154HW3

Jiyoon Clover Jeong

9/27/2017

1.

Since residual is $e_i = y_i - \hat{y_i}$, sum of residual can be written as

$$\sum_{i=1}^{n} y_i - \hat{y_i}$$

we can prove $\sum_{i=1}^{n} e_i = 0$ by proving

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \hat{y_i}$$

Since the normal equation is $X^T X\hat{\beta} = X^T Y$ and $\hat{Y} = X\hat{\beta}$,

$$X^T\hat{Y} = X^T Y$$

Since design matrix X is

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ . & \\ . & \\ 1 & x_n \end{bmatrix}$$

The first element of $X^T\hat{Y} = \begin{bmatrix} 1 & 1 & ... & 1 \end{bmatrix} \hat{Y}$ is $\sum_{i=1}^{n} \hat{y_i}$

The first element of $X^T Y = \begin{bmatrix} 1 & 1 & ... & 1 \end{bmatrix} Y$ is $\sum_{i=1}^{n} y_i$

Therefore, $\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \hat{y_i}$

and the given statement is proved.

```
# check by R
a <- lm(mpg ~ disp, data = mtcars)
#a$residuals
#residuals(a)
sum(residuals(a))
```

```
## [1] -3.608225e-15
```

2.

(a)

Since $X^T X$ is symmetric matrix, it is

$$\begin{bmatrix} 30 & 0 & 0 \\ 0 & 10 & 7 \\ 0 & 7 & 15 \end{bmatrix}$$

n is equal to 30

(b)

cor(X, Z) = $\frac{Cov(X,Z)}{SD(X)SD(Y)}$ =

$$\frac{\sum (x_i \ - \ \bar{x})(z_i \ - \ \bar{z})}{\sqrt{\sum (x_i \ - \ \bar{x})^2 \sum (z_i \ - \ \bar{z})^2}}$$

.

and

$$X^T X = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 10 & 7 \\ 0 & 7 & 15 \end{bmatrix} = \begin{bmatrix} ... & 1 & ... \\ ... & x & ... \\ ... & z & ... \end{bmatrix} \begin{bmatrix} . & . & . \\ . & . & . \\ 1 & x & z \\ . & . & . \\ . & . & . \end{bmatrix} = \begin{bmatrix} \sum 1 & \sum x_i & \sum z_i \\ \sum x_i & \sum x_i^2 & \sum x_i z_i \\ \sum z_i & \sum x_i z_i & \sum z_i^2 \end{bmatrix}$$

Therefore, divide sum of x and z by n=30 leads

$$\bar{x} \ = \ \frac{1}{30} \sum x_i \ = \ 0 \bar{z} \ = \ \frac{1}{30} \sum z_i \ = \ 0$$

.

So, cor(X, Z) =

$$\frac{\sum (x_i \ - \ \bar{x})(z_i \ - \ \bar{z})}{\sqrt{\sum (x_i \ - \ \bar{x})^2 \sum (z_i \ - \ \bar{z})^2}} = \frac{\sum x_i z_i}{\sqrt{\sum x_i^2 \sum z_i^2}} = \frac{7}{\sqrt{10}\sqrt{15}}$$

.

(c)

In ols equation, $\bar{y} = \bar{\hat{y}}$ so

$$\bar{\hat{y}} = \bar{y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x} + \hat{\beta}_2 \bar{z}$$

so $\bar{y} = -2 + \bar{x} + 2\bar{z}$. And $\bar{x} = 0$ and $\bar{z} = 0$ are from $X^T X$.

Therefore, $\bar{y}$ = -2.

(d)

$$R^2 = \frac{SS_{reg}}{SS_{total}} = \frac{SS_{reg}}{SS_{reg} + RSS}$$

RSS = 12

$$SS_{reg} = \sum (\hat{y_i} \ - \ \bar{y})^2 = \sum (-2 \ + \ x_i \ + \ 2z_i \ + \ 2)^2 = \sum (x_i \ + \ 2z_i)^2$$

$= \sum x_i^2 + 4 \sum x_i z_i + 4 \sum z_i^2 = 10 + 28 + 60 = 98$. So, $R^2 = \frac{SS_{reg}}{SS_{total}} = \frac{98}{12+98} = \frac{98}{110} = \frac{49}{55}$.

Thus, $R^2 = \frac{49}{55}$.

3.

(a)

```
set.seed(1)
x <- rnorm(100, 0, 1)
```

(b)

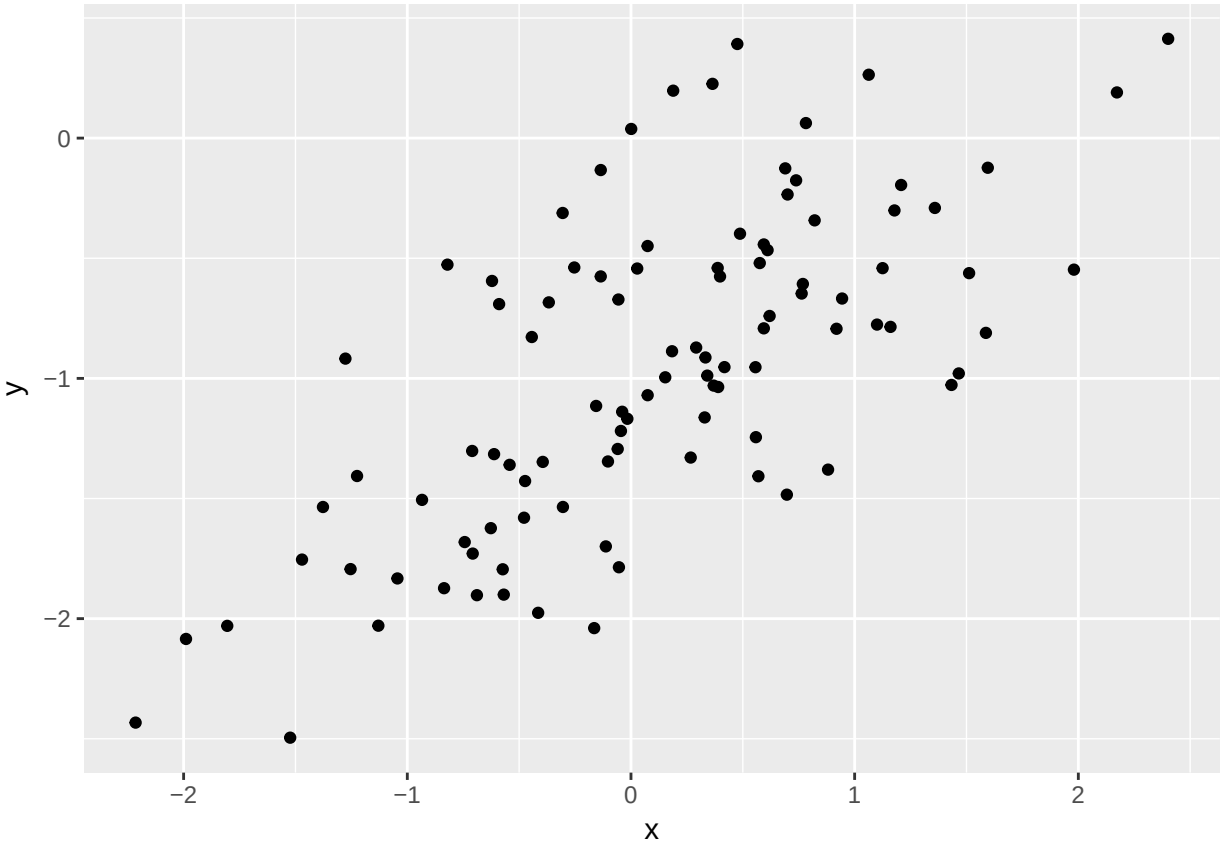
```
eps <- rnorm(100, 0, 0.5) # rnorm(n, mean = 0, sd = 1)
```

(c)

```
y <- -1 + 0.5*x + eps
```

(d)

```
ggplot(data= as.data.frame(cbind(x,y)), aes(x = x, y= y)) + geom_point()
```



It roughly looks like a linear line. We can see that x and y has some linear relationship.

(e)

```
lm <- lm(y ~ x)
cat("the original beta0 : " , -1, "\n")
```

```
## the original beta0 :  -1
```

```
cat("beta0 hat : ",lm$coefficients[1],"\n")
```

```
## beta0 hat :  -1.018846
```

```
cat("the original beta1 : " , 0.5, "\n")
```

```
## the original beta1 :  0.5
```

