

Homework 2

Yuan-An Liu A20375099

Q8.

(a)

i. Yes, p-value of the F-statistic is close to 0.

ii. strong relationship. The mean of mpg is 23.45. Because of the RSE is 4.906, it shows the percentage error of close to 20%. So A huge percentage of the variance in mpg is explained by horsepower.

iii. The relationship between mpg and horsepower is “Negative”

```
summary(Auto)
```

```
##           mpg           cylinders      displacement      horsepower
##  Min.       : 9.00    Min.       :3.000    Min.       : 68.0    Min.       : 46.0
##  1st Qu.:17.00    1st Qu.:4.000    1st Qu.:105.0    1st Qu.: 75.0
##  Median :22.75    Median :4.000    Median :151.0    Median : 93.5
##  Mean   :23.45    Mean   :5.472    Mean   :194.4    Mean   :104.5
##  3rd Qu.:29.00    3rd Qu.:8.000    3rd Qu.:275.8    3rd Qu.:126.0
##  Max.   :46.60    Max.   :8.000    Max.   :455.0    Max.   :230.0
##
##           weight      acceleration           year           origin
##  Min.       :1613    Min.       : 8.00    Min.       :70.00    Min.       :1.000
##  1st Qu.:2225    1st Qu.:13.78    1st Qu.:73.00    1st Qu.:1.000
##  Median :2804    Median :15.50    Median :76.00    Median :1.000
##  Mean   :2978    Mean   :15.54    Mean   :75.98    Mean   :1.577
##  3rd Qu.:3615    3rd Qu.:17.02    3rd Qu.:79.00    3rd Qu.:2.000
##  Max.   :5140    Max.   :24.80    Max.   :82.00    Max.   :3.000
##
##           name
##  amc matador      : 5
##  ford pinto       : 5
##  toyota corolla   : 5
##  amc gremlin      : 4
##  amc hornet       : 4
##  chevrolet chevette: 4
##  (Other)          :365
```

```
lm.fit =lm(mpg~horsepower ,data=Auto)
```

```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66  <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

iv.

```
# the predicted mpg associated with a horsepower of 98
```

```
predict(lm.fit, data.frame(horsepower=c(98)))
```

```
##           1
```

```
## 24.46708
```

```
# the associated 95% confidence interval
```

```
predict(lm.fit, data.frame(horsepower=c(98)), interval = "confidence")
```

```
##           fit          lwr          upr
```

```
## 1 24.46708 23.97308 24.96108
```

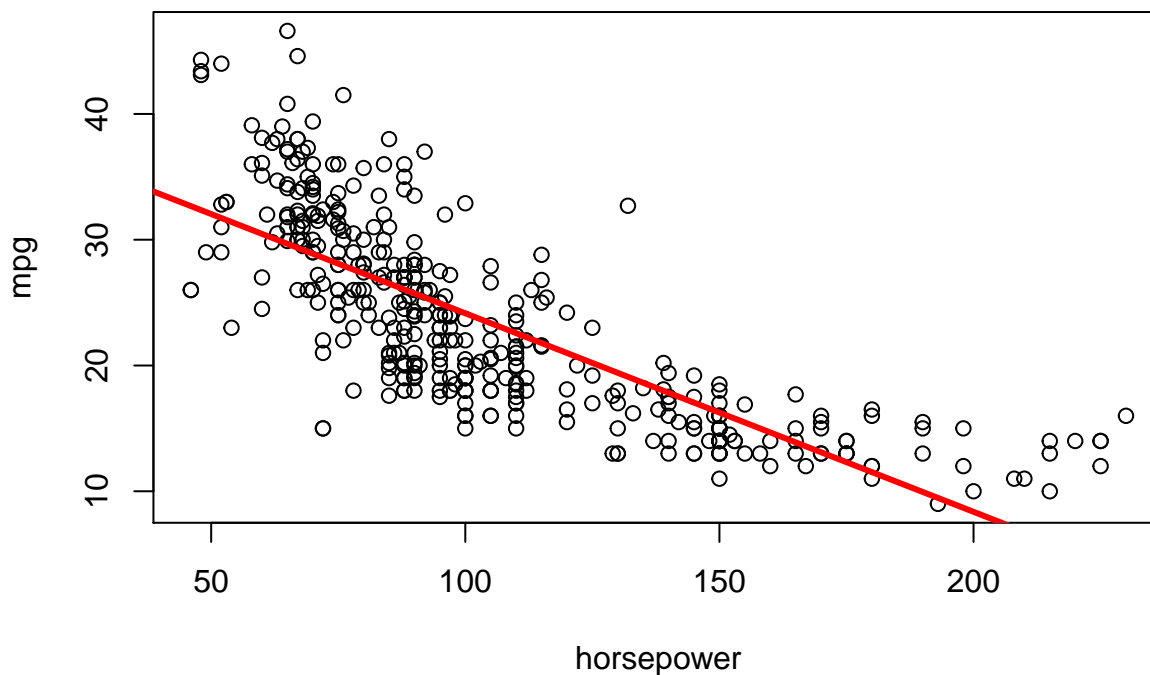
```
# the associated 95% prediction interval
```

