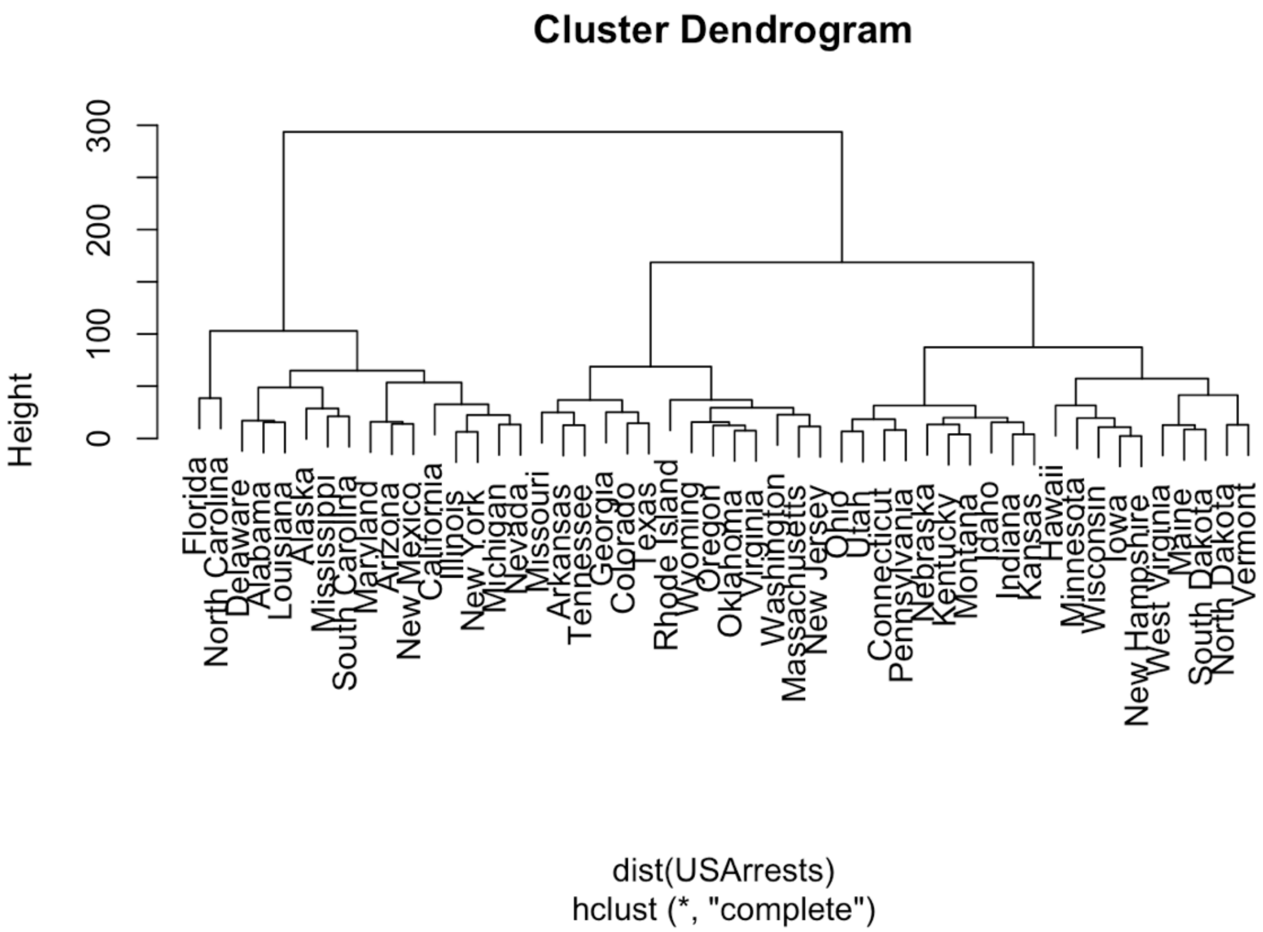
P416, 10.7.9

a

```
library(ISLR)
```

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

```
original = hclust(dist(USArrests), method="complete")
plot(original)
```