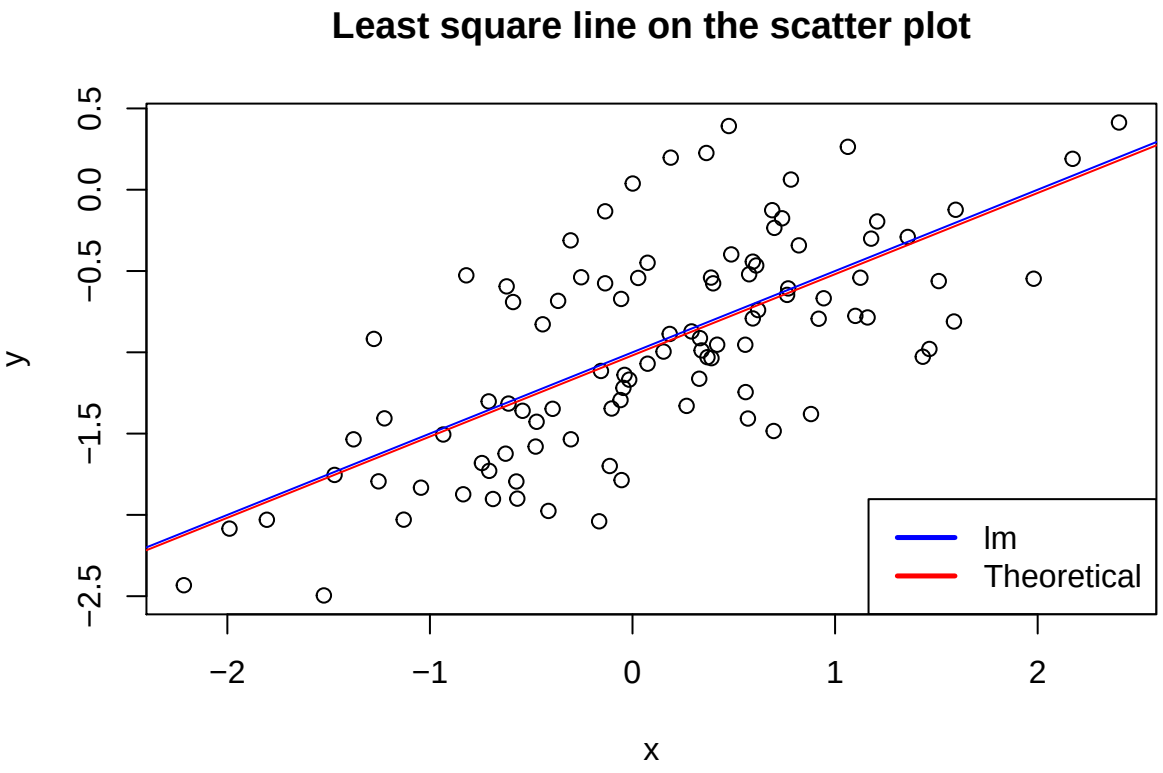
```
cat("beta1 hat : ",lm$coefficients[2],"\n")
```

```
## beta1 hat :  0.4994698
```

The original coefficients and the estimated coefficients are fairly similar.

(f)

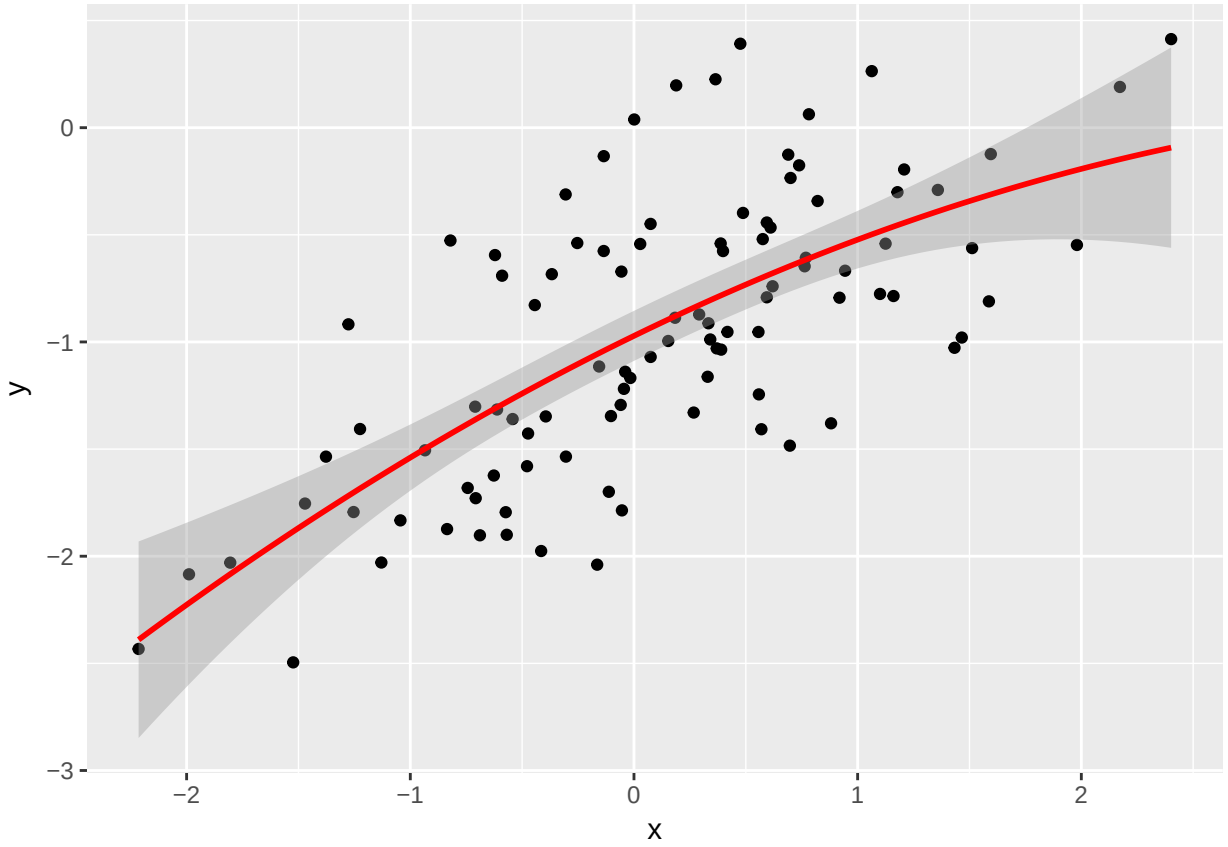
```
plot(x, y, main = "Least square line on the scatter plot")
abline(lm, col = "red")
abline(-1, 0.5, col = "blue")
legend("bottomright", legend = c("lm","Theoretical"),
       lty=c(1,1), # gives the legend appropriate symbols (lines)
       lwd=c(2.5,2.5),
       col=c("blue","red"))
```



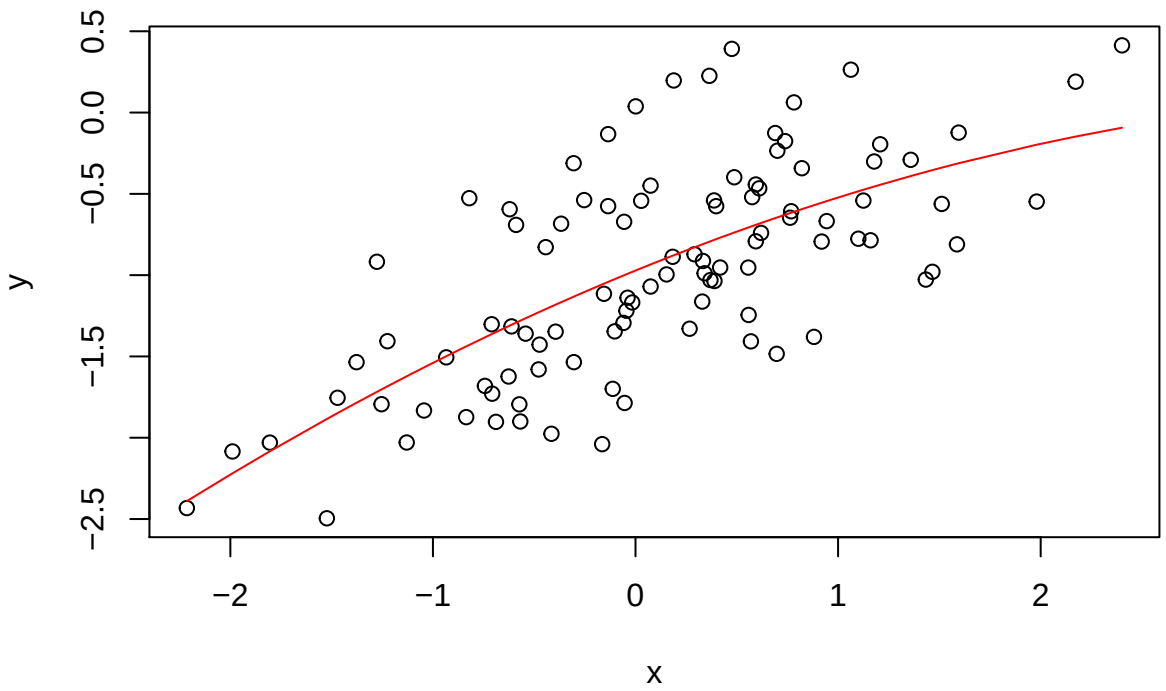
```
#ggplot(data= as.data.frame(cbind(x,y)), aes(x = x, y= y)) + geom_point()+   stat_smooth(method = "lm",
```

(g)

```
lm2 <- lm(y ~ x + I(x^2))
#summary(lm(y ~ poly(x, 2, raw = T)))


y2 <- lm2$coefficients[1] + lm2$coefficients[2]*x +
  lm2$coefficients[3]*x^2


ggplot(data= as.data.frame(cbind(x,y)), aes(x = x, y= y)) + geom_point()+stat_smooth(method = "lm", col
```




```
plot(x, y, main = "Polynomial regression")
smoothingSpline = smooth.spline(x, y2, spar = 0.5)
lines(smoothingSpline, col = "red")
```




```
summary(lm2)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98252 -0.31270 -0.06441  0.29014  1.13500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.97164    0.05883 -16.517  < 2e-16 ***
## x            0.50858    0.05399   9.420  2.4e-15 ***
## I(x^2)      -0.05946    0.04238  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.479 on 97 degrees of freedom
## Multiple R-squared:  0.4779, Adjusted R-squared:  0.4672
## F-statistic:  44.4 on 2 and 97 DF,  p-value: 2.038e-14
```

```
summary(lm)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93842 -0.30688 -0.06975  0.26970  1.17309
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01885    0.04849 -21.010  < 2e-16 ***
## x            0.49947    0.05386   9.273 4.58e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4814 on 98 degrees of freedom
## Multiple R-squared:  0.4674, Adjusted R-squared:  0.4619
## F-statistic: 85.99 on 1 and 98 DF,  p-value: 4.583e-15
```

It is not obvious if the quadratic term improves model fit or not. As we can see in summary of lm2 and lm, p-value gets bigger when we add quadratic terms, it suggests that it is probably not a good model to fit.