```
predict(lm.fit, data.frame(horsepower=c(98)), interval = "prediction")
```

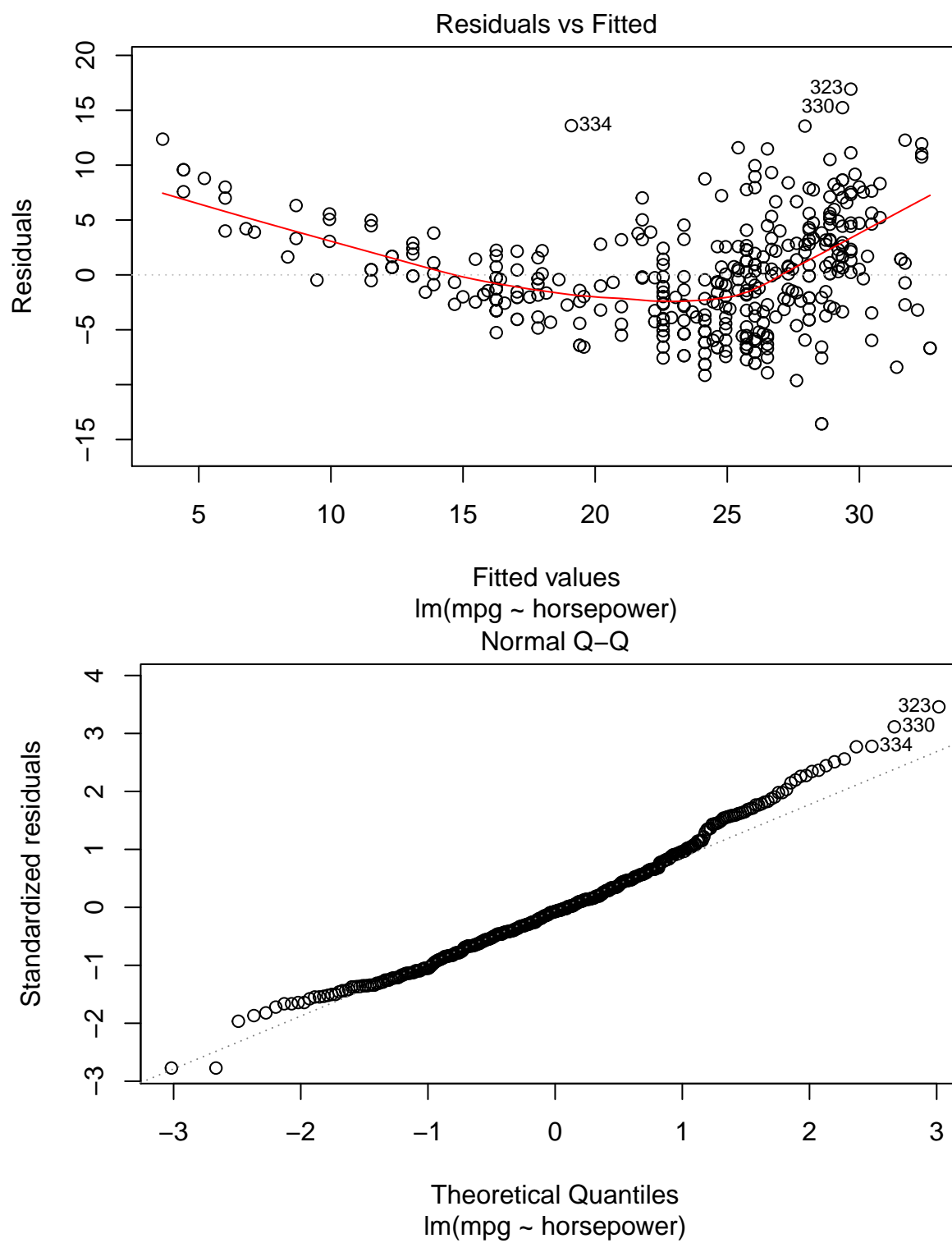
```
##           fit          lwr          upr
```

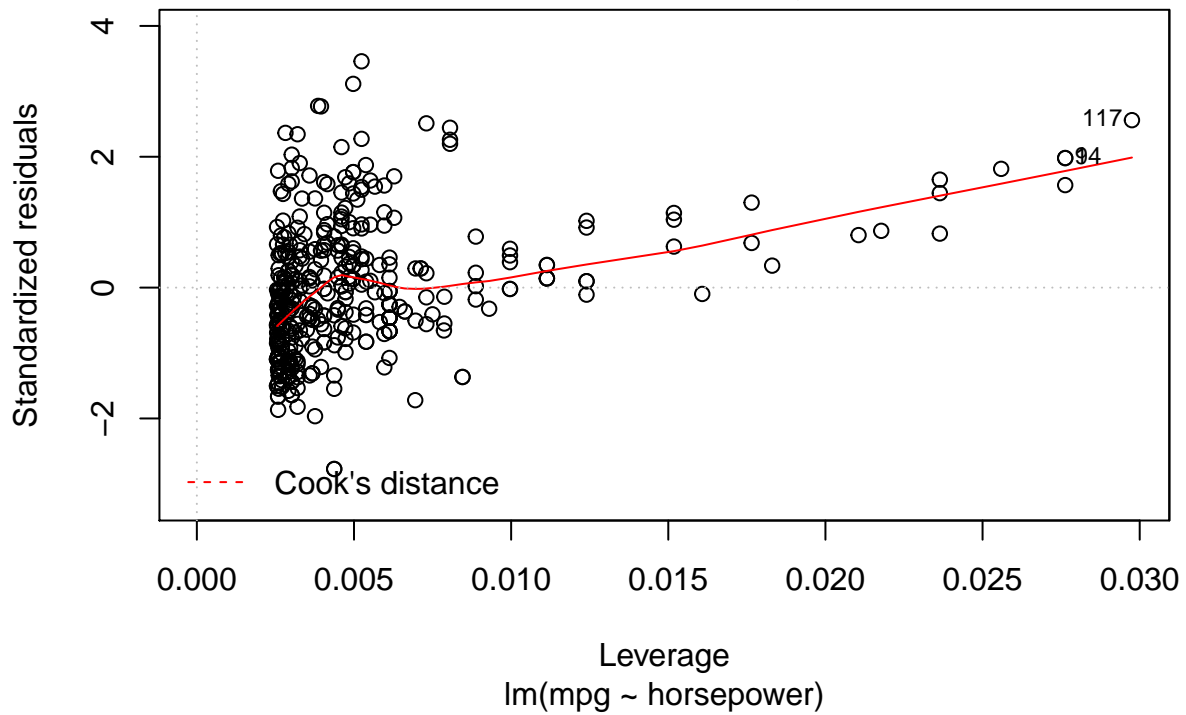
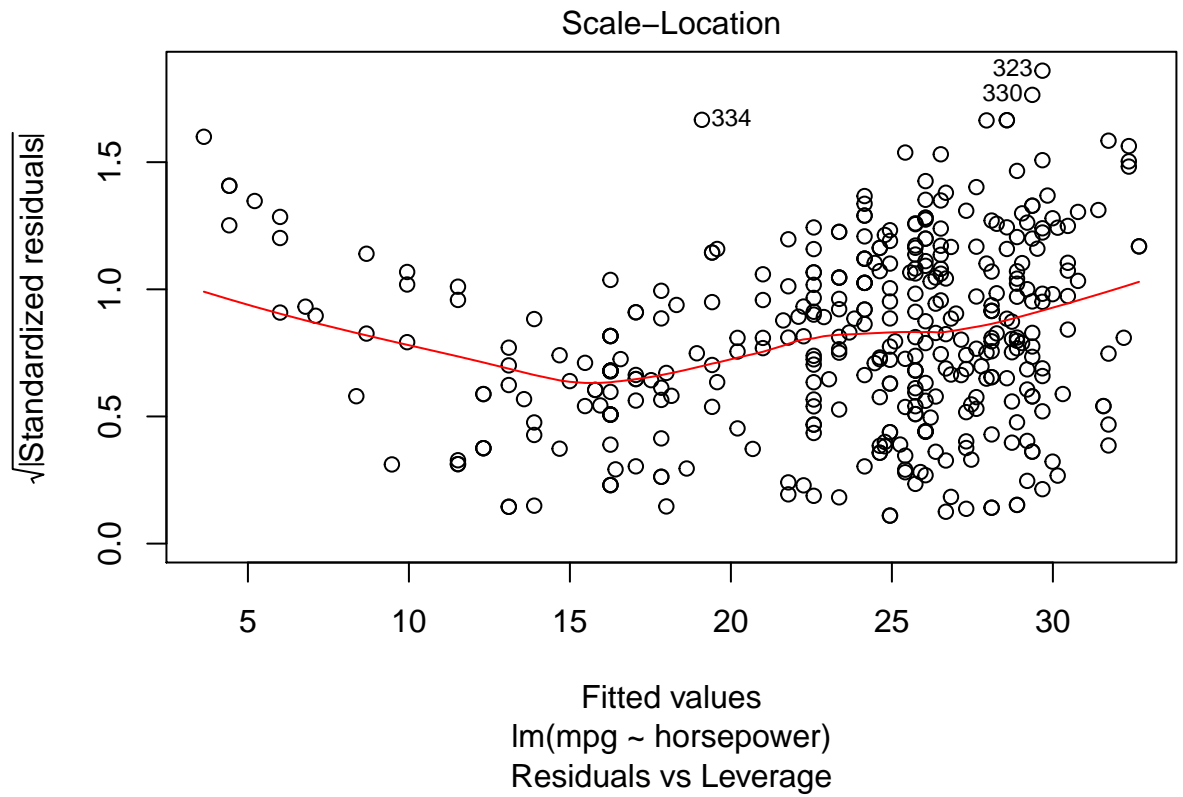
```
## 1 24.46708 14.8094 34.12476
```

(b)



(c)

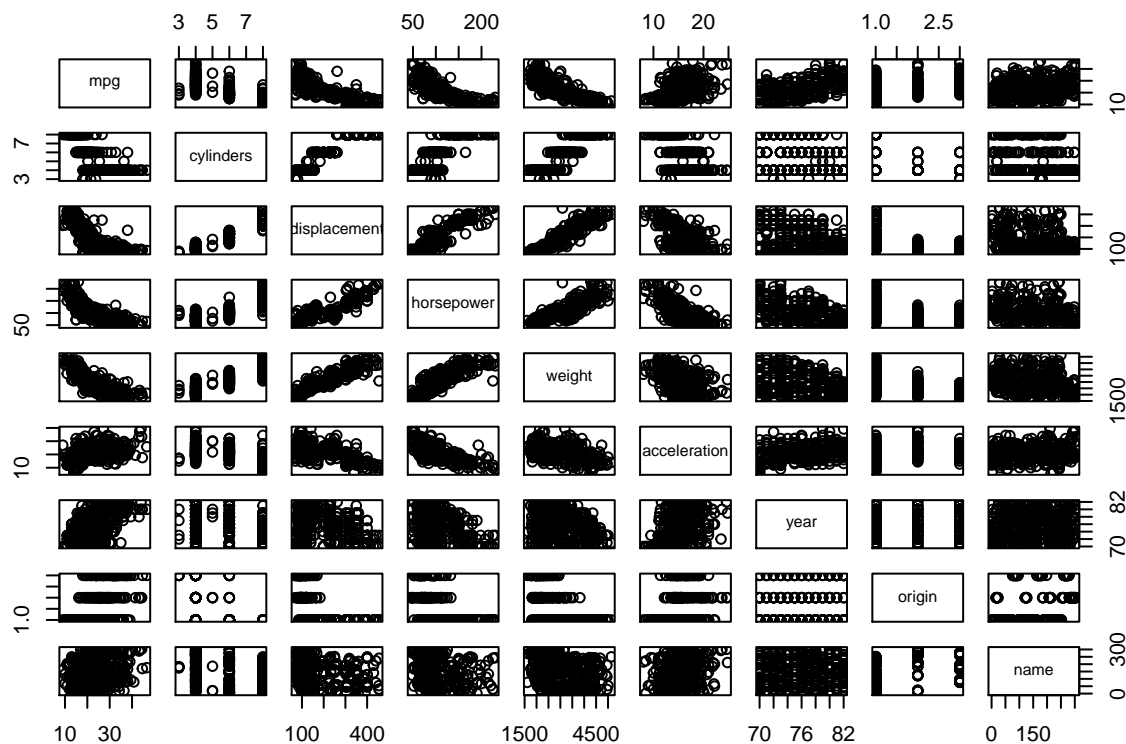




Q9.

(a)