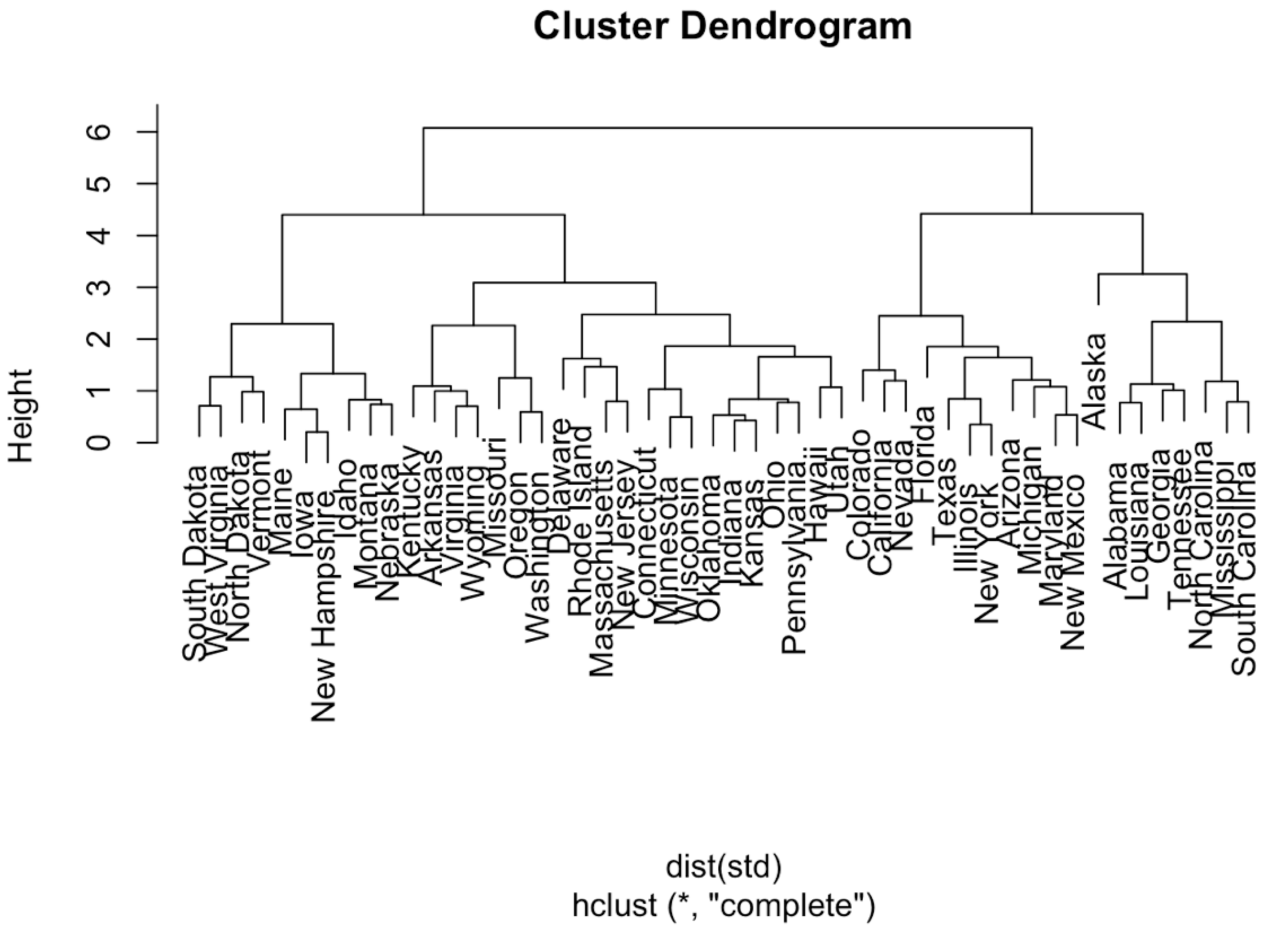
b

```
original_result = cutree(original, 3)
original_result
```

```
##        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

```
std = scale(USArrests)
scaled = hclust(dist(std), method="complete")
plot(scaled)
```



d

```
scaled_result = cutree(scaled, 3)
scaled_result
```

```
##        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
```

```
table(original_result, scaled_result)
```

```
##                scaled_result
## original_result  1  2  3
##               1  6  9  1
##               2  2  2 10
##               3  0  0 20
```

Though the dendogram seems alike for two methods, the clustering results are quite different. I think the dataset should be scaled before performing clustering because the metrics are easily influenced by the units adopted. In this dataset particularly, *UrbanPop* is different from other 3 coloumn from the perspective of unit.

```
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

8

P120 3.7.4

a

We could expect the cubic regression to have a lower training RSS than the linear regression for it has more flexibility and produces a tighter fit (though maybe meaningless).

b

The test RSS of cubic regression fit could be higher than the linear one for excessive predictors lead to overfitting.

c

We could always expect the cubic regression to have a lower training RSS than the linear regression for it has more flexibility and produces a tighter fit (regardless of what the true relationship is).

d

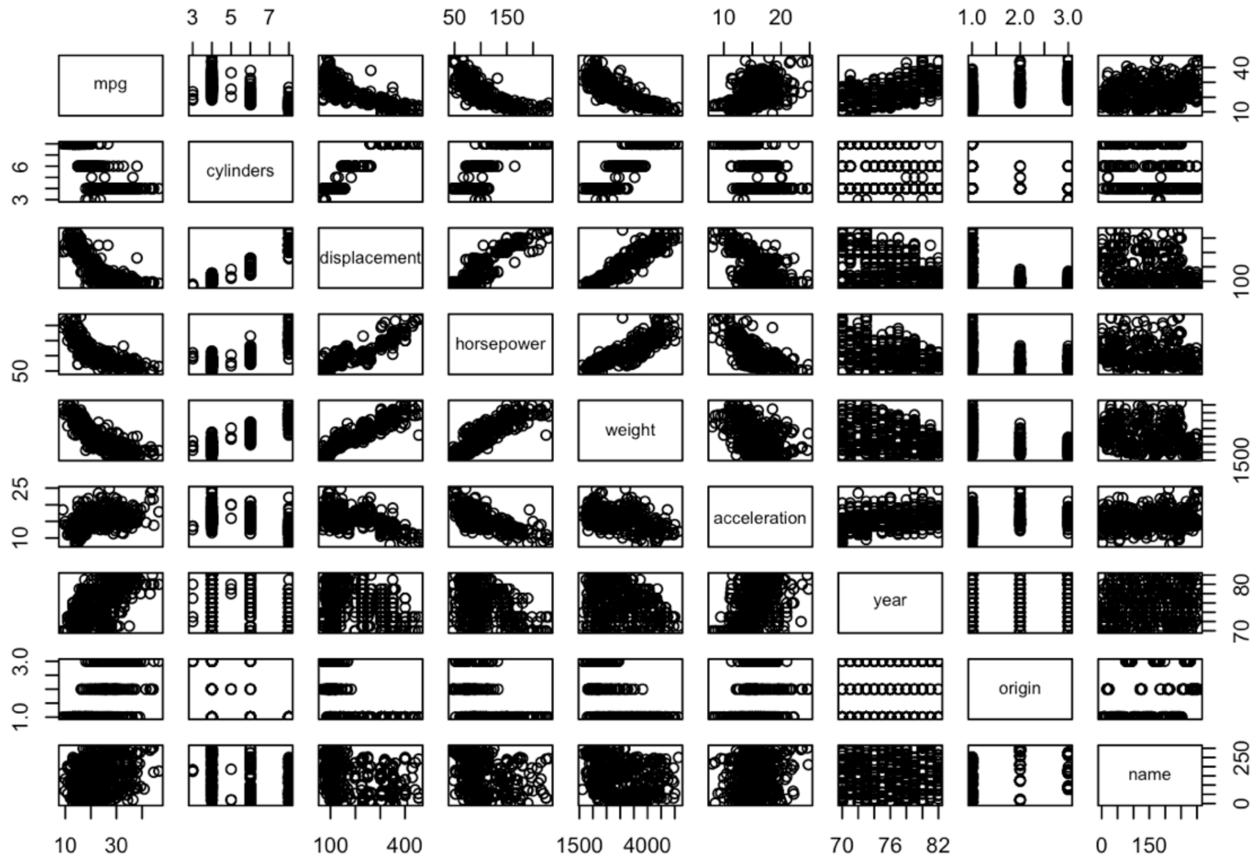
There is not enough information to tell. The result generally depends on whether the underlying relationship is more close to linear or cubic.