(h)

```
#Part a
set.seed(1)
x2 <- rnorm(100, 0, 1)
x2
```

```
##   [1] -0.626453811  0.183643324 -0.835628612  1.595280802  0.329507772
##   [6] -0.820468384  0.487429052  0.738324705  0.575781352 -0.305388387
##  [11]  1.511781168  0.389843236 -0.621240581 -2.214699887  1.124930918
##  [16] -0.044933609 -0.016190263  0.943836211  0.821221195  0.593901321
##  [21]  0.918977372  0.782136301  0.074564983 -1.989351696  0.619825748
##  [26] -0.056128740 -0.155795507 -1.470752384 -0.478150055  0.417941560
##  [31]  1.358679552 -0.102787727  0.387671612 -0.053805041 -1.377059557
##  [36] -0.414994563 -0.394289954 -0.059313397  1.100025372  0.763175748
##  [41] -0.164523596 -0.253361680  0.696963375  0.556663199 -0.688755695
##  [46] -0.707495157  0.364581962  0.768532925 -0.112346212  0.881107726
##  [51]  0.398105880 -0.612026393  0.341119691 -1.129363096  1.433023702
##  [56]  1.980399899 -0.367221476 -1.044134626  0.569719627 -0.135054604
##  [61]  2.401617761 -0.039240003  0.689739362  0.028002159 -0.743273209
##  [66]  0.188792300 -1.804958629  1.465554862  0.153253338  2.172611670
##  [71]  0.475509529 -0.709946431  0.610726353 -0.934097632 -1.253633400
##  [76]  0.291446236 -0.443291873  0.001105352  0.074341324 -0.589520946
##  [81] -0.568668733 -0.135178615  1.178086997 -1.523566800  0.593946188
##  [86]  0.332950371  1.063099837 -0.304183924  0.370018810  0.267098791
##  [91] -0.542520031  1.207867806  1.160402616  0.700213650  1.586833455
##  [96]  0.558486426 -1.276592208 -0.573265414 -1.224612615 -0.473400636
```

```
#Part b
eps2 <- rnorm(100, 0, 0.1) #sd^2 = var
eps2
```

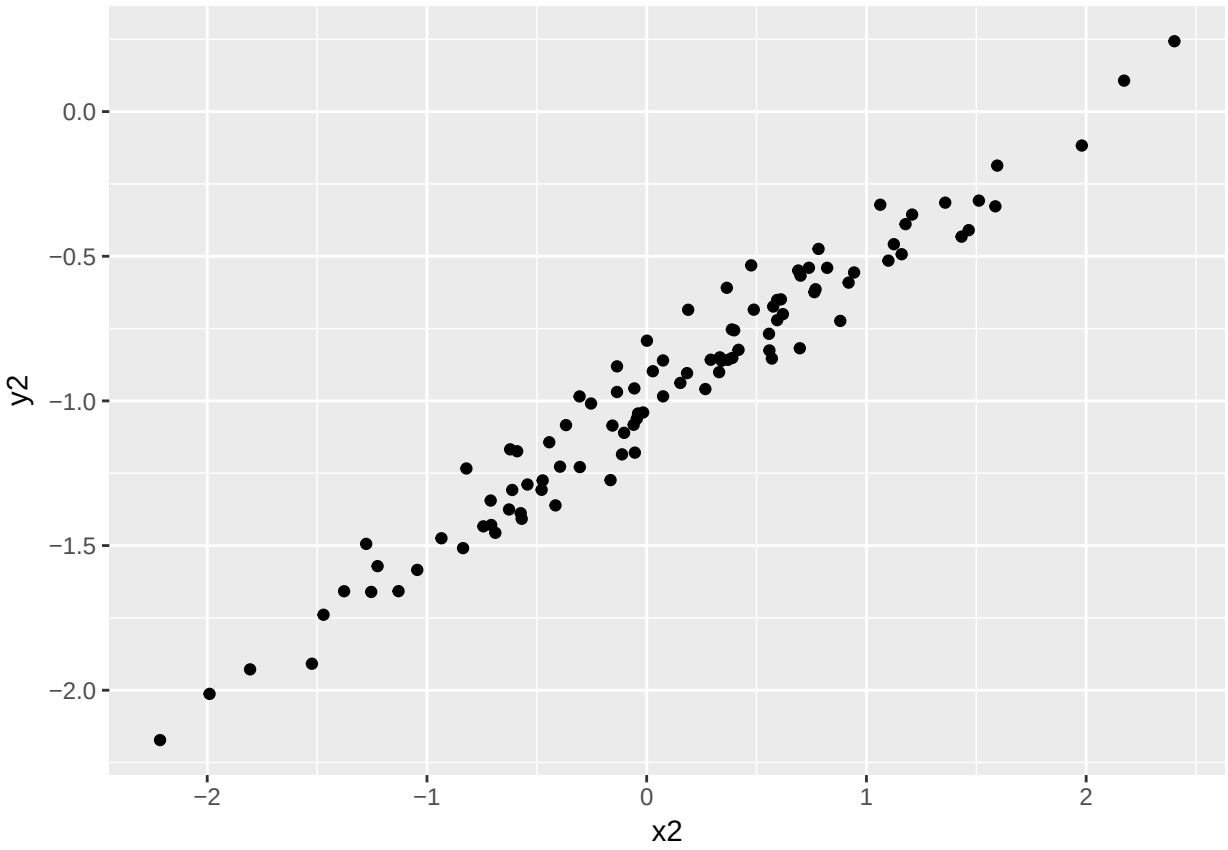
```
##   [1] -0.062036668  0.004211587 -0.091092165  0.015802877 -0.065458464
##   [6]  0.176728727  0.071670748  0.091017423  0.038418536  0.168217608
##  [11] -0.063573645 -0.046164473  0.143228224 -0.065069635 -0.020738074
##  [16] -0.039280793 -0.031999287 -0.027911330  0.049418833 -0.017733048
##  [21] -0.050595746  0.134303883 -0.021457941 -0.017955653 -0.010019074
##  [26]  0.071266631 -0.007356440 -0.003763417 -0.068166048 -0.032427027
##  [31]  0.006016044 -0.058889449  0.053149619 -0.151839408  0.030655786
##  [36] -0.153644982 -0.030097613 -0.052827990 -0.065209478 -0.005689678
```

```
##  [41] -0.191435943  0.117658331 -0.166497244 -0.046353040 -0.111592011
##  [46] -0.075081900  0.208716655  0.001739562 -0.128630053 -0.164060553
##  [51]  0.045018710 -0.001855983 -0.031806837 -0.092936215 -0.148746031
##  [56] -0.107519230  0.100002880 -0.062126669 -0.138442685  0.186929062
##  [61]  0.042510038 -0.023864710  0.105848305  0.088642265 -0.061924305
##  [66]  0.220610246 -0.025502703 -0.142449465 -0.014439960  0.020753834
##  [71]  0.230797840  0.010580237  0.045699881 -0.007715294 -0.033400084
##  [76] -0.003472603  0.078763961  0.207524501  0.102739244  0.120790840
##  [81] -0.123132342  0.098389557  0.021992480 -0.146725003  0.052102274
##  [86] -0.015875460  0.146458731 -0.076608200 -0.043021175 -0.092610950
##  [91] -0.017710396  0.040201178 -0.073174817  0.083037317 -0.120808279
##  [96] -0.104798441  0.144115771 -0.101584747  0.041197471 -0.038107605
```

```
#Part c
y2 <- -1 + 0.5*x2 + eps2
y2
```

```
##   [1] -1.3752636 -0.9039668 -1.5089065 -0.1865567 -0.9007046 -1.2335055
##   [7] -0.6846147 -0.5398202 -0.6736908 -0.9844766 -0.3076831 -0.8512429
##  [13] -1.1673921 -2.1724196 -0.4582726 -1.0617476 -1.0400944 -0.5559932
##  [19] -0.5399706 -0.7207824 -0.5911071 -0.4746280 -0.9841754 -2.0126315
##  [25] -0.7001062 -0.9567977 -1.0852542 -1.7391396 -1.3072411 -0.8234562
##  [31] -0.3146442 -1.1102833 -0.7530146 -1.1787419 -1.6578740 -1.3611423
##  [37] -1.2272426 -1.0824847 -0.5151968 -0.6241018 -1.2736977 -1.0090225
##  [43] -0.8180156 -0.7680214 -1.4559699 -1.4288295 -0.6089924 -0.6139940
##  [49] -1.1848032 -0.7235067 -0.7559283 -1.3078692 -0.8612470 -1.6576178
##  [55] -0.4322342 -0.1173193 -1.0836079 -1.5841940 -0.8535829 -0.8805982
##  [61]  0.2433189 -1.0434847 -0.5492820 -0.8973567 -1.4335609 -0.6849936
##  [67] -1.9279820 -0.4096720 -0.9378133  0.1070597 -0.5314474 -1.3443930
##  [73] -0.6489369 -1.4747641 -1.6602168 -0.8577495 -1.1428820 -0.7919228
##  [79] -0.8600901 -1.1739696 -1.4074667 -0.9691998 -0.3889640 -1.9085084
##  [85] -0.6509246 -0.8494003 -0.3219914 -1.2287002 -0.8580118 -0.9590616
##  [91] -1.2889704 -0.3558649 -0.4929735 -0.5668559 -0.3273916 -0.8255552
##  [97] -1.4941803 -1.3882175 -1.5711088 -1.2748079
```

```
#Part d

ggplot(data= as.data.frame(cbind(x2,y2)), aes(x = x2, y= y2)) + geom_point()
```




```
#Part e

lm <- lm(y2 ~ x2)
cat("the original beta0 : " , -1, "\n")
```

```
## the original beta0 :  -1
```

```
cat("beta0 hat : ",lm$coefficients[1],"\n")
```

```
## beta0 hat :  -1.003769
```

