Assignment 1

1)- Use Auto data to answer following.

```
library(readr)
Auto <- read_csv("Auto.csv")
Auto$horsepower <- as.numeric(Auto$horsepower)
Auto$name <- as.factor(Auto$name)
str(Auto)
## spec_tbl_df [397 × 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ mpg         : num [1:397] 18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num [1:397] 8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num [1:397] 307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num [1:397] 130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num [1:397] 3504 3693 3436 3433 3449 ...
##  $ acceleration: num [1:397] 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num [1:397] 70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num [1:397] 1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36
231 14 161 141 54 223 241 2 ...
```

(a)

Use the lm() function to perform a simple linear regression with mpg as the response and horsepower as the predictor. Use the summary() function to print the results. Comment on the output.

```
auto_fit <- lm(mpg~horsepower, data = Auto)
summary(auto_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
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

From the results, the estimated equation would be as follows:

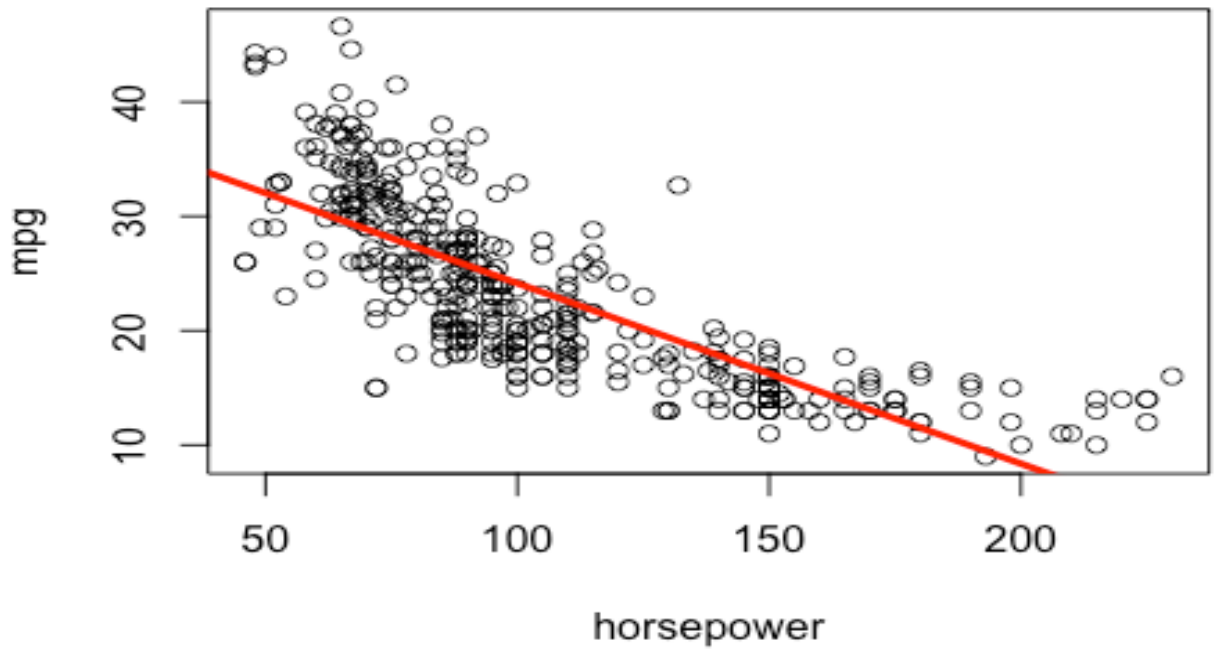
$$\widehat{\text{mpg}} = 39.936 - 0.158 \text{ horsepower}$$