9

P122 3.7.9

a

```r
library(ISLR)
data(Auto)
pairs(Auto)
```



b

```
cor(subset(Auto, select=-name))
```

```
##                     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

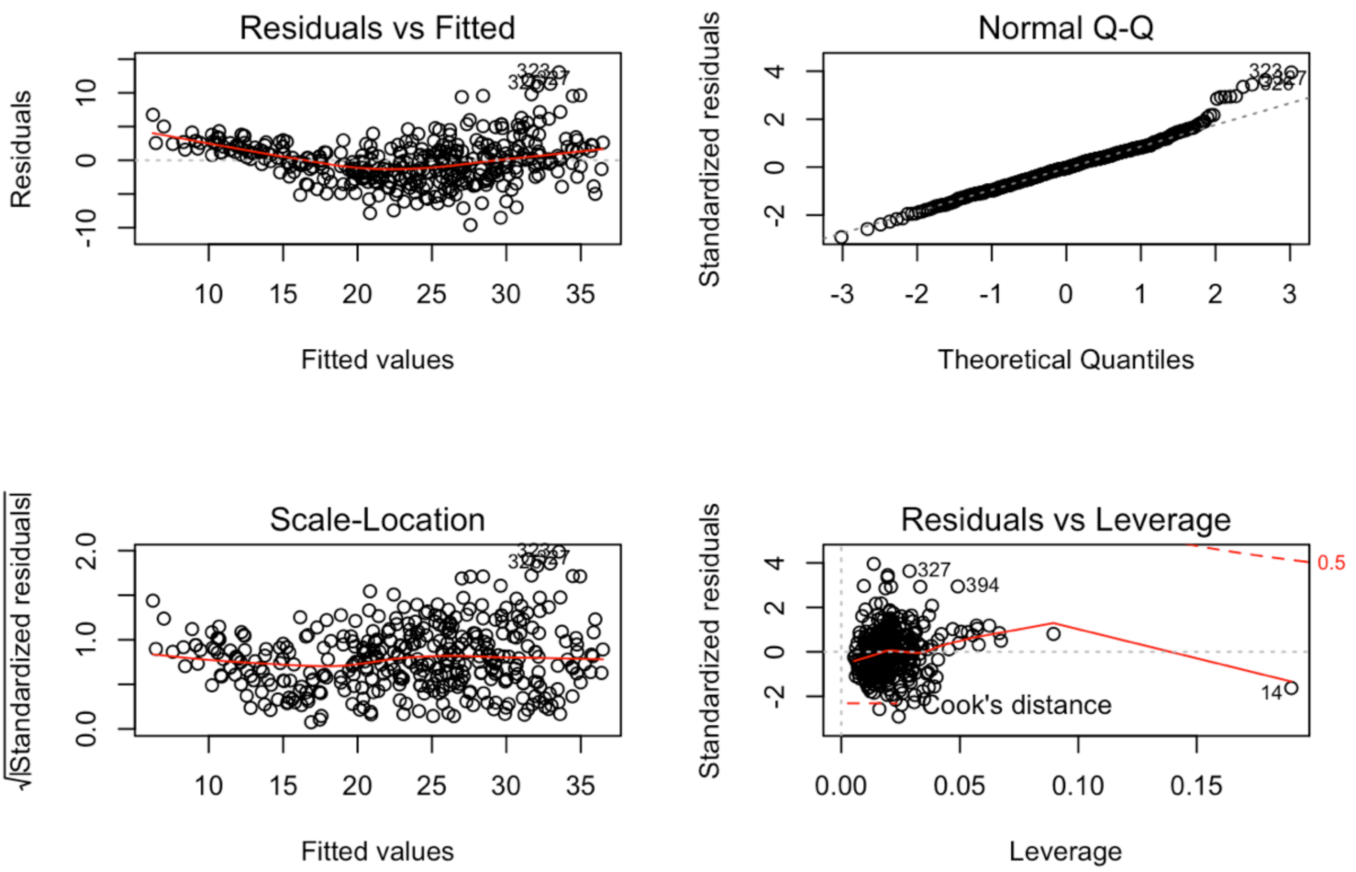
```
lmans = lm(mpg~.-name, data=Auto)
summary(lmans)
```

```
##
## 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. Yes. The F-statistic suggests that the null hypothesis is wrong.
2. A low p-valueindicates that the predictor is important. Important variables: displacement, weight, year, origin.
3. The estimated coefficient suggests `year` has a relatively strong postive effect on `mpg`.

d

```
par(mfrow=c(2,2))
plot(lmans)
```



The linear regression result is not good enough because the residual plots are distributed on a curve rather than randomly.