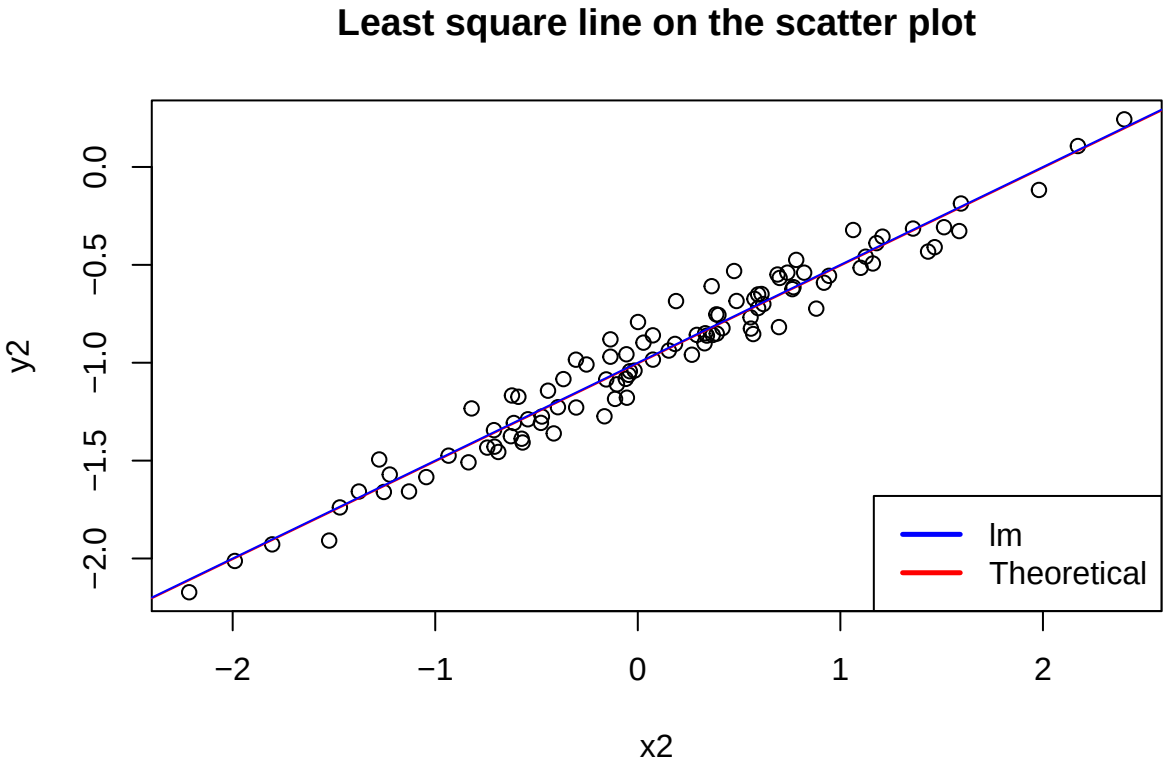
```
cat("the original beta1 : " , 0.5, "\n")
```

```
## the original beta1 :  0.5
```

```
cat("beta1 hat : ",lm$coefficients[2],"\n")
```

```
## beta1 hat :  0.499894
```

```
#Part f

plot(x2, y2, main = "Least square line on the scatter plot")
abline(lm, col = "red")
abline(-1, 0.5, col = "blue")
legend("bottomright", legend = c("lm","Theoretical"),
       lty=c(1,1), # gives the legend appropriate symbols (lines)
       lwd=c(2.5,2.5),
       col=c("blue","red"))
```



When there is less noise in the data, least square linear model fits well.

(i)

```
#Part a
set.seed(1)
x2 <- rnorm(100, 0, 1)
x2
```

```
##   [1] -0.626453811  0.183643324 -0.835628612  1.595280802  0.329507772
##   [6] -0.820468384  0.487429052  0.738324705  0.575781352 -0.305388387
##  [11]  1.511781168  0.389843236 -0.621240581 -2.214699887  1.124930918
##  [16] -0.044933609 -0.016190263  0.943836211  0.821221195  0.593901321
##  [21]  0.918977372  0.782136301  0.074564983 -1.989351696  0.619825748
##  [26] -0.056128740 -0.155795507 -1.470752384 -0.478150055  0.417941560
##  [31]  1.358679552 -0.102787727  0.387671612 -0.053805041 -1.377059557
##  [36] -0.414994563 -0.394289954 -0.059313397  1.100025372  0.763175748
##  [41] -0.164523596 -0.253361680  0.696963375  0.556663199 -0.688755695
##  [46] -0.707495157  0.364581962  0.768532925 -0.112346212  0.881107726
##  [51]  0.398105880 -0.612026393  0.341119691 -1.129363096  1.433023702
##  [56]  1.980399899 -0.367221476 -1.044134626  0.569719627 -0.135054604
##  [61]  2.401617761 -0.039240003  0.689739362  0.028002159 -0.743273209
##  [66]  0.188792300 -1.804958629  1.465554862  0.153253338  2.172611670
##  [71]  0.475509529 -0.709946431  0.610726353 -0.934097632 -1.253633400
##  [76]  0.291446236 -0.443291873  0.001105352  0.074341324 -0.589520946
##  [81] -0.568668733 -0.135178615  1.178086997 -1.523566800  0.593946188
##  [86]  0.332950371  1.063099837 -0.304183924  0.370018810  0.267098791
##  [91] -0.542520031  1.207867806  1.160402616  0.700213650  1.586833455
##  [96]  0.558486426 -1.276592208 -0.573265414 -1.224612615 -0.473400636
```

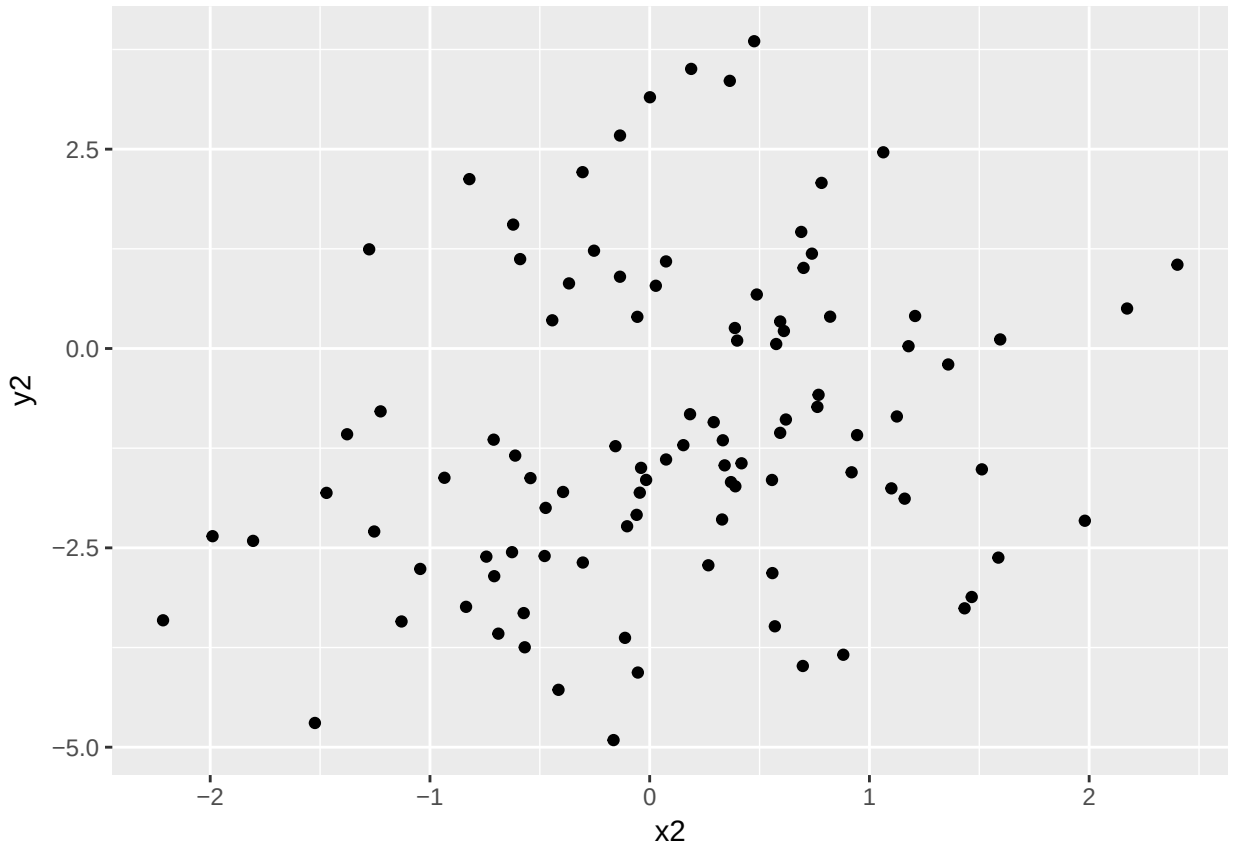
```r
#Part b
eps2 <- rnorm(100, 0, 2) #sd^2 = var
eps2
```

```
##   [1] -1.24073335  0.08423175 -1.82184330  0.31605754 -1.30916929
##   [6]  3.53457454  1.43341495  1.82034846  0.76837072  3.36435216
##  [11] -1.27147291 -0.92328946  2.86456448 -1.30139271 -0.41476149
##  [16] -0.78561586 -0.63998574 -0.55822661  0.98837666 -0.35466096
##  [21] -1.01191492  2.68607765 -0.42915882 -0.35911306 -0.20038148
##  [26]  1.42533261 -0.14712881 -0.07526834 -1.36332096 -0.64854054
##  [31]  0.12032088 -1.17778897  1.06299239 -3.03678816  0.61311572
##  [36] -3.07289965 -0.60195225 -1.05655981 -1.30418956 -0.11379356
##  [41] -3.82871885  2.35316662 -3.32994487 -0.92706080 -2.23184021
##  [46] -1.50163800  4.17433309  0.03479124 -2.57260106 -3.28121107
##  [51]  0.90037420 -0.03711967 -0.63613675 -1.85872429 -2.97492062
##  [56] -2.15038459  2.00005761 -1.24253339 -2.76885369  3.73858124
##  [61]  0.85020075 -0.47729420  2.11696610  1.77284530 -1.23848610
##  [66]  4.41220493 -0.51005406 -2.84898930 -0.28879920  0.41507668
##  [71]  4.61595680  0.21160474  0.91399761 -0.15430587 -0.66800168
##  [76] -0.06945206  1.57527921  4.15049002  2.05478488  2.41581680
##  [81] -2.46264684  1.96779114  0.43984961 -2.93450006  1.04204549
##  [86] -0.31750921  2.92917462 -1.53216400 -0.86042351 -1.85221899
##  [91] -0.35420792  0.80402356 -1.46349635  1.66074634 -2.41616557
##  [96] -2.09596883  2.88231541 -2.03169493  0.82394942 -0.76215210
```

```r
#Part c
y2 <- -1 + 0.5*x2 + eps2
y2
```

```
##   [1] -2.55396026 -0.82394659 -3.23965760  0.11369795 -2.14441540
##   [6]  2.12434035  0.67712948  1.18951081  0.05626139  2.21165797
##  [11] -1.51558232 -1.72836784  1.55394419 -3.40874265 -0.85229603
##  [16] -1.80808266 -1.64808087 -1.08630850  0.39898726 -1.05771030
##  [21] -1.55242624  2.07714580 -1.39187633 -2.35378891 -0.89046861
##  [26]  0.39726824 -1.22502656 -1.81064453 -2.60239599 -1.43956976
##  [31] -0.20033934 -2.22918284  0.25682819 -4.06369068 -1.07541406
##  [36] -4.28039693 -1.79909723 -2.08621651 -1.75417688 -0.73220568
##  [41] -4.91098065  1.22648578 -3.98146318 -1.64872920 -3.57621806
##  [46] -2.85538558  3.35662407 -0.58094230 -3.62877417 -3.84065721
##  [51]  0.09942714 -1.34313286 -1.46557690 -3.42340584 -3.25840877
##  [56] -2.16018464  0.81644687 -2.76460070 -3.48399388  2.67105394
##  [61]  1.05100963 -1.49691420  1.46183578  0.78684638 -2.61012270
##  [66]  3.50660108 -2.41253337 -3.11621187 -1.21217253  0.50138251
##  [71]  3.85371156 -1.14336848  0.21936079 -1.62135469 -2.29481838
##  [76] -0.92372894  0.35363327  3.15104269  1.09195554  1.12105632
##  [81] -3.74698121  0.90020183  0.02889311 -4.69628346  0.33901858
##  [86] -1.15103402  2.46072454 -2.68425596 -1.67541410 -2.71866960
##  [91] -1.62546794  0.40795746 -1.88329504  1.01085316 -2.62274885
##  [96] -2.81672561  1.24401931 -3.31832764 -0.78835688 -1.99885242
```

```r
#Part d

ggplot(data= as.data.frame(cbind(x2,y2)), aes(x = x2, y= y2)) + geom_point()
```




```
#Part e

lm <- lm(y2 ~ x2)
cat("the original beta0 : " , -1, "\n")
```

```
## the original beta0 :  -1
```

```
cat("beta0 hat : ",lm$coefficients[1],"\n")
```

```
## beta0 hat :  -1.075385
```