```
pairs(Auto) # scatterplot matrix
```



(b)

```
cor(Auto[,1:8]) # correlations between the variables without names
```

```
##           mpg  cylinders displacement horsepower  weight
## mpg      1.000000 -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.fit =lm(mpg~.-name,data=Auto)
summary(lm.fit)
```

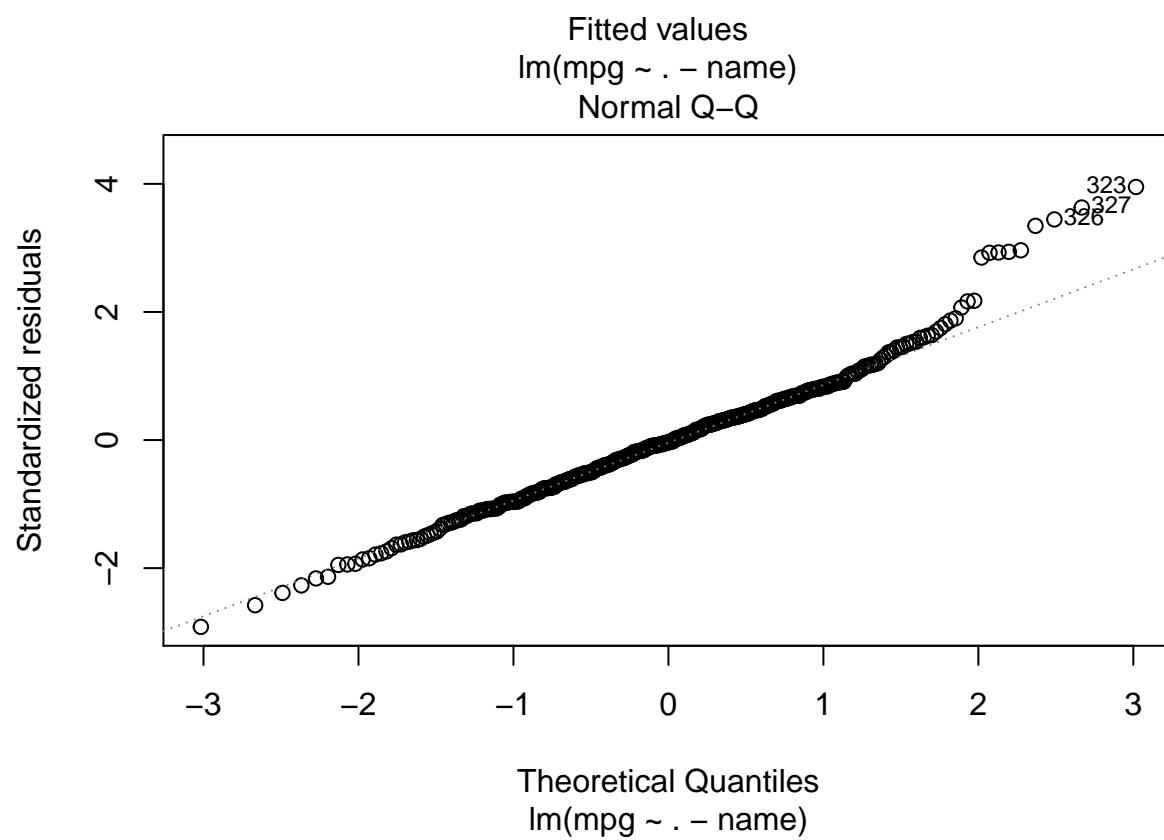
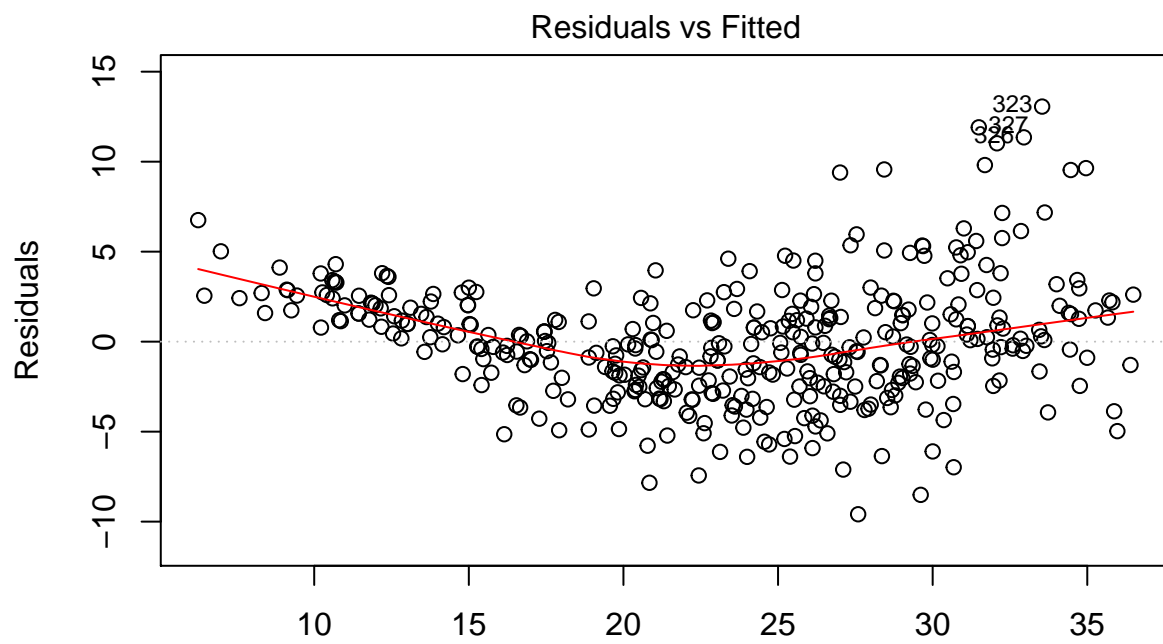
```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.218435   4.644294  -3.707  0.00024 ***