Observation 14 has an unsually high leverage.

e

Here are two examples of statistically significant interaction exxfects:

```
try1 = lm(mpg~displacement+weight+year*origin, data=Auto)
summary(try1)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + weight + year * origin, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7541 -1.8722 -0.0936  1.6900 12.4650
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.927e+00  8.873e+00   0.893 0.372229
## displacement  1.551e-03  4.859e-03   0.319 0.749735
## weight       -6.394e-03  5.526e-04 -11.571  < 2e-16 ***
## year          4.313e-01  1.130e-01   3.818 0.000157 ***
## origin       -1.449e+01  4.707e+00  -3.079 0.002225 **
## year:origin   2.023e-01  6.047e-02   3.345 0.000904 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.303 on 386 degrees of freedom
## Multiple R-squared:  0.8232, Adjusted R-squared:  0.8209
## F-statistic: 359.5 on 5 and 386 DF,  p-value: < 2.2e-16
```

```
try2 = lm(mpg~displacement*weight+year+origin, data=Auto)
summary(try2)
```

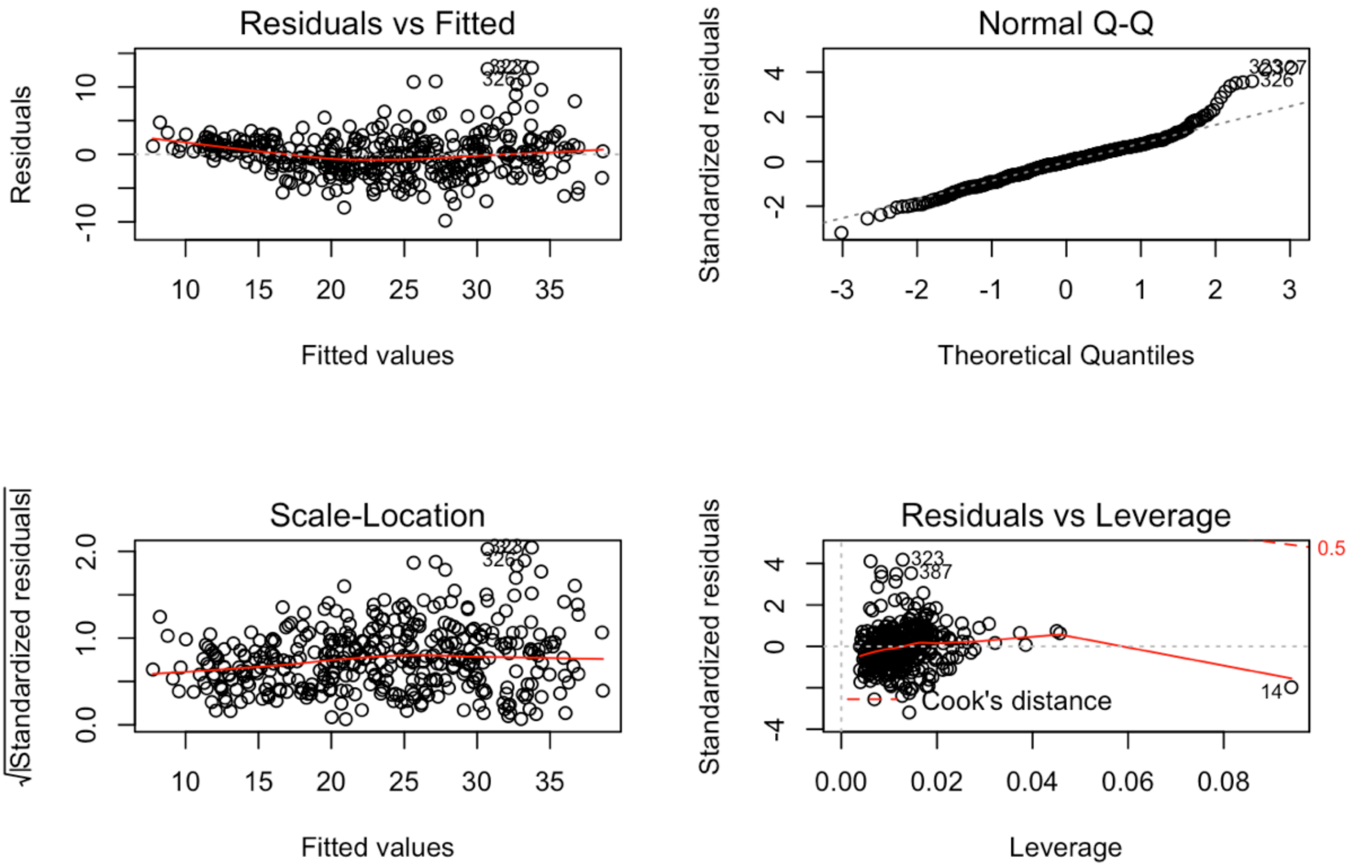
```
##
## Call:
## lm(formula = mpg ~ displacement * weight + year + origin, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6119  -1.7290  -0.0115   1.5609  12.5584
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -8.007e+00  3.798e+00  -2.108   0.0357 *
## displacement        -7.148e-02  9.176e-03  -7.790 6.27e-14 ***
## weight              -1.054e-02  6.530e-04 -16.146  < 2e-16 ***
## year                 8.194e-01  4.518e-02  18.136  < 2e-16 ***
## origin               3.567e-01  2.574e-01   1.386   0.1666
## displacement:weight  2.104e-05  2.214e-06   9.506  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.016 on 386 degrees of freedom
## Multiple R-squared:  0.8526, Adjusted R-squared:  0.8507
## F-statistic: 446.5 on 5 and 386 DF,  p-value: < 2.2e-16
```