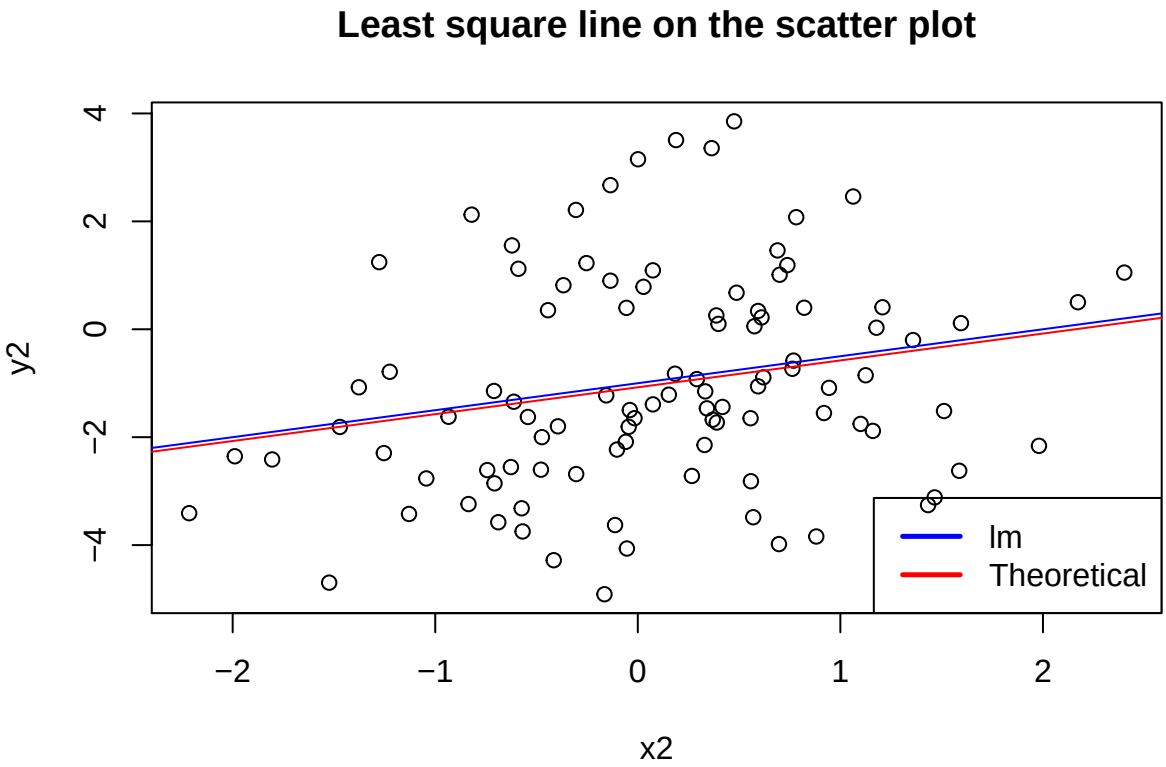
```
cat("the original beta1 : " , 0.5, "\n")
```

```
## the original beta1 :  0.5
```

```
cat("beta1 hat : ",lm$coefficients[2],"\n")
```

```
## beta1 hat :  0.4978792
```

```
#Part f

plot(x2, y2, main = "Least square line on the scatter plot")
abline(lm, col = "red")
abline(-1, 0.5, col = "blue")
legend("bottomright", legend = c("lm","Theoretical"),
       lty=c(1,1), # gives the legend appropriate symbols (lines)
       lwd=c(2.5,2.5),
       col=c("blue","red"))
```



When there is more noise in the data, least square linear model doesn't fit well and it seems like an inefficient way to predict model.

4

```
ols_fit <- function(X, y){
  coefficients <- solve(t(X) %*% X) %*% t(X) %*% y
  y_values <- y
  fitted_values <- X %*% coefficients
  residuals <- y - fitted_values
  n <- nrow(X)
  q <- ncol(X)
  returnvalue <- list(coefficients = coefficients, y_values = y_values, fitted_values = fitted_values, 
  return(returnvalue)
}


X <- as.matrix(cbind(1,mtcars[,c(3,4),drop = F]))
y <- as.matrix(mtcars[,1, drop = F])


fit <- ols_fit(X, y)
names(fit)
```

```
## [1] "coefficients"  "y_values"      "fitted_values" "residuals"
## [5] "n"             "q"
```

```
fit$coefficients
```

```
##             mpg
```

```
## 1    30.73590425
## disp -0.03034628
## hp   -0.02484008
```

```
summary(fit$fitted_values)
```

```
##       mpg
##  Min.   :11.32
##  1st Qu.:15.16
##  Median :21.64
##  Mean   :20.09
##  3rd Qu.:24.70
##  Max.   :27.15
```

```
summary(fit$residuals)    # Q : slightly off?
```

```
##       mpg
##  Min.   :-4.7945
##  1st Qu.:-2.3036
##  Median :-0.8246
##  Mean   : 0.0000
##  3rd Qu.: 1.8582
##  Max.   : 6.9363
```

5

(a)

```
R2 <- function(fit){
  r2 <- sum((fit$fitted_values - mean(fit$y_values))^2) /
    sum((fit$y_values - mean(fit$y_values))^2)
  return(r2)
}

R2(fit)
```

```
## [1] 0.7482402
```