The results indicate that the relationship between mpg and horsepower is negative which means if the horsepower increases by 1 unit, the mpg decreases by 0.158 miles per gallon. The R-Squared is 0.6059 meaning that 61% of the variation in the response variable (mpg) is explained by the predictor variable (horsepower). The coefficient of horsepower is statistically significant with a significance level of 0.05.

(b)

Plot the response and the predictor. Use the abline() function to display the least squares regression line.

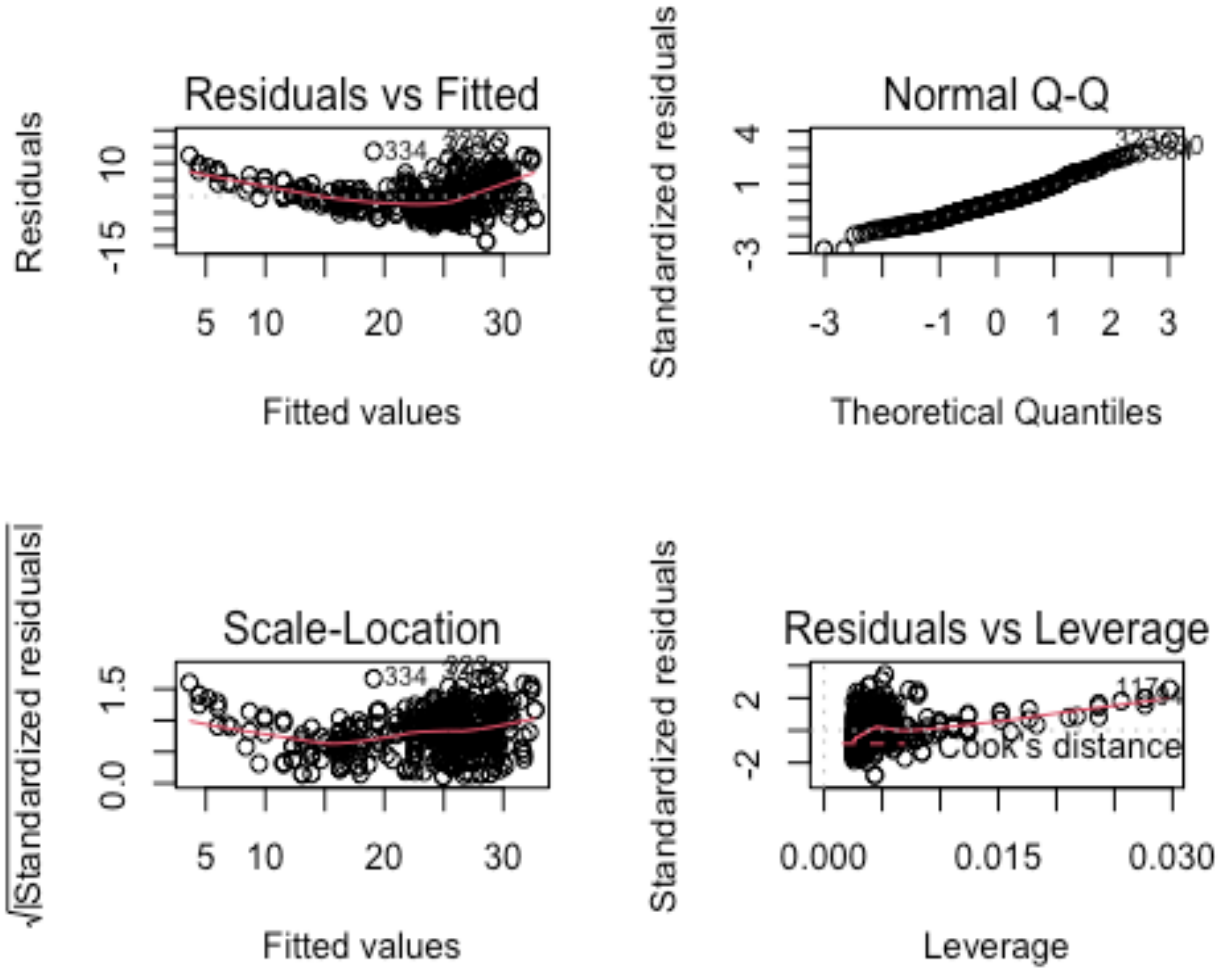
```
plot(mpg~horsepower, lwd = 0.75, data = Auto)
abline(auto_fit, lwd = 3, col = 'red')
```



(c)

Use the plot() function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.

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



From the plots, we can observe as follows:

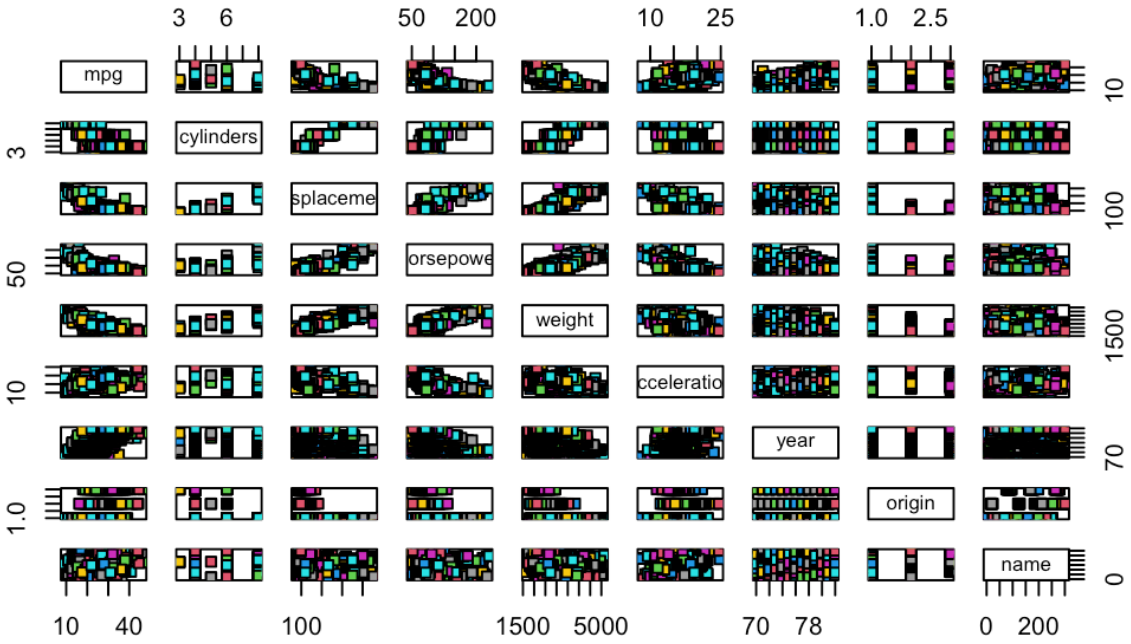
- The residual plot seems to be a pattern of U-shaped plot. This may suggest a non-linear relationship between the predictor and response variables.
- The QQ plot shows that the residuals are normally distributed.
- Finally, the residual vs. leverage plot indicates that there are few outliers and high leverage points in the data.

2)- Use Auto data to answer following.

(a)

Produce a scatterplot matrix which includes all of the variables in the data set.