f

```
try3 = lm(mpg~I(displacement^2)+I(log(weight))+sqrt(year)+origin, data=Auto)
summary(try3)
```

```
##
## Call:
## lm(formula = mpg ~ I(displacement^2) + I(log(weight)) + sqrt(year) +
##     origin, data = Auto)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.812 -1.834 -0.051  1.633 12.854
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.391e+01  1.113e+01   6.640 1.06e-10 ***
## I(displacement^2)  2.185e-05  6.647e-06   3.287  0.00111 **
## I(log(weight))    -2.231e+01  1.184e+00 -18.840  < 2e-16 ***
## sqrt(year)         1.432e+01  8.083e-01  17.716  < 2e-16 ***
## origin             7.790e-01  2.450e-01   3.180  0.00159 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.093 on 387 degrees of freedom
## Multiple R-squared:  0.8445, Adjusted R-squared:  0.8429
## F-statistic: 525.6 on 4 and 387 DF,  p-value: < 2.2e-16
```

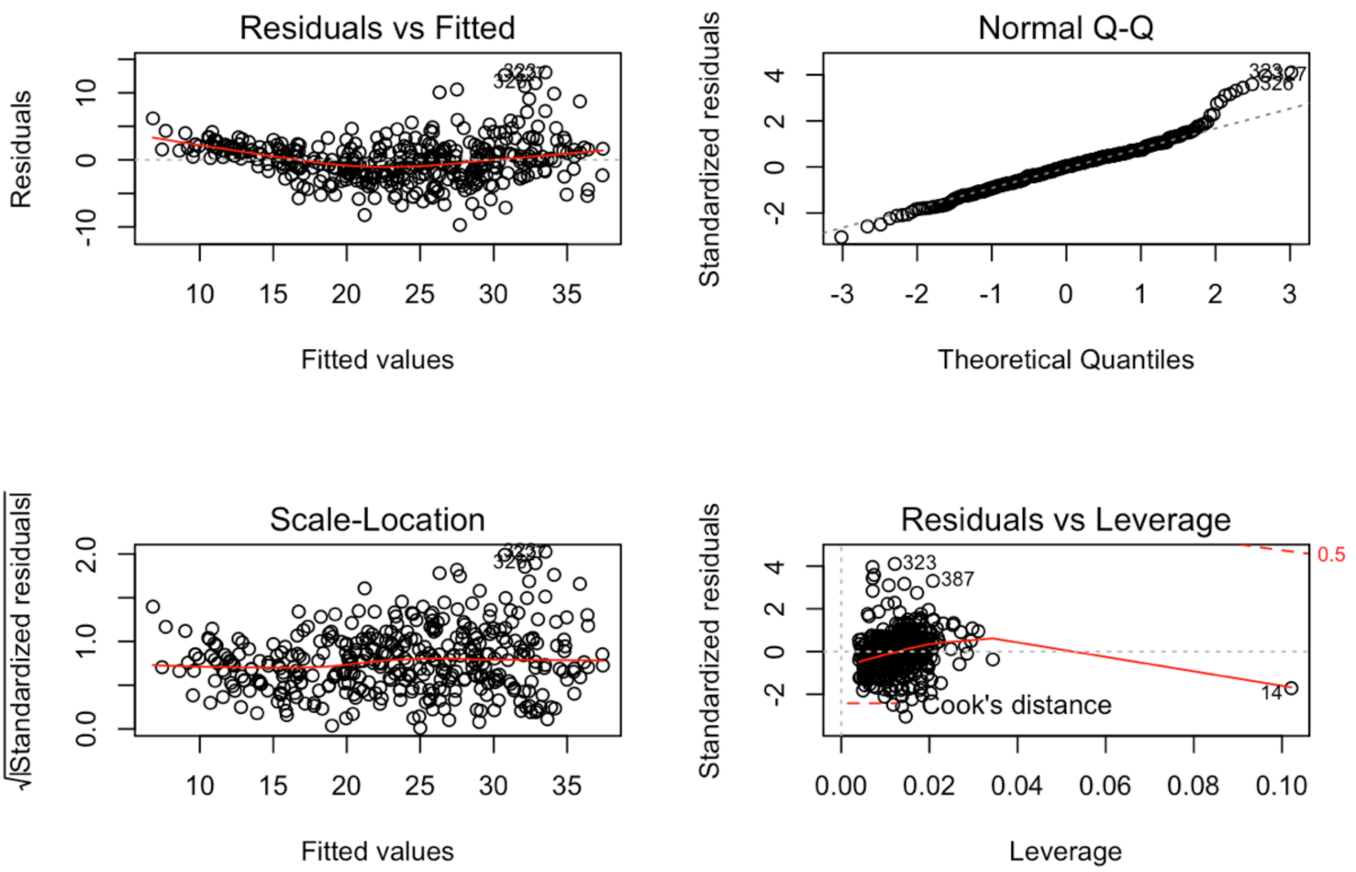
```
par(mfrow=c(2,2))
plot(try3)
```




```
try3 = lm(mpg~displacement+I(sqrt(weight))+year+sqrt(origin), data=Auto)
summary(try3)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + I(sqrt(weight)) + year + sqrt(origin),
##     data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7017 -2.0180  0.0714  1.6836 13.0757
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.308703   4.439889   0.070   0.9446
## displacement     0.009042   0.004436   2.038   0.0422 *
## I(sqrt(weight)) -0.786885   0.057595 -13.662  < 2e-16 ***
## year             0.794031   0.047867  16.588  < 2e-16 ***
## sqrt(origin)     2.921962   0.698142   4.185 3.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.205 on 387 degrees of freedom
## Multiple R-squared:  0.8331, Adjusted R-squared:  0.8314
## F-statistic: 482.9 on 4 and 387 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(try3)
```



By incresing the flexibility of the models properly, the performance generally improves.