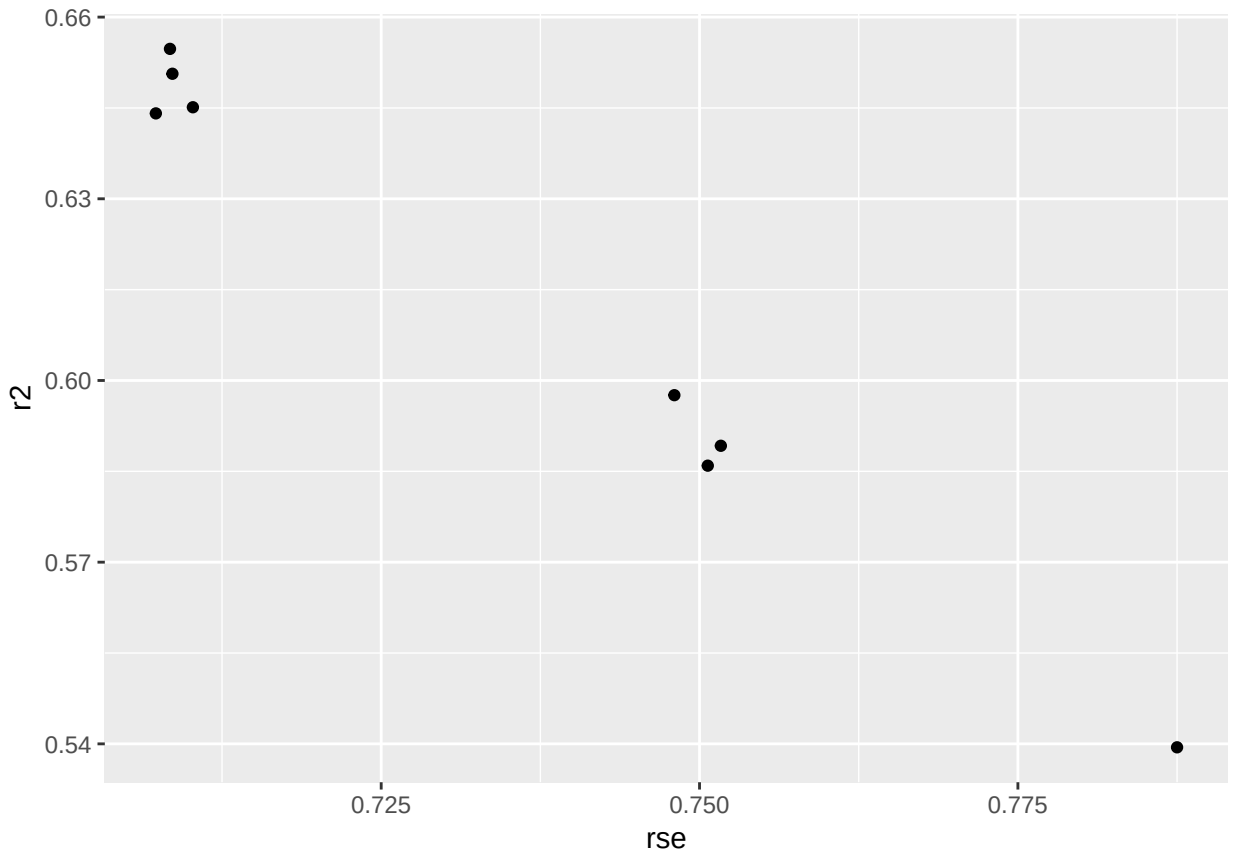
(b)

```
RSE <- function(fit){
  rse <- sqrt(sum((fit$y_values - fit$fitted_values)^2) / (fit$n - fit$q))
  return(rse)

}

RSE(fit)
```

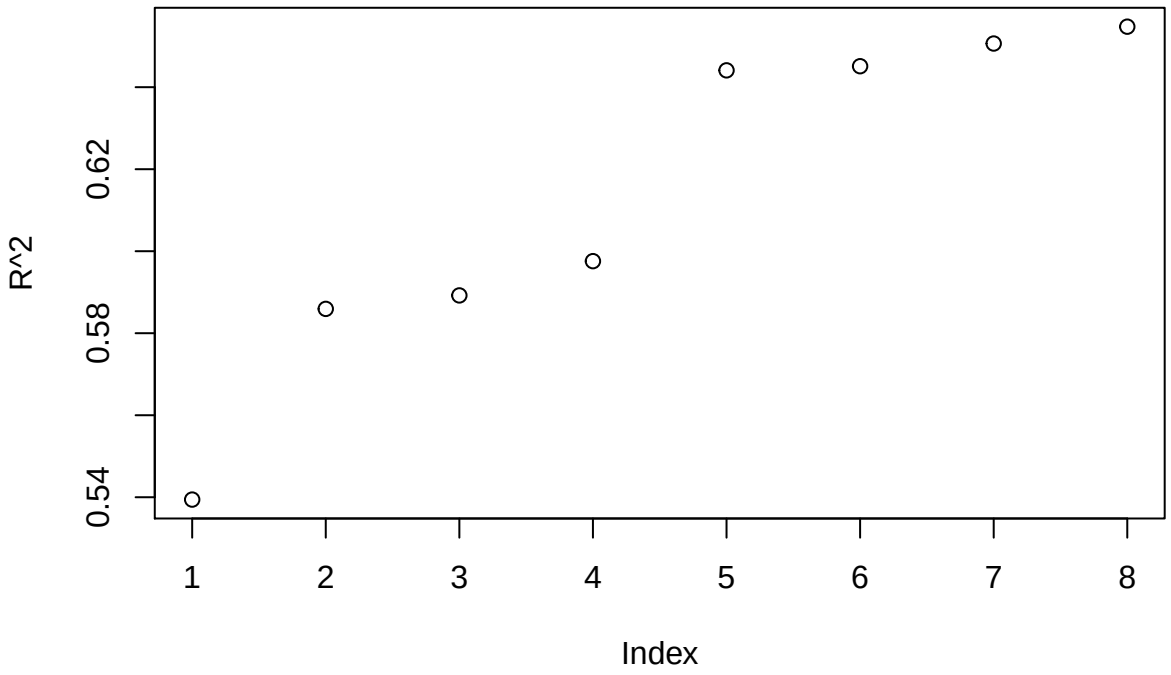
```
## [1] 3.126601
```

6

```
prostate <- read.table("/Users/cloverjiyoon/2017Fall/Stat 154/data/prostate.txt")

y <- as.matrix(prostate[,c("lpsa"), drop = F])
X <- as.matrix(cbind(1, prostate[,c(1), drop = F]))  # lcavol as predictor

fit <- ols_fit(X, y)
r2 <- c(0)
rse <- c(0)
r2[1] <- R2(fit)
rse[1] <- RSE(fit)

for(i in 2:ncol(prostate)-1){
  X <- as.matrix(cbind(1, prostate[,c(1:i), drop = F]))    # Q or c(1,i)?
  fit <- ols_fit(X, y)
  r2[i] <- R2(fit)
  rse[i] <- RSE(fit)
}

ggplot(data = as.data.frame(cbind(r2,rse)), aes(x = rse, y = r2)) + geom_point()
```



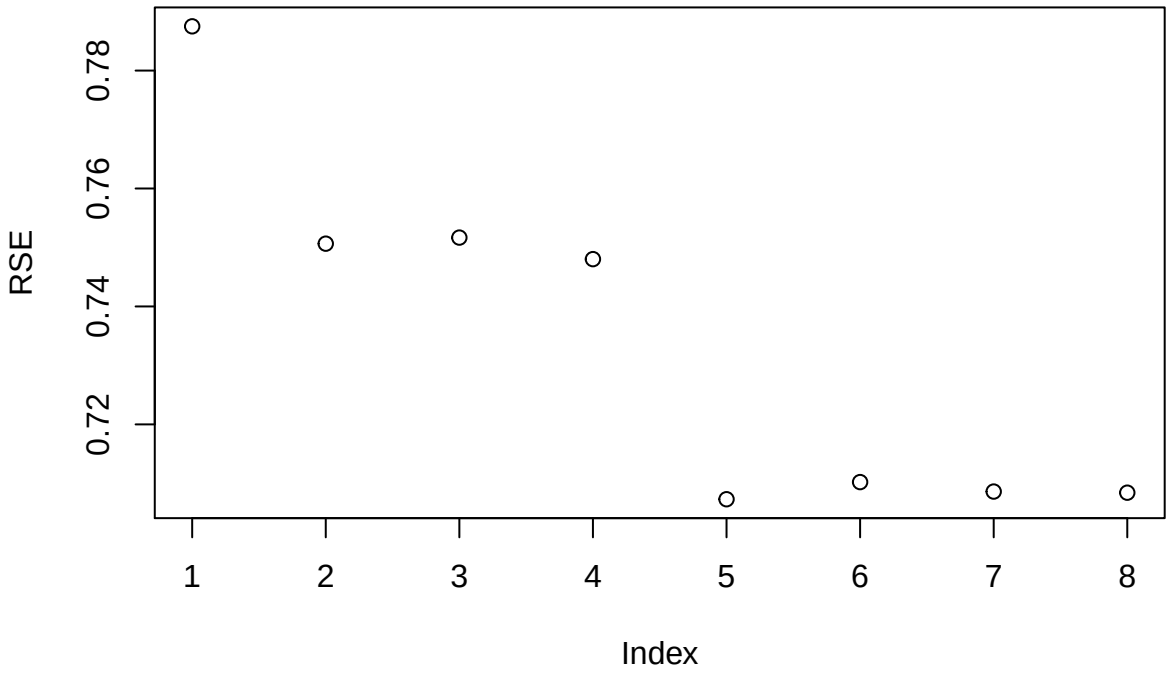

```
plot(r2, ylab = "R^2")
```




```
r2
```

```
## [1] 0.5394319 0.5859345 0.5892177 0.5975750 0.6441024 0.6451130 0.6506440
## [8] 0.6547541
```

```
plot(rse, ylab = "RSE")
```