```
pairs(Auto, bg = c(1:nrow(Auto)), pch = 22)
```



(b)

Compute the matrix of correlations between the variables using the function cor(). You will need to exclude the name variable, cor() which is qualitative.

```
cor(Auto[,!(names(Auto)=="name")])
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7762599   -0.8044430         NA -0.8317389
## cylinders    -0.7762599  1.0000000    0.9509199         NA  0.8970169
## displacement -0.8044430  0.9509199    1.0000000         NA  0.9331044
## horsepower           NA         NA           NA          1         NA
## weight       -0.8317389  0.8970169    0.9331044         NA  1.0000000
## acceleration  0.4222974 -0.5040606   -0.5441618         NA -0.4195023
## year          0.5814695 -0.3467172   -0.3698041         NA -0.3079004
## origin        0.5636979 -0.5649716   -0.6106643         NA -0.5812652
##               acceleration       year     origin
## mpg              0.4222974  0.5814695  0.5636979
## cylinders       -0.5040606 -0.3467172 -0.5649716
## displacement    -0.5441618 -0.3698041 -0.6106643
## horsepower              NA         NA         NA
## weight          -0.4195023 -0.3079004 -0.5812652
## acceleration     1.0000000  0.2829009  0.2100836
## year             0.2829009  1.0000000  0.1843141
## origin           0.2100836  0.1843141  1.0000000
```

(c)

Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summary() function to print the results. Comment on the output.

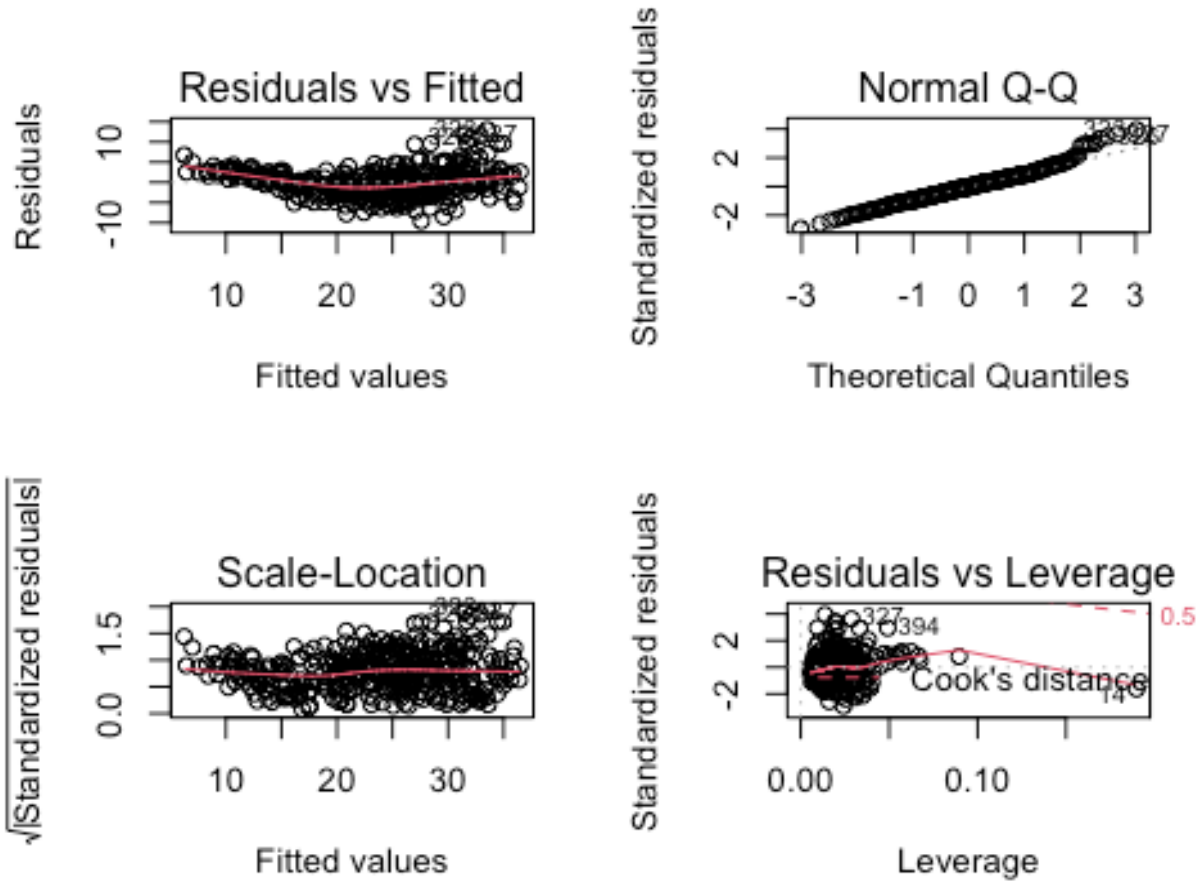
```
mlm_fit <- lm(mpg~.-name, data = Auto)
summary(mlm_fit)
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -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
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

The results indicate that the coefficients of displacement, weight, year, and origin are statistically significant with a significance level of 0.05. The R-Squared is 0.8215 meaning that 82% of the variation in the response variable (mpg) is explained by the predictor variables. This may suggest that this model fits better than the previous model which includes only horsepower.

(d)

Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

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



From the plots, we can observe as follows:

- The residual plot suggests a non-linear relationship between the predictor and response variables.
- The Q-Q plot shows that the residuals are normally distributed.
- The residuals and leverage plot indicates that there is one leverage point which is the observation 14, and there are a few outliers such as in case of standardized residuals are higher than 2.

(e)

Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

```
lm_fit2 <- lm(mpg~(.-name) *(.-name), data = Auto)
summary(lm_fit2)
##
## Call:
## lm(formula = mpg ~ (. - name) * (. - name), data = Auto)
##
## 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
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8893, Adjusted R-squared:  0.8808
## F-statistic: 104.2 on 28 and 363 DF,  p-value: < 2.2e-16
```

The model suggests an improvement in R^2 from 0.82 to 0.89. The most significant interaction term is acceleration and origin compare to other terms as it is statistically significant with a significance level of 0.01. In addition, the interaction terms of displacement and year, acceleration and origin are statistically significant with a significance level of 0.05. The interaction terms of cylinders and acceleration, cylinders and year, displacement and weight, horsepower and acceleration, and year and origin are statistically significant with a significance level of 0.1.

(f)

Try a few different transformations of the variables, such as log(X), square root of X, etc. Comment on your findings.