## 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
```

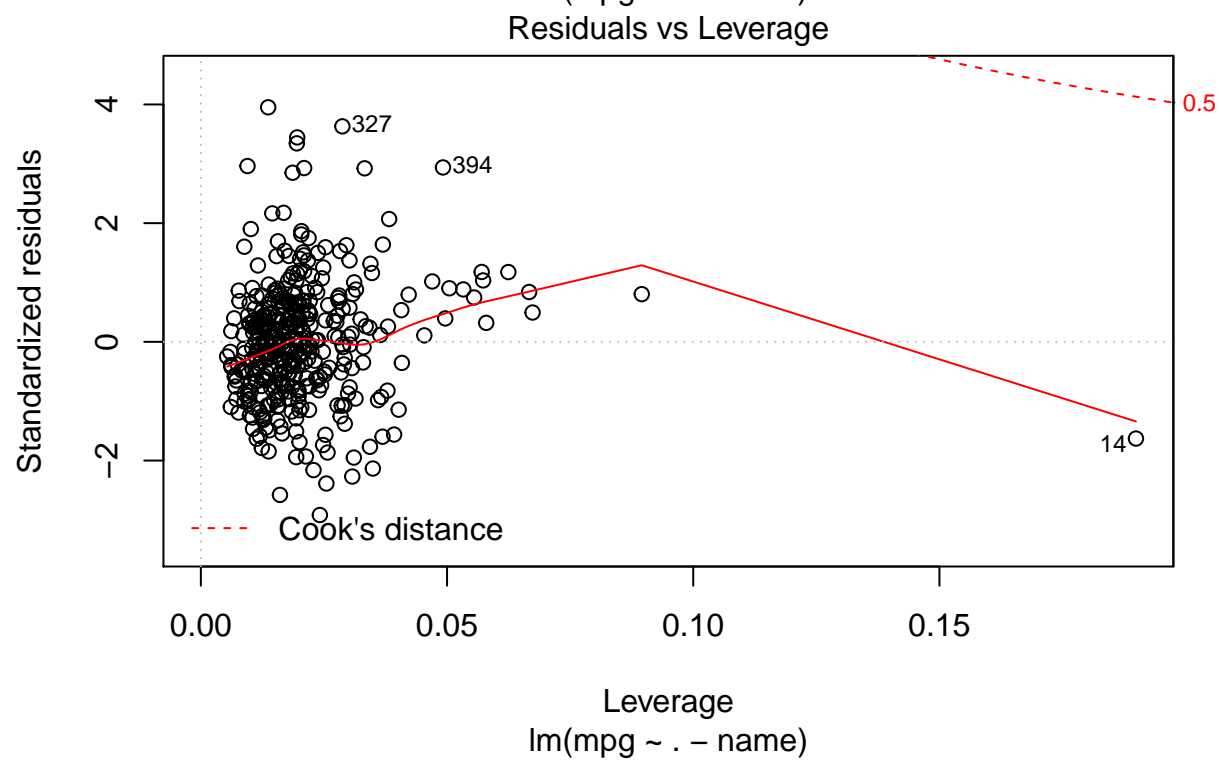
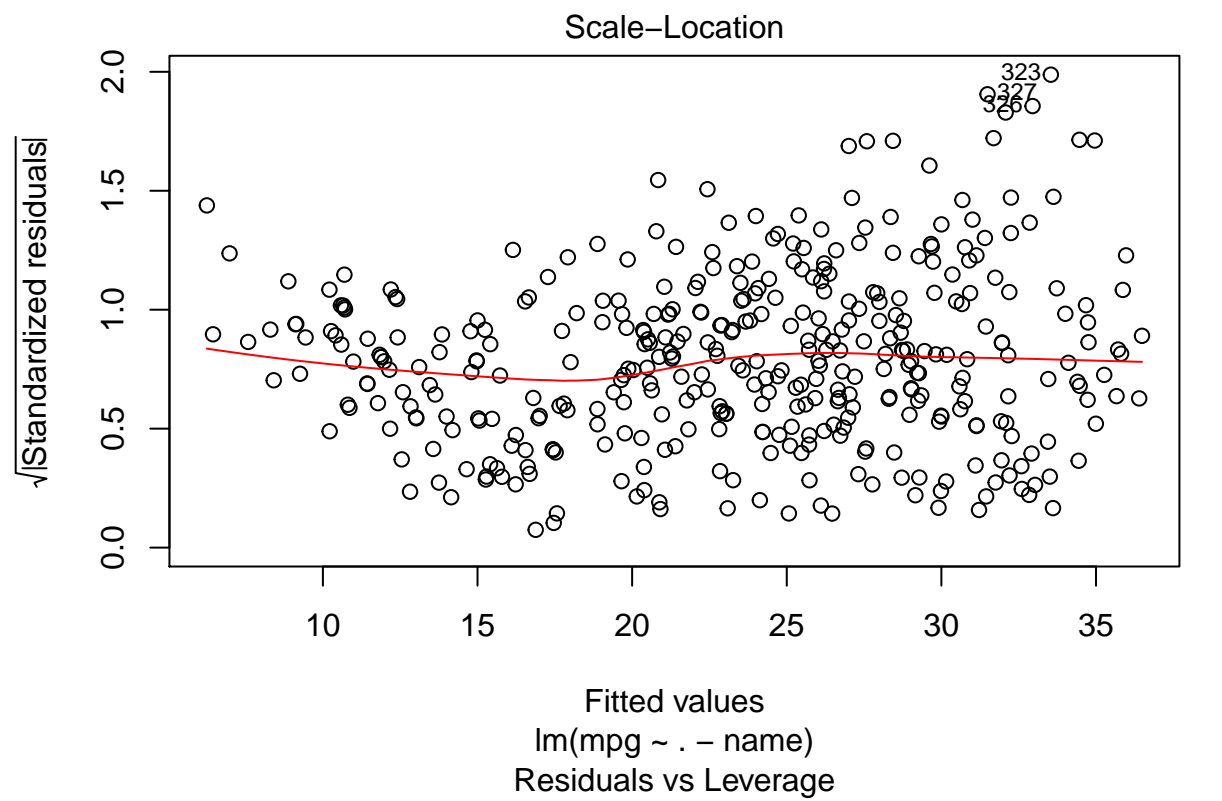
i. Yes

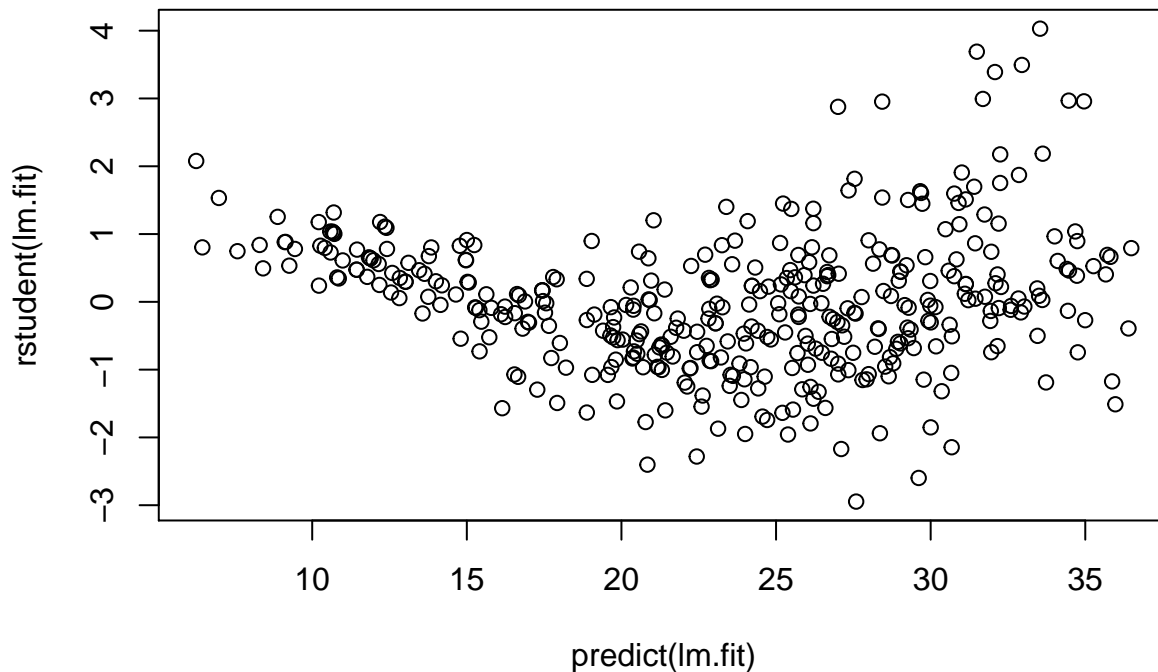
ii. displacement, weight, year and origin

iii. Regression coefficient of year: 0.75 The lower the “year”, the higher the “mpg”

(d)







- The fit seems not linear. There is a curve.
- There are some values in “rstudent” > 3

(e)