10

P125 3.7.14

a

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

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

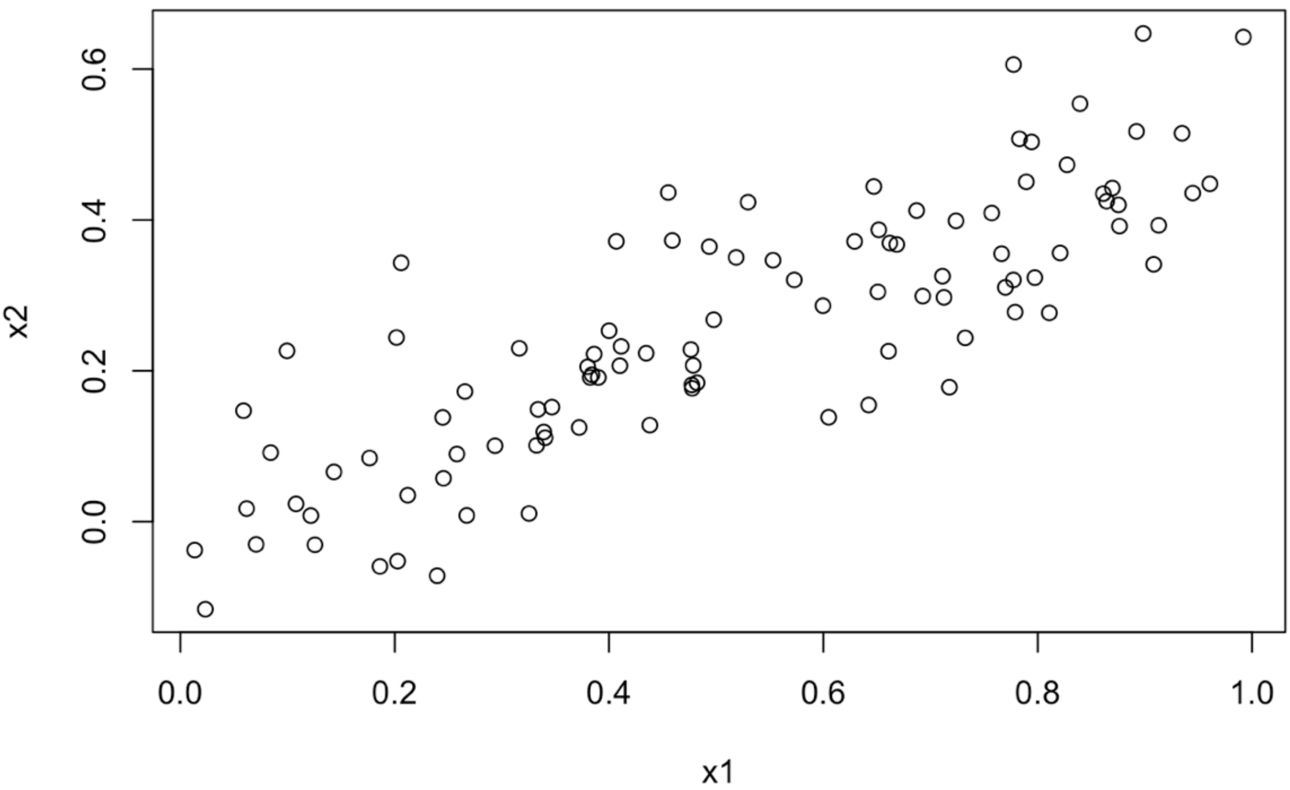
$\beta_0 = 2, \beta_1 = 2, \beta_2 = 0.3$

b

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

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

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



C

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

```
##
## 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
```

Estimated beta coefficients: $\hat{\beta_0} = 2.13, \hat{\beta_1} = 1.44, \hat{\beta_2} = 1.01$.

$\hat{\beta_0}$ is close to the true β_0, while $\hat{\beta_1}$, and $\hat{\beta_2}$ have high error.

Reject $H_0 : \beta_1 = 0$; Cannot reject $H_0 : \beta_2 = 0$.

d

```
fity1 <- lm(y~x1)
summary(fity1)
```

```
##
## 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
```

The null hypothesis can be rejected because the p-value for its t-statistic is small enough.

e

```
fity2 <- lm(y~x2)
summary(fity2)
```

```
##
## 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
```

The null hypothesis can be rejected because the p-value for its t-statistic is small enough.

f

No. The two input variables, x_1 and x_2 are related to one another, making it difficult to separate out the individual effects of two variables. This is called Collinearity.

g

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

```
##
## 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
```