Log transformations of the variables

```
X<- Auto[-c(1,9)]
log.Auto <- data.frame(mpg = Auto$mpg, log = log(X))
lm.fit_log <- lm(mpg~., data = log.Auto)
summary(lm.fit_log)
##
## Call:
## lm(formula = mpg ~ ., data = log.Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5987 -1.8172 -0.0181  1.5906 12.8132
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -66.5643    17.5053  -3.803 0.000167 ***
## log.cylinders      1.4818     1.6589   0.893 0.372273
## log.displacement  -1.0551     1.5385  -0.686 0.493230
## log.horsepower    -6.9657     1.5569  -4.474 1.01e-05 ***
## log.weight       -12.5728     2.2251  -5.650 3.12e-08 ***
## log.acceleration  -4.9831     1.6078  -3.099 0.002082 **
## log.year          54.9857     3.5555  15.465  < 2e-16 ***
## log.origin         1.5822     0.5083   3.113 0.001991 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.069 on 384 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8482, Adjusted R-squared:  0.8454
## F-statistic: 306.5 on 7 and 384 DF,  p-value: < 2.2e-16
```

Square-Root transformations of the variables

```
sqrt.Auto <- data.frame(mpg = Auto$mpg, sqrt = sqrt(X))
lm.fit_sqrt <- lm(mpg~., data = sqrt.Auto)
summary(lm.fit_sqrt)
##
## Call:
## lm(formula = mpg ~ ., data = sqrt.Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5250 -1.9822 -0.1111  1.7347 13.0681
##
```

```
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -49.79814    9.17832  -5.426 1.02e-07 ***
## sqrt.cylinders     -0.23699    1.53753  -0.154   0.8776
## sqrt.displacement   0.22580    0.22940   0.984   0.3256
## sqrt.horsepower    -0.77976    0.30788  -2.533   0.0117 *
## sqrt.weight        -0.62172    0.07898  -7.872 3.59e-14 ***
## sqrt.acceleration  -0.82529    0.83443  -0.989   0.3233
## sqrt.year          12.79030    0.85891  14.891  < 2e-16 ***
## sqrt.origin         3.26036    0.76767   4.247 2.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.21 on 384 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8338, Adjusted R-squared:  0.8308
## F-statistic: 275.3 on 7 and 384 DF,  p-value: < 2.2e-16
```

Quadratic transformations of the variables

```
poly.Auto <- data.frame(mpg = Auto$mpg, poly = X^2)
lm.fit_poly <- lm(mpg~., data = poly.Auto)
summary(lm.fit_poly)
##
## Call:
## lm(formula = mpg ~ ., data = poly.Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6786 -2.3227 -0.0582  1.9073 12.9807
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.208e+00  2.356e+00   0.513 0.608382
## poly.cylinders    -8.829e-02  2.521e-02  -3.502 0.000515 ***
## poly.displacement  5.680e-05  1.382e-05   4.109 4.87e-05 ***
## poly.horsepower   -3.621e-05  4.975e-05  -0.728 0.467201
## poly.weight       -9.351e-07  8.978e-08 -10.416  < 2e-16 ***
## poly.acceleration  6.278e-03  2.690e-03   2.334 0.020130 *
## poly.year          4.999e-03  3.530e-04  14.160  < 2e-16 ***
## poly.origin        4.129e-01  6.914e-02   5.971 5.37e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.539 on 384 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.7981, Adjusted R-squared:  0.7944
## F-statistic: 216.8 on 7 and 384 DF,  p-value: < 2.2e-16
```

From the results, as we transformed independent variables using log, square-root, and square transformations, we can observe that the model with the log-transformation has the highest R-Squared, which is equal to 0.8482. This may suggest that the log-transformed model seems to be the most linear relationship. However, the coefficients of horsepower in the models with the square-root and square transformations are statistically significant, while the log-transformed model is not statistically significant.

3) Create simulated data and fit simple linear regression models to it. Make sure to use set.seed(1) prior to starting part (a) to ensure consistent results.

(a)

Using the rnorm() function, create a vector, x, containing 100 observations drawn from a $N(0,1)$ distribution. This represents a feature, X

```
set.seed(1)
x <- rnorm(100)
str(x)
##  num [1:100] -0.626 0.184 -0.836 1.595 0.33 ...
```

(b)

Using the rnorm() function, create a vector, eps, containing 100 observations drawn from a $N(0,0.25)$ distribution. i.e. a normal distribution with mean zero and variance 0.25.

```
eps <- rnorm(100, mean = 0, sd = sqrt(0.25))
str(eps)
##  num [1:100] -0.3102 0.0211 -0.4555 0.079 -0.3273 ...
```

(c)

Using x and eps, generate a vector y according to the model

$$Y = -1 + 0.5X + \epsilon$$

What is the length of the vector y? What are the values of β_0 and β_1 in this linear model?

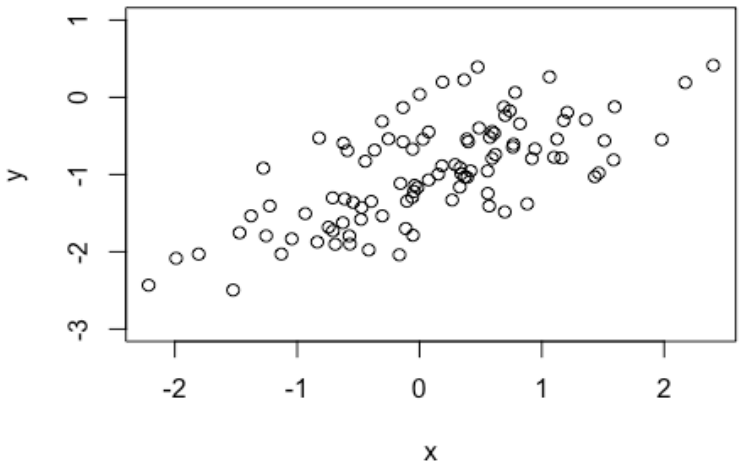
```
y = -1 + 0.5*x + eps
length(y)
## [1] 100
```

The length of the vector y is 100, and the values of β_0 and β_1 are -1 and 0.5 respectively.

(d)

Create a scatterplot displaying the relationship between x and y. Comment on what you observe.

```
plot(y~x,ylim = c(-3,1))
```



The plot indicates a linear relationship between x and y with some random noise created by adding the error term, eps.

(e)

Fit a least squares linear model to predict y using x. Comment on the model obtained. How do $\hat{\beta}_0$ and $\hat{\beta}_1$ compare to β_0 and β_1?