```
Auto2 = Auto[,1:8]
lm2.fit = lm(mpg~*., data = Auto2)
summary(lm2.fit)
```

```
##
## Call:
## lm(formula = mpg ~ . * ., data = Auto2)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-7.6303	-1.4481	0.0596	1.2739	11.1386

```
##
## Coefficients:
```

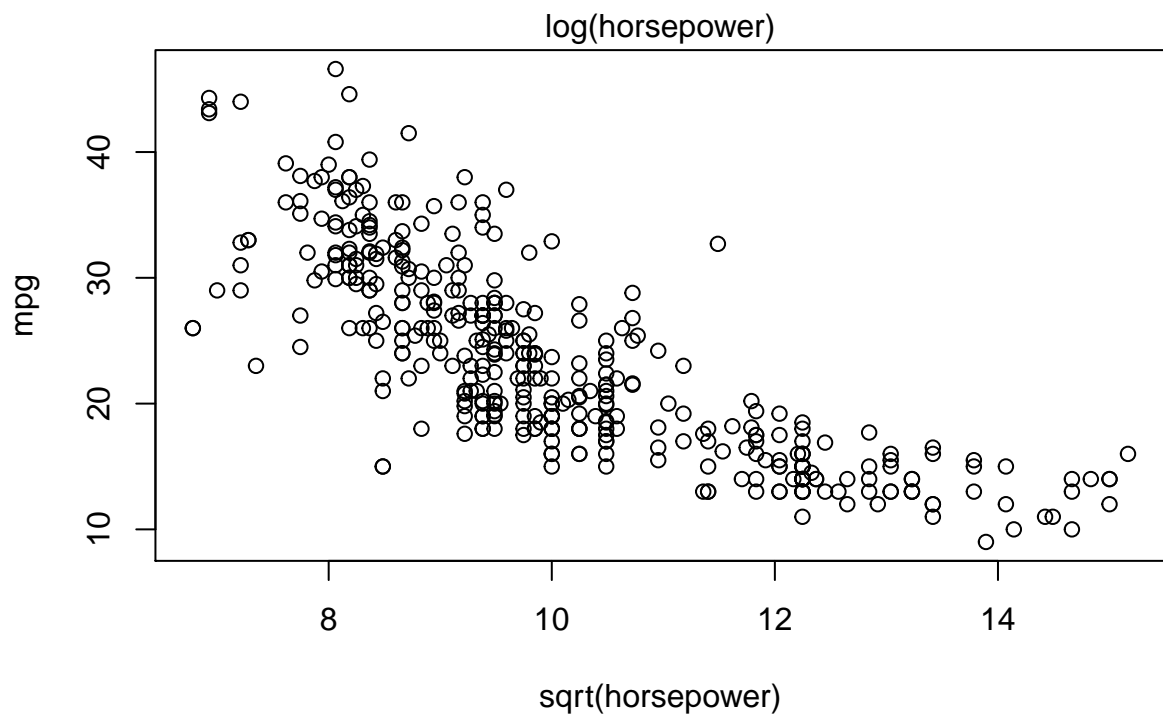
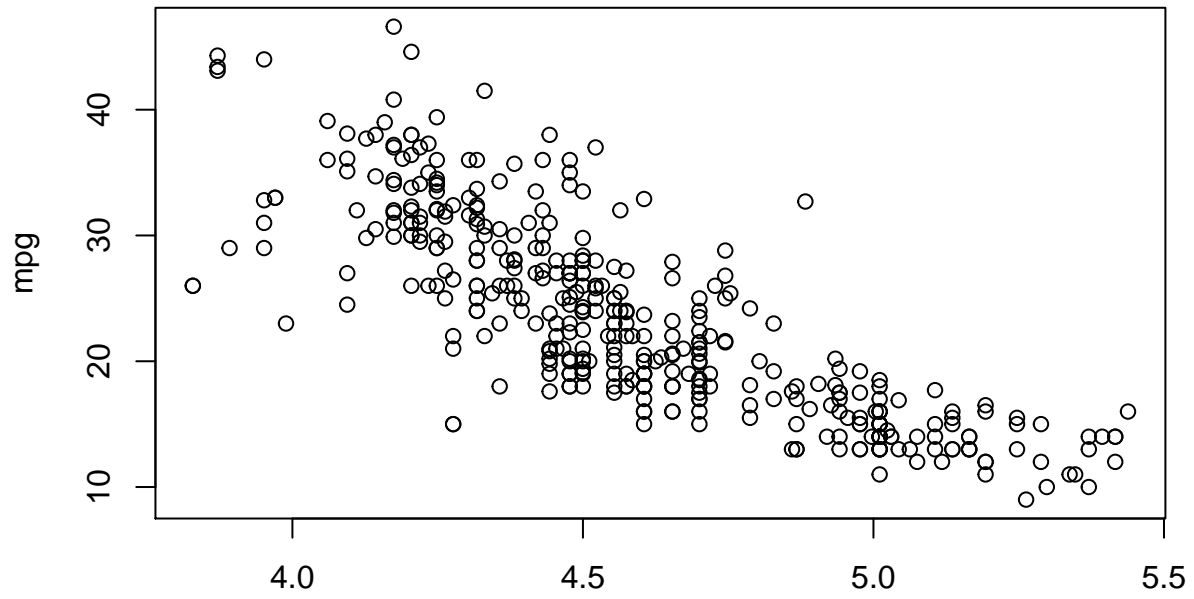
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.548e+01	5.314e+01	0.668	0.50475
cylinders	6.989e+00	8.248e+00	0.847	0.39738
displacement	-4.785e-01	1.894e-01	-2.527	0.01192 *
horsepower	5.034e-01	3.470e-01	1.451	0.14769
weight	4.133e-03	1.759e-02	0.235	0.81442
acceleration	-5.859e+00	2.174e+00	-2.696	0.00735 **
year	6.974e-01	6.097e-01	1.144	0.25340
origin	-2.090e+01	7.097e+00	-2.944	0.00345 **
cylinders:displacement	-3.383e-03	6.455e-03	-0.524	0.60051
cylinders:horsepower	1.161e-02	2.420e-02	0.480	0.63157
cylinders:weight	3.575e-04	8.955e-04	0.399	0.69000