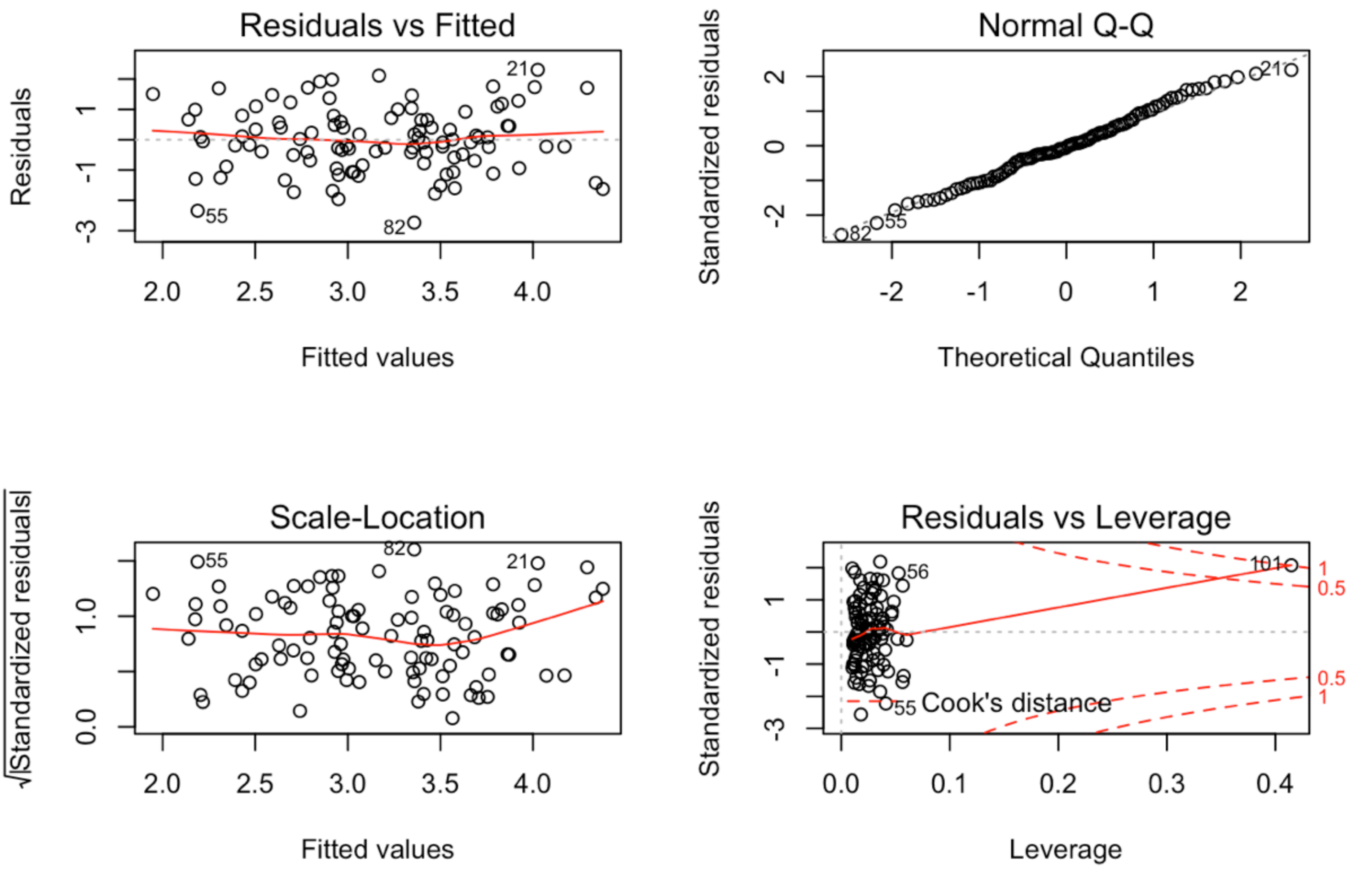
```
fity1 <- lm(y~x1)
summary(fity1)
```

```
##
## 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
```

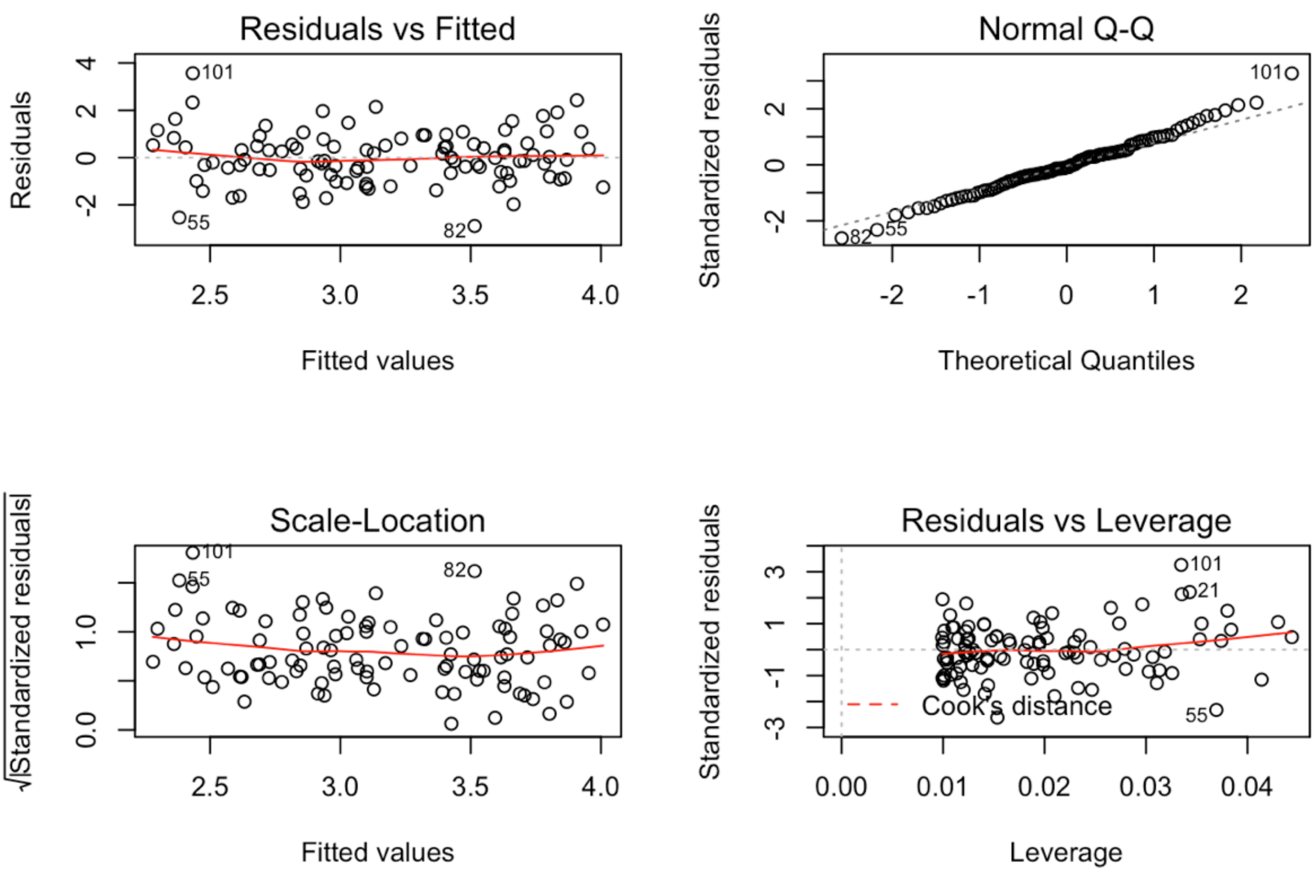
```
fity2 <- lm(y~x2)
summary(fity2)
```

```
##
## 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
```