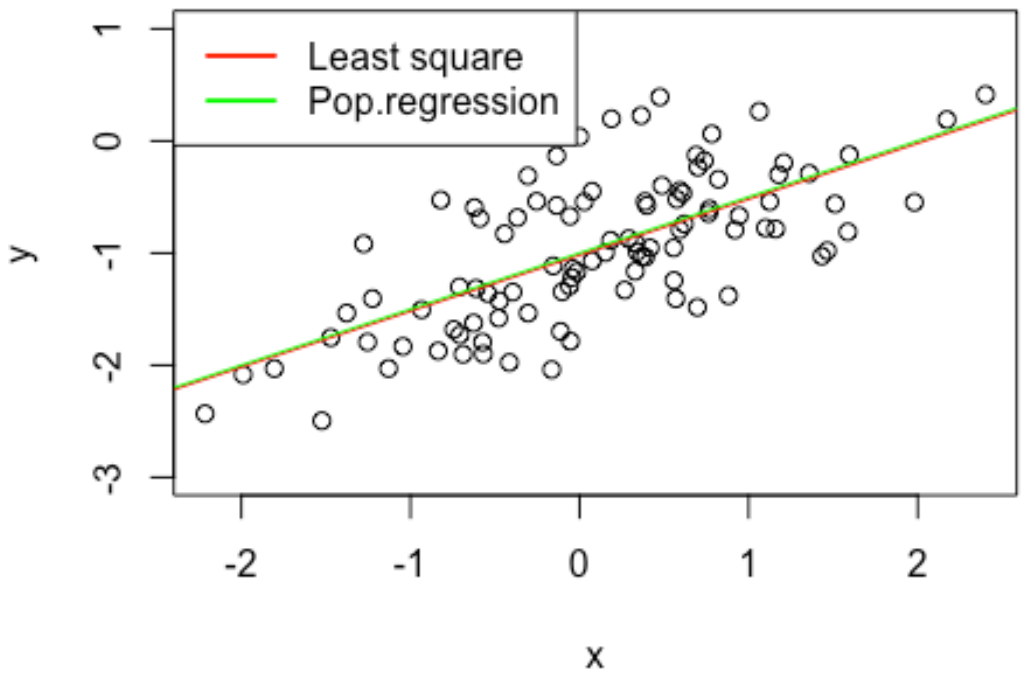
```
lm.fit <-  lm(y~x)
summary(lm.fit)
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93842 -0.30688 -0.06975  0.26970  1.17309
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01885    0.04849 -21.010  < 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
```

The values of $\hat{\beta}_0$ and $\hat{\beta}_1$ are -1.01 and 0.50 respectively which are very close to β_0 and β_1. In the other word, it can be concluded that $\hat{\beta}_0$ and $\hat{\beta}_1$ are equal to β_0 and β_1 and statistically significant.

(f)

Display the least squares line on the scatterplot obtained in (d). Draw the population regression line on the plot, in a different color. Use the legend() function to create an appropriate legend.

```
plot(y~x, ylim = c(-3,1)); abline(lm.fit, col = 'red', lwd = 1);abline(-
1,0.5,col= 'green',lwd = 1)
legend('topleft', c("Least square","Pop.regression"), col = c("red",
"green"), lty = c(1,1), lwd = 2)
```



(g)

Now fit a polynomial regression model that predicts y using x and x^2. Is there evidence that the quadratic term improves the model fit? Explain your answer.

```
poly.fit <- lm(y~x+ I(x^2))
summary(poly.fit)
##
## 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
```

There is no evidence that the quadratic term improves the model fit since the p-value of the coefficient of x^2 is greater than 0.05; therefore, the coefficient of x^2 is not statistically significant.

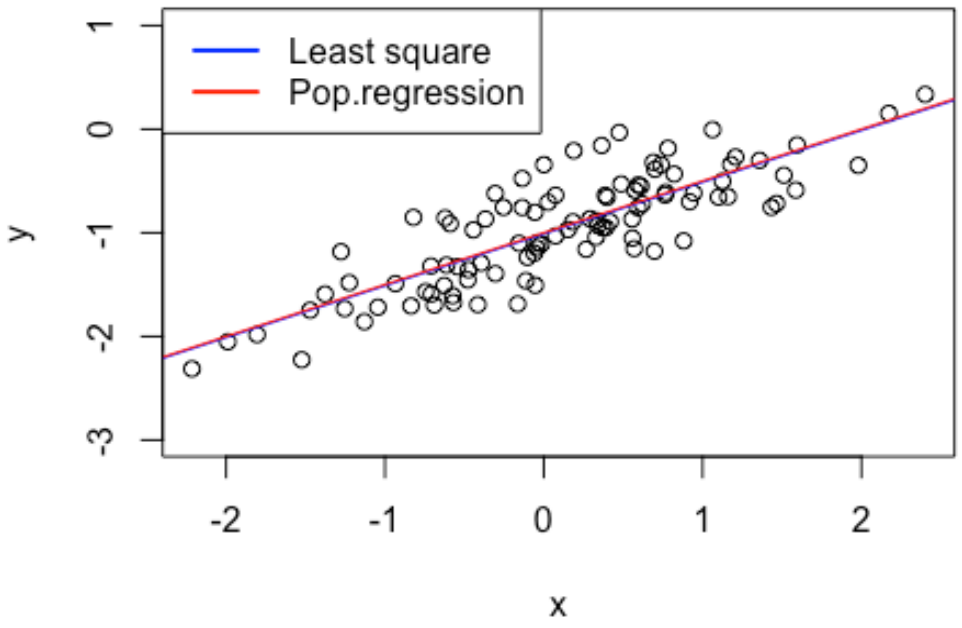
(h)

Repeat (a)-(f) after modifying the data generation process in such a way that there is less noise in the data. The model should remain the same. You can do this by decreasing the variance of the normal distribution used to generate the error term in (b). Describe your results.

```
set.seed(1)
x <- rnorm(100)
eps <- rnorm(100, mean = 0, sd = sqrt(0.1))
y <- -1+0.5*x+eps
lm.fit2 <- lm(y~x)
summary(lm.fit2)
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59351 -0.19409 -0.04411  0.17057  0.74193
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01192    0.03067  -32.99   <2e-16 ***
## x            0.49966    0.03407   14.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3044 on 98 degrees of freedom
```

```
## Multiple R-squared:  0.687,  Adjusted R-squared:  0.6838
## F-statistic: 215.1 on 1 and 98 DF,  p-value: < 2.2e-16


plot(y~x, ylim = c(-3,1)); abline(lm.fit2, col = 'blue', lwd = 1);abline(-
1,0.5,col= 'red',lwd = 1)
legend('topleft', c("Least square","Pop.regression"), col = c("blue", "red"),
lty = c(1,1), lwd = 2)
```



We reduced the variance of the normal distribution used to generate the error term to 0.10. It can be observed that the values of $\hat{\beta}_0$ and $\hat{\beta}_1$ are -1.01 and 0.5 which are very close the previous model. R-squared is equal to 0.69, so we have a higher R^2 showing that this model fits better compare to the previous model.