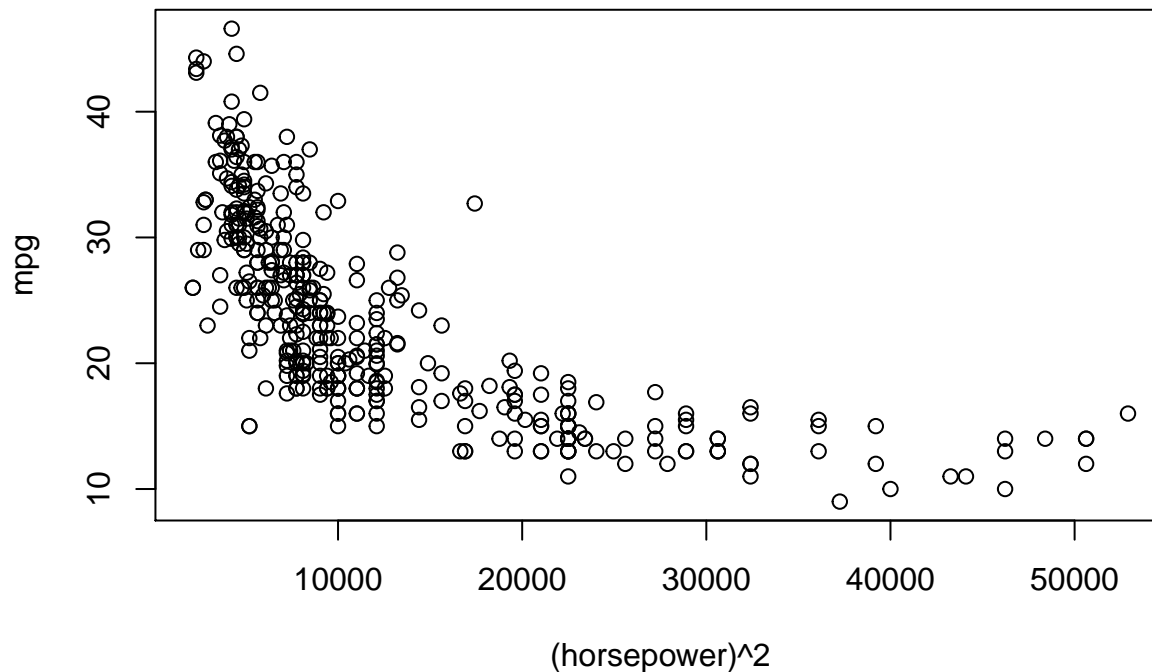
```
## cylinders:acceleration      2.779e-01  1.664e-01   1.670  0.09584 .
## cylinders:year             -1.741e-01  9.714e-02  -1.793  0.07389 .
## cylinders:origin           4.022e-01  4.926e-01   0.816  0.41482
## displacement:horsepower    -8.491e-05  2.885e-04  -0.294  0.76867
## displacement:weight        2.472e-05  1.470e-05   1.682  0.09342 .
## displacement:acceleration  -3.479e-03  3.342e-03  -1.041  0.29853
## displacement:year          5.934e-03  2.391e-03   2.482  0.01352 *
## displacement:origin        2.398e-02  1.947e-02   1.232  0.21875
## horsepower:weight          -1.968e-05  2.924e-05  -0.673  0.50124
## horsepower:acceleration    -7.213e-03  3.719e-03  -1.939  0.05325 .
## horsepower:year            -5.838e-03  3.938e-03  -1.482  0.13916
## horsepower:origin          2.233e-03  2.930e-02   0.076  0.93931
## weight:acceleration         2.346e-04  2.289e-04   1.025  0.30596
## weight:year                 -2.245e-04  2.127e-04  -1.056  0.29182
## weight:origin               -5.789e-04  1.591e-03  -0.364  0.71623
## acceleration:year           5.562e-02  2.558e-02   2.174  0.03033 *
## acceleration:origin         4.583e-01  1.567e-01   2.926  0.00365 **
## year:origin                 1.393e-01  7.399e-02   1.882  0.06062 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.695 on 363 degrees of freedom
## Multiple R-squared:  0.8893, Adjusted R-squared:  0.8808
## F-statistic: 104.2 on 28 and 363 DF,  p-value: < 2.2e-16
```

From the p-values, e.g. acceleration:origin is statistically significant

(f)

```
## The following objects are masked from Auto (pos = 3):
##
##      acceleration, cylinders, displacement, horsepower, mpg, name,
##      origin, weight, year
```





It is a better fit than the original one.

Q10.