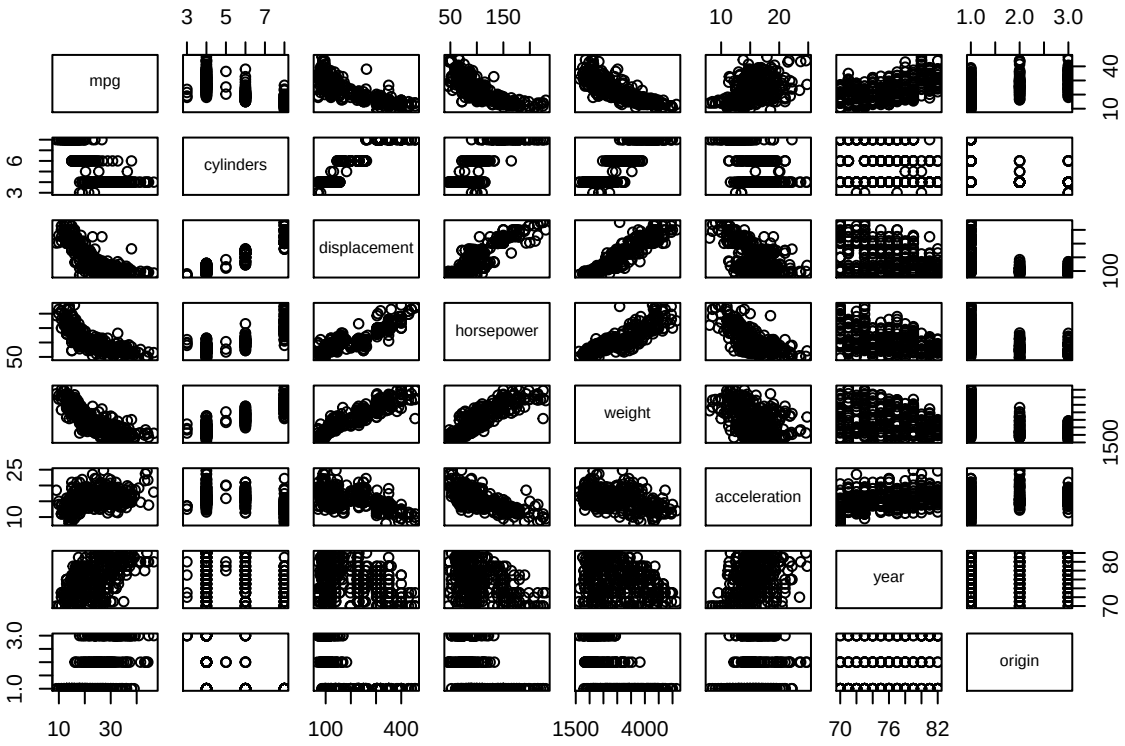
7

(a)

```
auto <- read.table("/Users/cloverjiyoon/2017Fall/Stat 154/data/Auto.data.txt", header = T, stringsAsFac


auto$horsepower <- as.numeric(auto$horsepower)
```

```
## Warning: NAs introduced by coercion
```

```
auto <- na.omit(auto)    # Q

auto <- auto[,-9]

pairs(auto)
```



(b)

```
cor <- cor(auto)
cor
```

```
##                     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)

```
fit <- lm(mpg ~., data = auto)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ ., 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
```

Is there a relationship between the predictors and the response?

The p-values of cylinders, horsepower, and acceleration variables are high and it means these variables are not significant to form a a good linear relationship with the response when (significance level) $\alpha = 0.01$.

Which predictors appear to have a statistically significant relationship to the response?

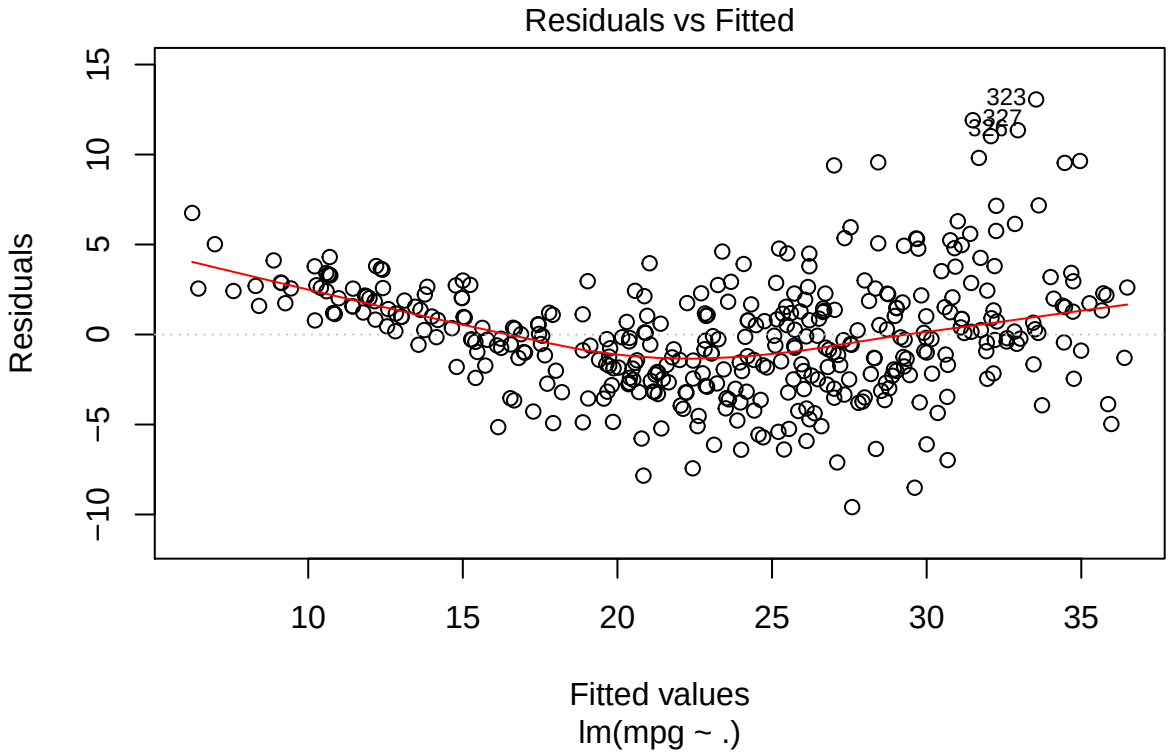
By p-values, weight and year variables are the most significant predictors.

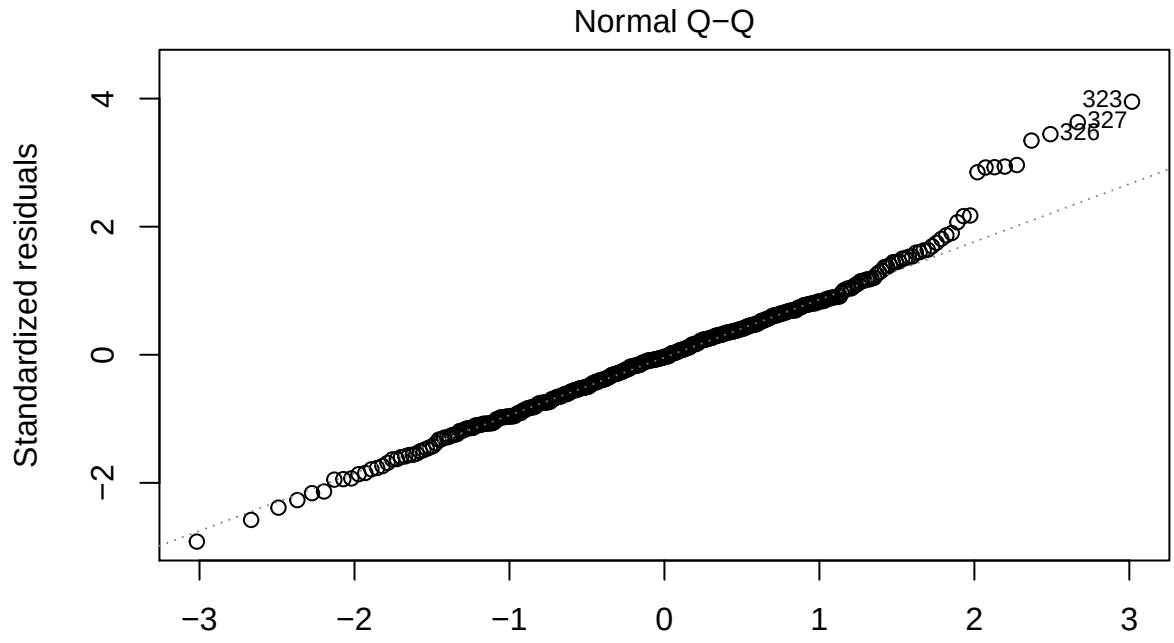
What does the coefficient for the year variable suggest?

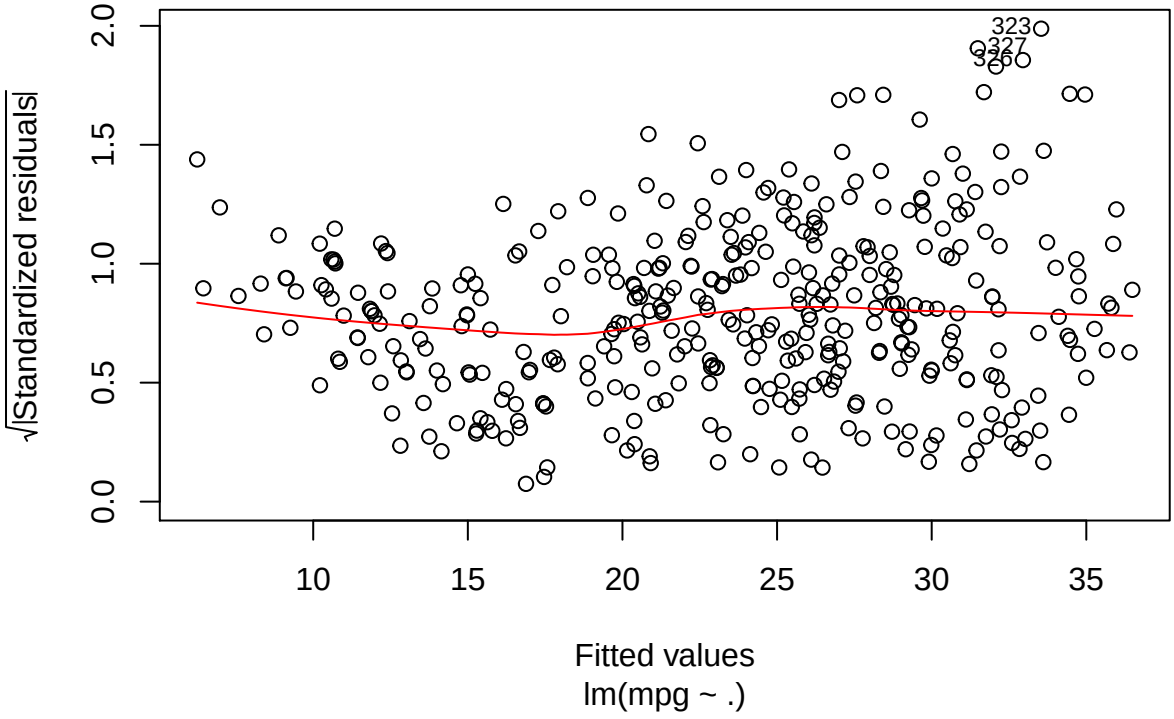
When all the other variables are fixed, if year increase 1 unit(1 year), mpg will increase 0.75 unit.

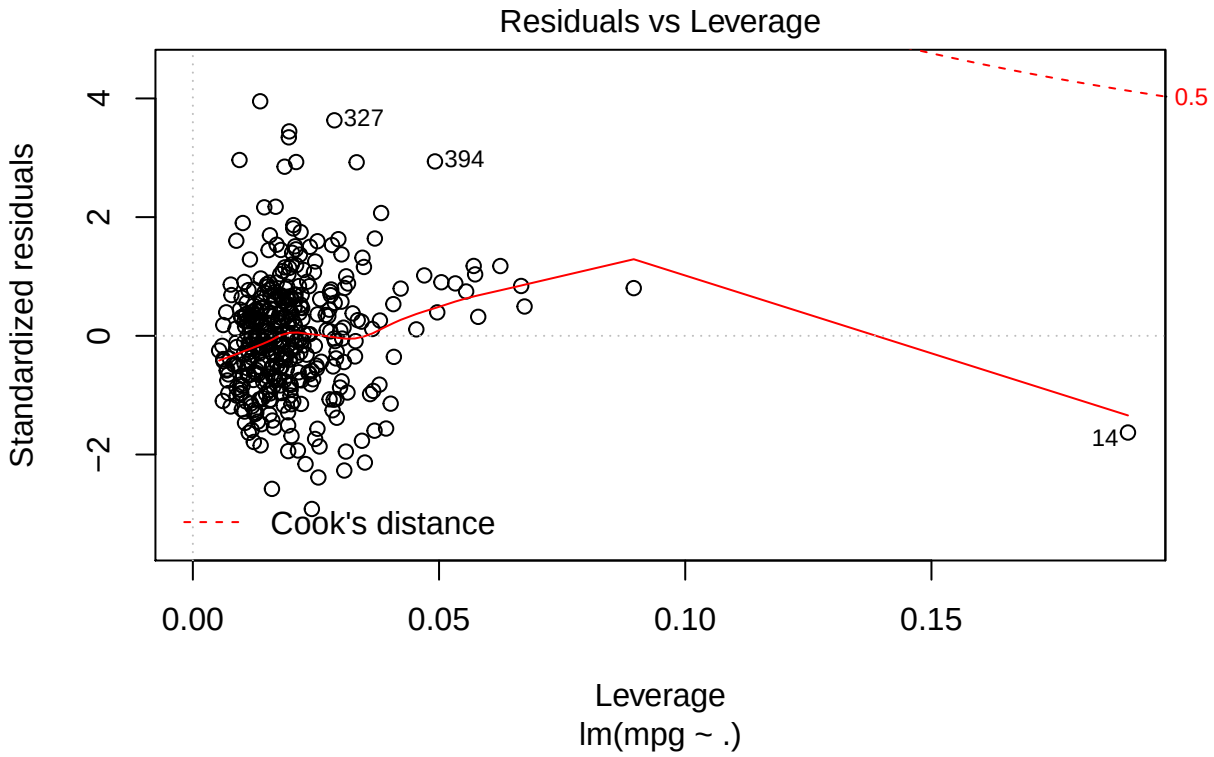
(d)