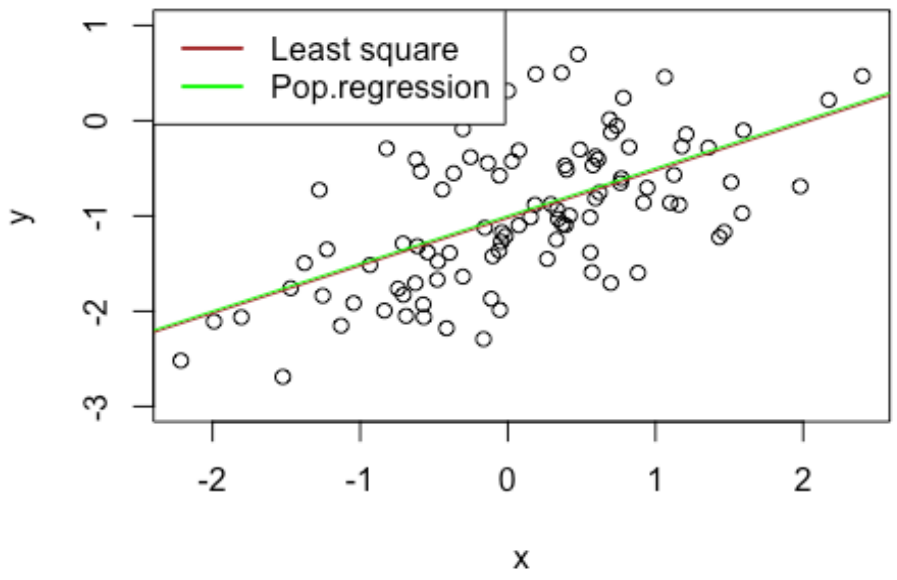
(i)

Repeat (a)-(f) after modifying the data generation process in such a way that there is more noise in the data. The model should remain the same. You can do this by decreasing the variance of the normal distribution used to generate the error term in (b). Describe your results.

```
set.seed(1)
x <- rnorm(100, mean = 0, sd = 1)
eps <- rnorm(100, mean = 0, sd = sqrt(0.40))
y <- -1+0.5*x+eps
```

```
lm.fit3 <- lm(y~x)
summary(lm.fit3)
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.18702 -0.38818 -0.08823  0.34115  1.48385
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.02384    0.06134 -16.691  < 2e-16 ***
## x            0.49933    0.06813   7.329 6.66e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6089 on 98 degrees of freedom
## Multiple R-squared:  0.354,  Adjusted R-squared:  0.3474
## F-statistic: 53.71 on 1 and 98 DF,  p-value: 6.657e-11

plot(y~x, ylim = c(-3,1)); abline(lm.fit3, col = 'brown', lwd = 1);abline(-
1,0.5,col= 'green',lwd = 1)
legend('topleft', c("Least square","Pop.regression"), col = c("brown",
"green"), lty = c(1,1), lwd = 2)
```



We increased the variance of the normal distribution used to generate the error term to 0.40. It can be observed that the values of $\hat{\beta}_0$ and $\hat{\beta}_1$ are -1.02 and 0.5 which are very close the

original model. R-squared is equal to 0.35, which is lower than the other models. Therefore, this model does not fit the data points quite well.

(j)

What are the confidence intervals for β_0 and β_1 based on the original data set, the noisier data set, and the less noisy data set? Comment on your results.

The confidence intervals for β_0 and β_1 based on the original data set

```
confint(lm.fit, level = 0.95)
##                   2.5 %     97.5 %
## (Intercept) -1.1150804 -0.9226122
## x            0.3925794  0.6063602
```

The confidence intervals for β_0 and β_1 based on the less noisy data set

```
confint(lm.fit2, level = 0.95)
##                   2.5 %     97.5 %
## (Intercept) -1.0727832 -0.9510557
## x            0.4320613  0.5672681
```

The confidence intervals for β_0 and β_1 based on the noisier data set

```
confint(lm.fit3, level = 0.95)
##                   2.5 %     97.5 %
## (Intercept) -1.1455664 -0.9021114
## x            0.3641225  0.6345362
```

From the results, the more noise in the data causes the confidence intervals of β_0 and β_1 widen.

4) Perform the following commands in R:

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

(a)

The last line corresponds to creating a linear model in which y is a function of x1 and x2. Write out the form of the linear model. What are the regression coefficients?

The form of linear model is as follow:

$$\mathbf{Y = 2 + 2X_1 + 0.3X_2 + \epsilon},$$

where $\epsilon \sim N(0,1)$.
The regression coefficients are 2, 2, and 0.3 respectively.