(a)

```
lm.fit = lm(Sales ~ Price + Urban + US, data = Carseats)
summary(lm.fit)

##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.043469   0.651012  20.036 < 2e-16 ***
## Price        -0.054459   0.005242 -10.389 < 2e-16 ***
## UrbanYes     -0.021916   0.271650  -0.081  0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF, p-value: < 2.2e-16
```

(b)

Price

urbanYes

USYes

(c)

$\text{Sales} = 13.04 + -0.05 \text{ Price} + -0.02 \text{ UrbanYes} + 1.20 \text{ USYes}$

(d)

Price and USYes

(e)

```
#uses the predictors for which there is evidence of association with the outcome
lm2.fit = lm(Sales~Price + US, data = Carseats)
summary(lm2.fit)
```

```
##
## Call:
## lm(formula = Sales ~ Price + US, data = Carseats)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.03079    0.63098   20.652 < 2e-16 ***
## Price        -0.05448    0.00523  -10.416 < 2e-16 ***
## USYes         1.19964    0.25846   4.641 4.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

(f)

```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036 < 2e-16 ***
## Price       -0.054459   0.005242 -10.389 < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081  0.936
## USYes       1.200573    0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

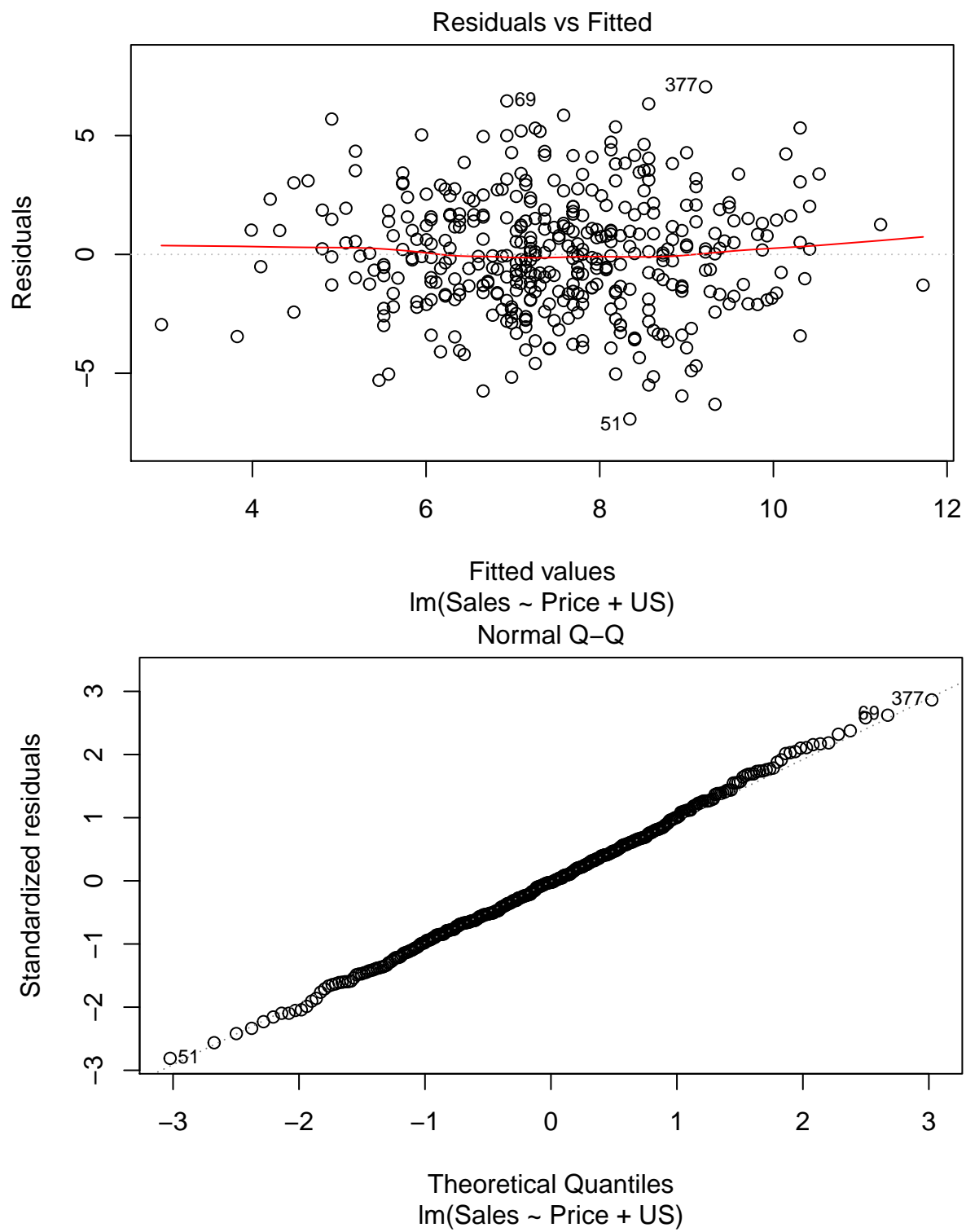
Comparing to lm2.fit, the two models are similarly fit.

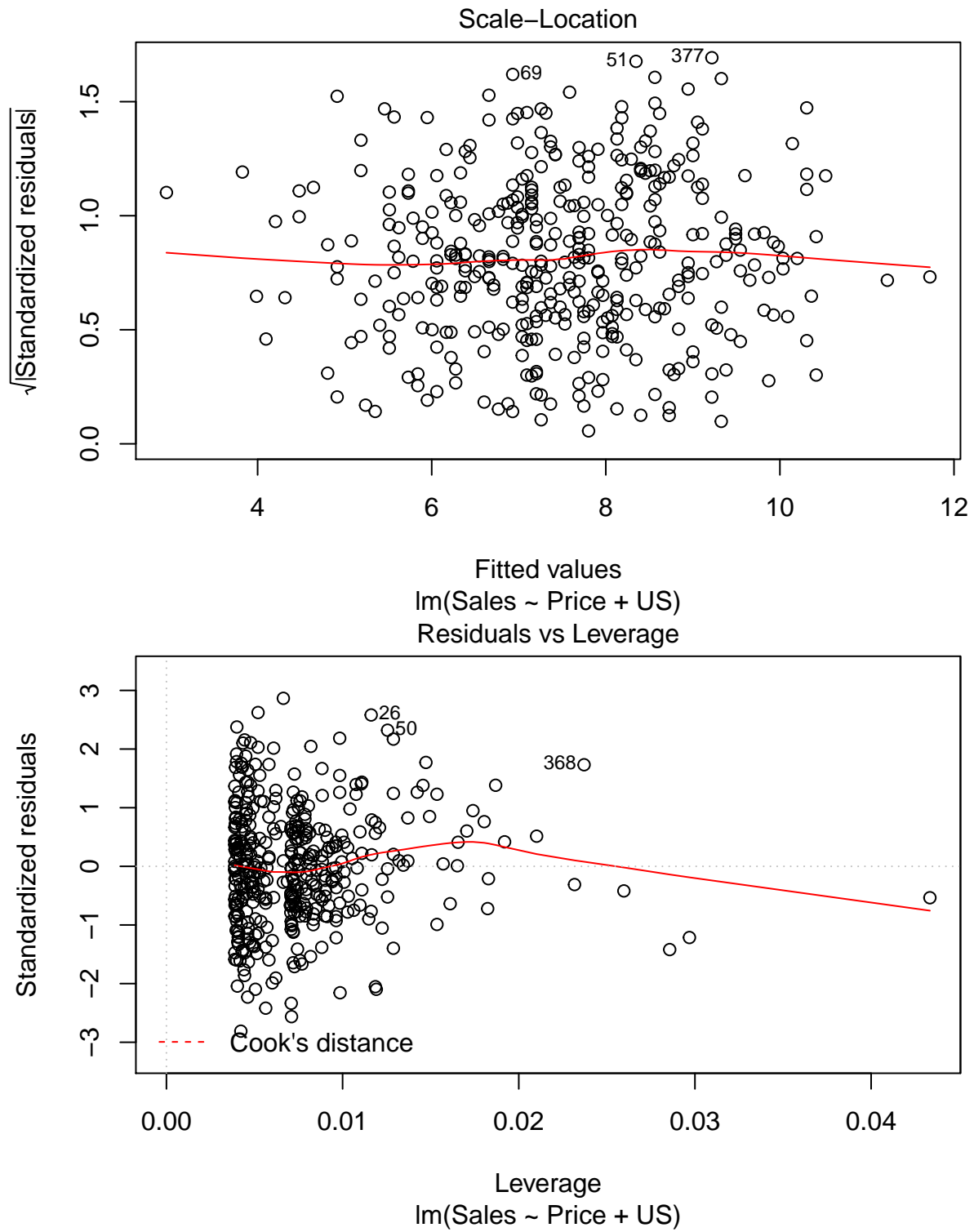
(g)

```
# Confidence Intervals:
confint(lm2.fit)
```

```
##              2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes       0.69151957  1.70776632
```