```
plot(fit)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(mpg ~ .)
323
327
326


Scale−Location
√|Standardized residuals|
Fitted values
lm(mpg ~ .)
323
327
326



There is a pattern in residuals vs fitted values plot. This suggests that the linear model we fitted violates the assumption for linear modeling (we assumed that $\hat{y}$ is independent with residuals). Also, the variance of residual seems to increase as fitted values increase. (heteroscedasitic)

There are some outliers, but they are not unusally large since the outliers are gradually increased as we can see in the plot.

Leverage plot identifies observation with unusual high leverage on the right bottom side (14 on the dot).

(e)

- – **include both of main/single and interaction effects** ∗ include only interaction effect.

```
# Q
fit <- lm(mpg ~.+ weight*year, data = auto)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ . + weight * year, data = auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9995 -1.8495 -0.1559  1.6061 11.7042
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.186e+02  1.338e+01  -8.864  < 2e-16 ***
## cylinders    -1.218e-01  3.032e-01  -0.402   0.6881
## displacement  1.293e-02  7.019e-03   1.842   0.0663 .
## horsepower   -2.877e-02  1.286e-02  -2.236   0.0259 *
```

```
## weight        3.044e-02  4.652e-03   6.543 1.94e-10 ***
## acceleration  1.447e-01  9.196e-02   1.574   0.1164
## year          2.084e+00  1.732e-01  12.033  < 2e-16 ***
## origin        1.174e+00  2.597e-01   4.519 8.30e-06 ***
## weight:year  -4.879e-04  6.097e-05  -8.002 1.47e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.084 on 383 degrees of freedom
## Multiple R-squared:  0.847,  Adjusted R-squared:  0.8439
## F-statistic: 265.1 on 8 and 383 DF,  p-value: < 2.2e-16
```

```
fit <- lm(mpg ~ weight:year, data = auto)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ weight:year, data = auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.3849  -3.3041  -0.5901   2.6158  17.5737
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.571e+01  9.581e-01   47.71   <2e-16 ***
## weight:year -9.882e-05  4.105e-06  -24.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.957 on 390 degrees of freedom
## Multiple R-squared:  0.5977, Adjusted R-squared:  0.5967
## F-statistic: 579.4 on 1 and 390 DF,  p-value: < 2.2e-16
```

Since Interaction term's p value is close to 0, it suggests that there is an interaction effect and it is statistically significant.

(f)

```
# log(X)
newdata <- log(auto)
fitnew <- lm(mpg ~., data = newdata)
summary(fitnew)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = newdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41298 -0.07098  0.00055  0.06150  0.39532
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  -0.155391   0.648230  -0.240  0.81068
## cylinders    -0.082815   0.061429  -1.348  0.17841
## displacement  0.006625   0.056970   0.116  0.90748
## horsepower   -0.294389   0.057652  -5.106 5.18e-07 ***
## weight       -0.569666   0.082397  -6.914 1.98e-11 ***
## acceleration -0.179239   0.059536  -3.011  0.00278 **
## year          2.243989   0.131661  17.044  < 2e-16 ***
## origin        0.044848   0.018821   2.383  0.01767 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1136 on 384 degrees of freedom
## Multiple R-squared:  0.8903, Adjusted R-squared:  0.8883
## F-statistic: 445.3 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
# sqrt(X)
newdata <- sqrt(auto)
fitnew <- lm(mpg ~., data = newdata)
summary(fitnew)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = newdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98667 -0.17280 -0.00315  0.16145  1.02245
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.949286   0.847481  -2.300 0.021979 *
## cylinders    -0.108552   0.141968  -0.765 0.444964
## displacement  0.019707   0.021182   0.930 0.352752
## horsepower   -0.090896   0.028428  -3.197 0.001502 **
## weight       -0.061414   0.007292  -8.422 7.48e-16 ***
## acceleration -0.107258   0.077048  -1.392 0.164699
## year          1.266015   0.079308  15.963  < 2e-16 ***
## origin        0.272324   0.070883   3.842 0.000143 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2964 on 384 degrees of freedom
## Multiple R-squared:  0.8662, Adjusted R-squared:  0.8638
## F-statistic: 355.1 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
# X^2
newdata <-as.data.frame(auto^2)
fitnew <- lm(mpg ~., data = newdata)
summary(fitnew)
```

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

```
##     Min      1Q  Median      3Q     Max
## -501.89 -145.36  -18.91  111.41 1034.08
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7.523e+02  1.456e+02  -5.165 3.87e-07 ***
## cylinders    -3.746e+00  1.559e+00  -2.403 0.016713 *
## displacement  3.356e-03  8.547e-04   3.926 0.000102 ***
## horsepower    1.279e-04  3.076e-03   0.042 0.966851
## weight       -4.833e-05  5.551e-06  -8.707  < 2e-16 ***
## acceleration  4.892e-01  1.663e-01   2.941 0.003474 **
## year          2.731e-01  2.183e-02  12.513  < 2e-16 ***
## origin        2.608e+01  4.275e+00   6.101 2.57e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 218.8 on 384 degrees of freedom
## Multiple R-squared:  0.7055, Adjusted R-squared:  0.7001
## F-statistic: 131.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

I tranformed both X and Y data by using the given form.

First transformation : Horsepower, weight, acceleration, and year are the significant variables

Second transformation : Horsepower, weight, year, and origin are the significant variables

Third transformation : Intercept, displacement, weight, acceleration, year, origin are the significant variables.

Third transformation leads to have the most significant variables among all those given transformation.