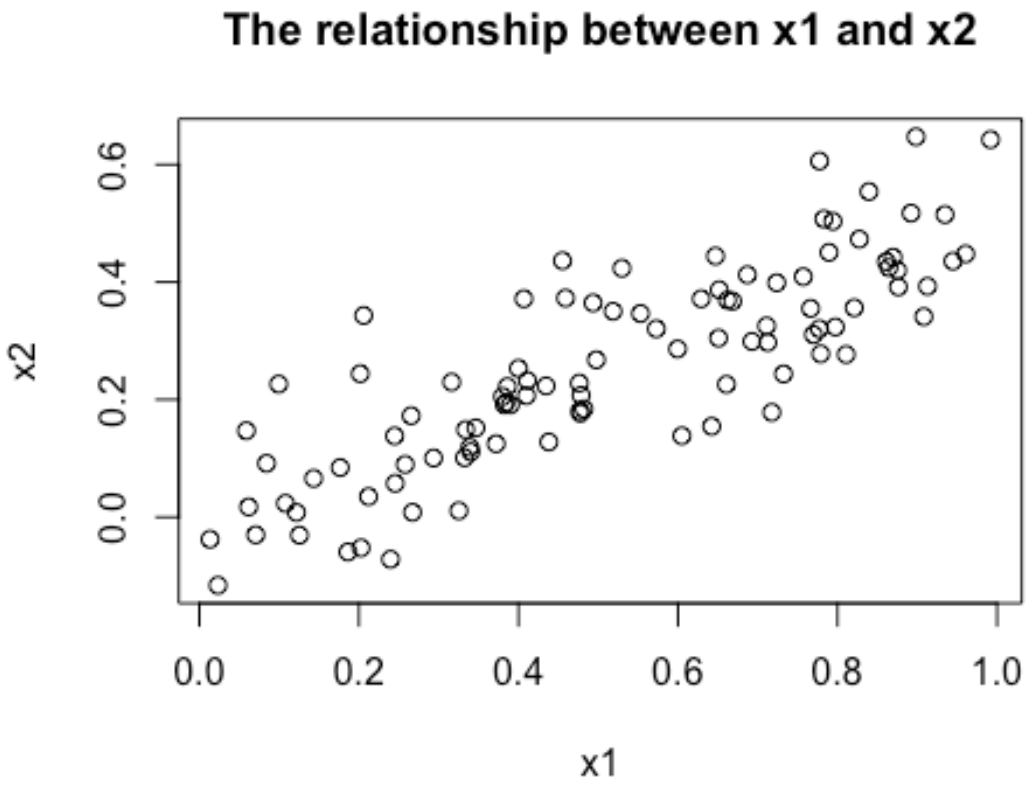
(b)

What is the correlation between x1 and x2? Create a scatterplot displaying the relationship between the variables.

```
cor <- cor(x1,x2)
sprintf("%s is %0.3f", "The correlation between x1 and x2",cor)

## [1] "The correlation between x1 and x2 is 0.835"


plot(x1,x2,main = "The relationship between x1 and x2")
```



(c)

Using this data, fit a least squares regression to predict y using x1 and x2. Describe the results obtained. What are $\hat{\beta}_0$, $\hat{\beta}_1$, and β_2? How do these relate to the true β_0, β_1, and β_2? Can you reject the null hypothesis $H_0: \beta_1 = 0$? How about the null hypothesis $H_0: \beta_2 = 0$?

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

$\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ are 2.1305,1.4396, and 1.0097 respectively. As β_0, β_1, and β_2 are equal to 2, 2, and 0.3 respectively, only $\hat{\beta}_0$ is close to the true β_0. We can reject the null hypothesis $H_0: \beta_1 = 0$ since the p-value of this coefficient is less than 0.05. However, we fail to reject the null hypothesis of $H_0: \beta_2 = 0$ because the p-value is greater than 0.05.

(d)

Now fit a least squares regression to predict y using only x1. Comment on your results. Can you reject the null hypothesis $H_0: \beta_1 = 0$

```
lm.fit2<-lm(y~x1)
summary(lm.fit2)
##
## 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 coefficient of x_1 is equal to 1.9759 which is different than the original model. We can reject the null hypothesis with a significant level of 0.05 because the p-value is less than 0.05. We would say that the coefficient of x_1 from this model is highly significant.

(e)

Now fit a least squares regression to predict y using only x2. Comment on your results. Can you reject the null hypothesis $H_0: \beta_2 = 0$

```
lm.fit3<-lm(y~x2)
summary(lm.fit3)
##
## 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 coefficient of x_2 is equal to 2.8996 which is very different than the original model. We can reject the null hypothesis with a significant level of 0.05 because the p-value is less than 0.05. We would say that the coefficient of x_2 from this model is statistically significant.

(f)

Do the results obtained in (c)-(e) contradict each other? Explain your answer.

No, the results obtained in (c) - (e) do not contradict each other. Recall from part (b) that we found that the correlation between x_1 and x_2 is equal to 0.835 which shows that x_1 and x_2 are highly correlated. This suggests collinearity between these two variables. The presence of collinearity can pose problem in the regression context, since it can be difficult to separate out the individual effects of collinear variables on the response. Collinearity increases the error of the regression estimation. Therefore, there is no contradiction as we can fail to reject the null hypothesis of a model including x_1 and x_2.

(g)

Now suppose we obtain one additional observation, which was unfortunately mismeasured.

```
x1 <- c(x1, 0.1)
x2 <- c(x2, 0.8)
y <- c(y, 6)
```