(h)





Q13.

```
set.seed(1)
```


set.seed(1) prior to starting part (a): according to the question

(a)

```
x = rnorm(100, 0, 1)
```

(b)

```
eps = rnorm(100, 0, 0.25)
```

(c)

```
y = -1 + 0.5*x + eps  
length(y)
```

```
## [1] 100
```

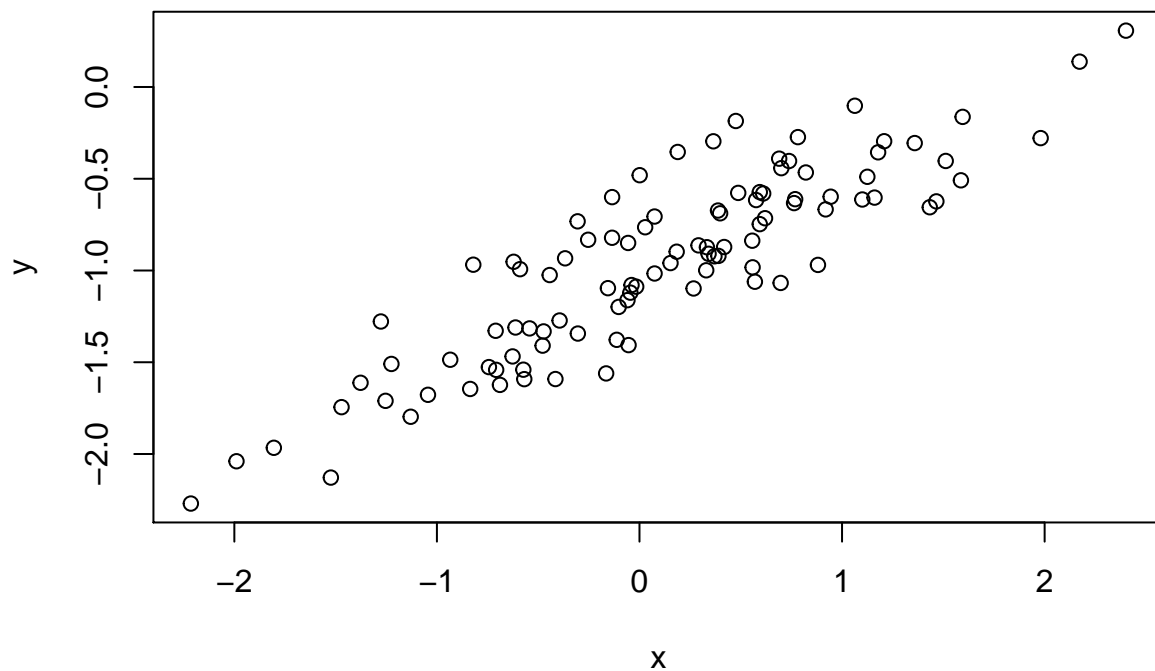
```
summary(y)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
## -2.2700 -1.3294 -0.9215 -0.9550 -0.6021  0.3071
```

length of vector y = 100 ; beta0 = -1 ; beta1 = 0.5

(d)

```
## Warning in abline(lm.fit): only using the first two of 4 regression  
## coefficients
```



Linear relationship

(e)

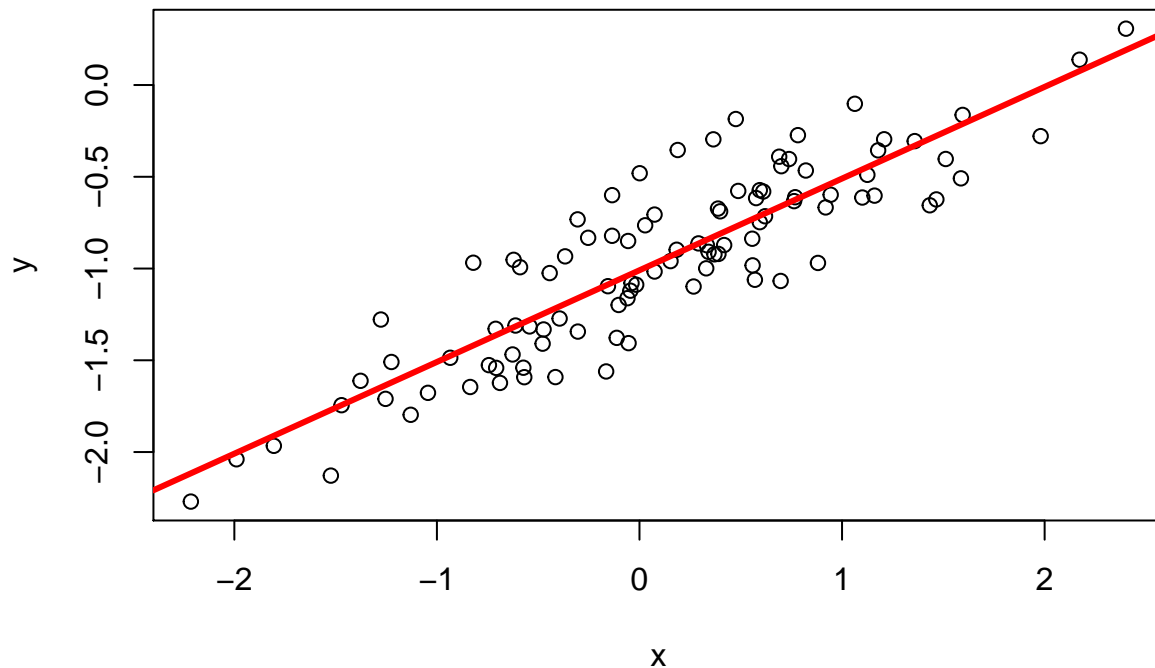
```
lm.fit = lm(y~x)
summary(lm.fit)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46921 -0.15344 -0.03487  0.13485  0.58654
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00942    0.02425  -41.63  <2e-16 ***
## x            0.49973    0.02693   18.56  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2407 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

beta0 and 1 are similar to the original values

(f)

```
plot(x,y)
abline(lm.fit)
abline (lm.fit ,lwd =3, col ="red")
```



(g)

```
lm2.fit = lm(y~poly(x, 2))
summary(lm2.fit)
```

```
##
## Call:
## lm(formula = y ~ poly(x, 2))
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-0.4913	-0.1563	-0.0322	0.1451	0.5675

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.95501	0.02395	-39.874	<2e-16 ***
poly(x, 2)1	4.46612	0.23951	18.647	<2e-16 ***
poly(x, 2)2	-0.33602	0.23951	-1.403	0.164

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2395 on 97 degrees of freedom
## Multiple R-squared:  0.7828, Adjusted R-squared:  0.7784
## F-statistic: 174.8 on 2 and 97 DF,  p-value: < 2.2e-16
```

Regression coefficient of the model is insignificant