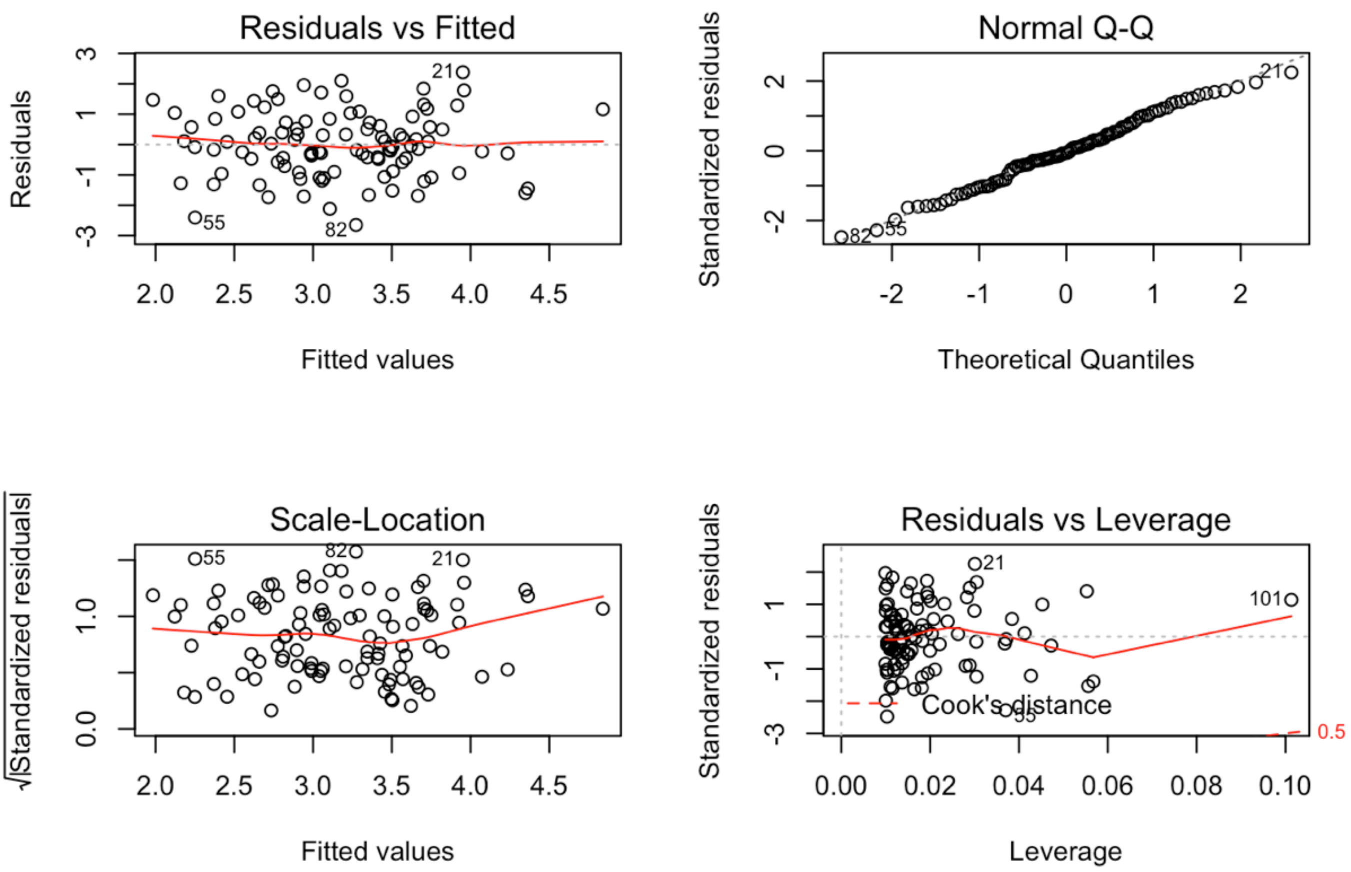
```
par(mfrow=c(2,2))
plot(fity)
```



```
par(mfrow=c(2,2))
plot(fity1)
```




```
par(mfrow=c(2,2))
plot(fity2)
```



- In the first model, x1 turns statistically insignificance and x2 turns statistiscal significance.
- The new observation has more effects in the first model.
- The new observation is an outlier in the first and the third model.
- The new observation is a high-leverage in the first and the third model.