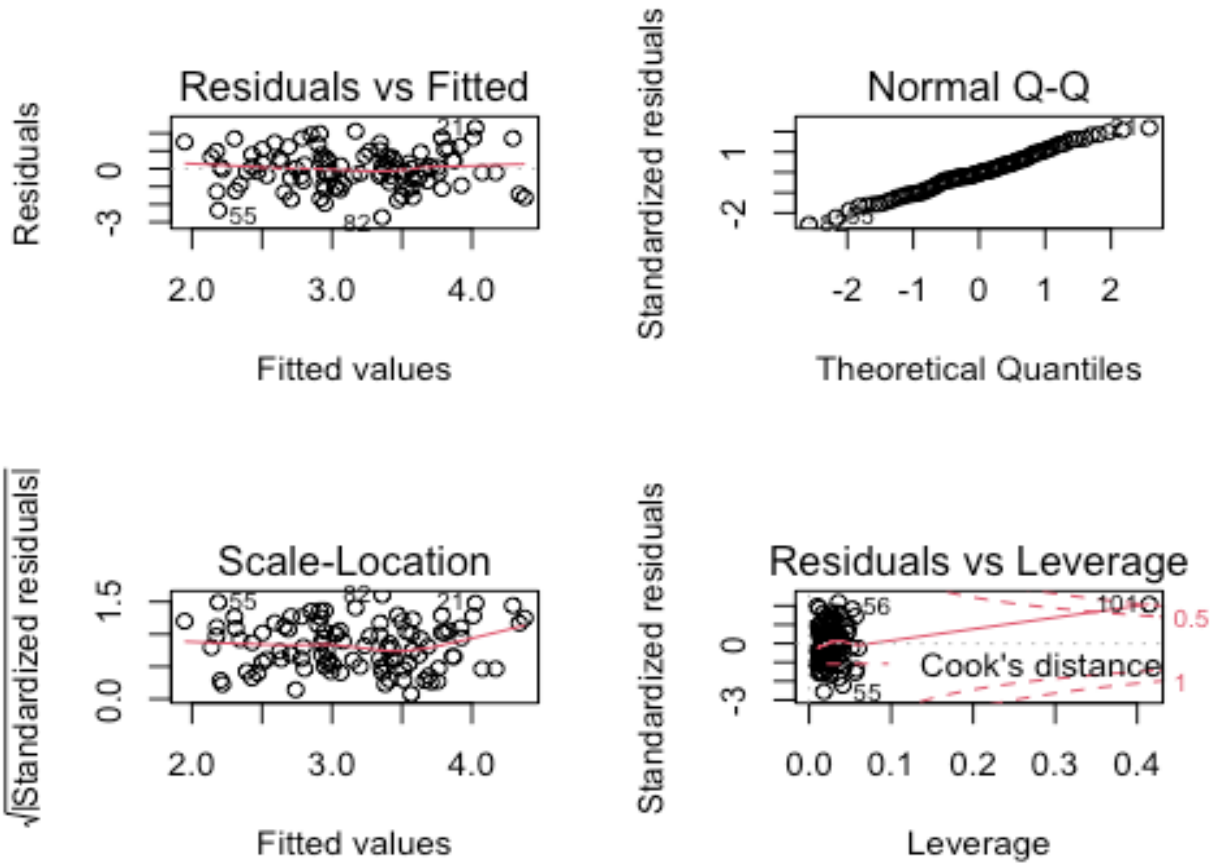
Re-fit the linear models from (c) to (e) using this new data. What effect does this new observation have on each of the models? In each model, is this observation an outlier? A high-leverage point? Explain your answers.

Fitting y using x_1 *and* x_2

```
lm.fit4<- lm(y~x1+x2)
lm.fit5 <- lm(y~x1)
lm.fit6 <- lm(y~x2)
summary(lm.fit4)
##
## 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
```

To predict y using x_1 and x_2, we can observe that x_1 is not statistically significant, but x_2 is statistically significant as the p-value is greater than 0.05.

```
par(mfrow = c(2,2))
plot(lm.fit4)
```



The observation 101 is an outlier in this model as it does not label in the residual plot even though it may not be an outlier in Q-Q plot, but it has the high-leverage point in the plot of residual and leverage.

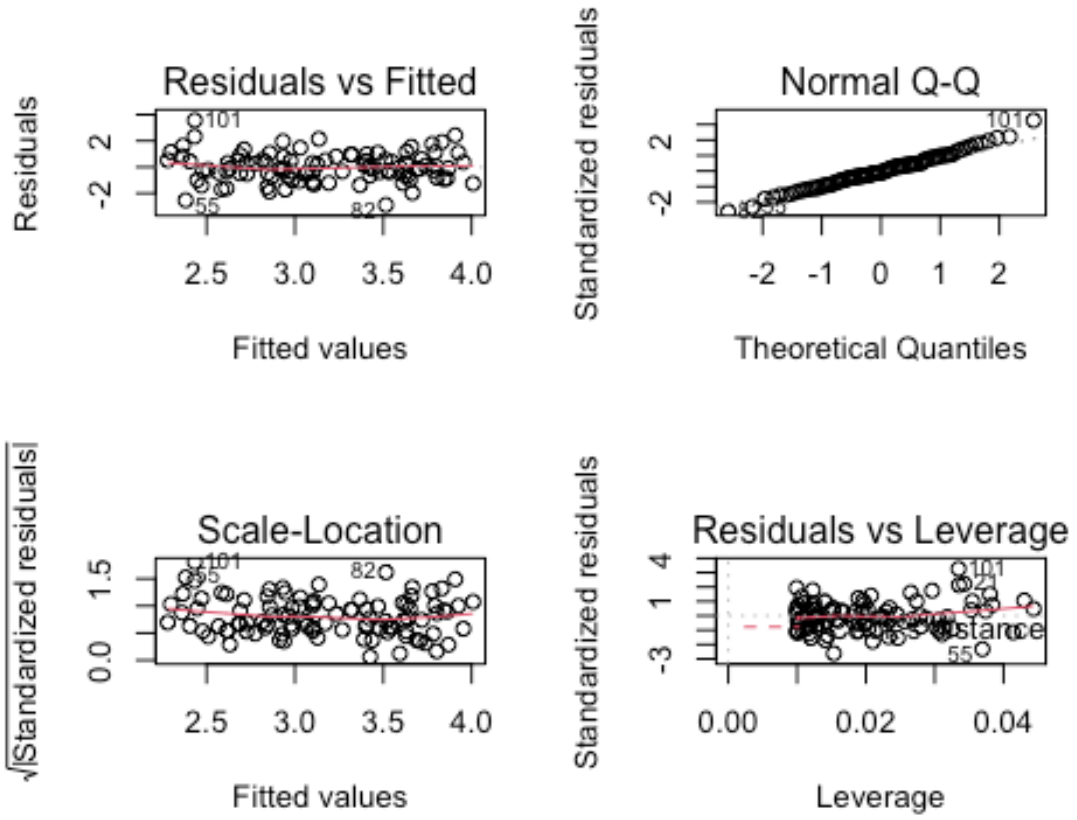
Fitting y using x_1

```
summary(lm.fit5)
##
## 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
```

To predict y using x_1, we can observe that x_1 is statistically significant as the p-value is greater than 0.05.

```
par(mfrow = c(2,2))
plot(lm.fit5)
```



The observation 101 is not an outlier in this model as it does label in the residual plot, but it is an outlier in Q-Q plot. However, it is not the high-leverage point in the plot of residual and leverage.

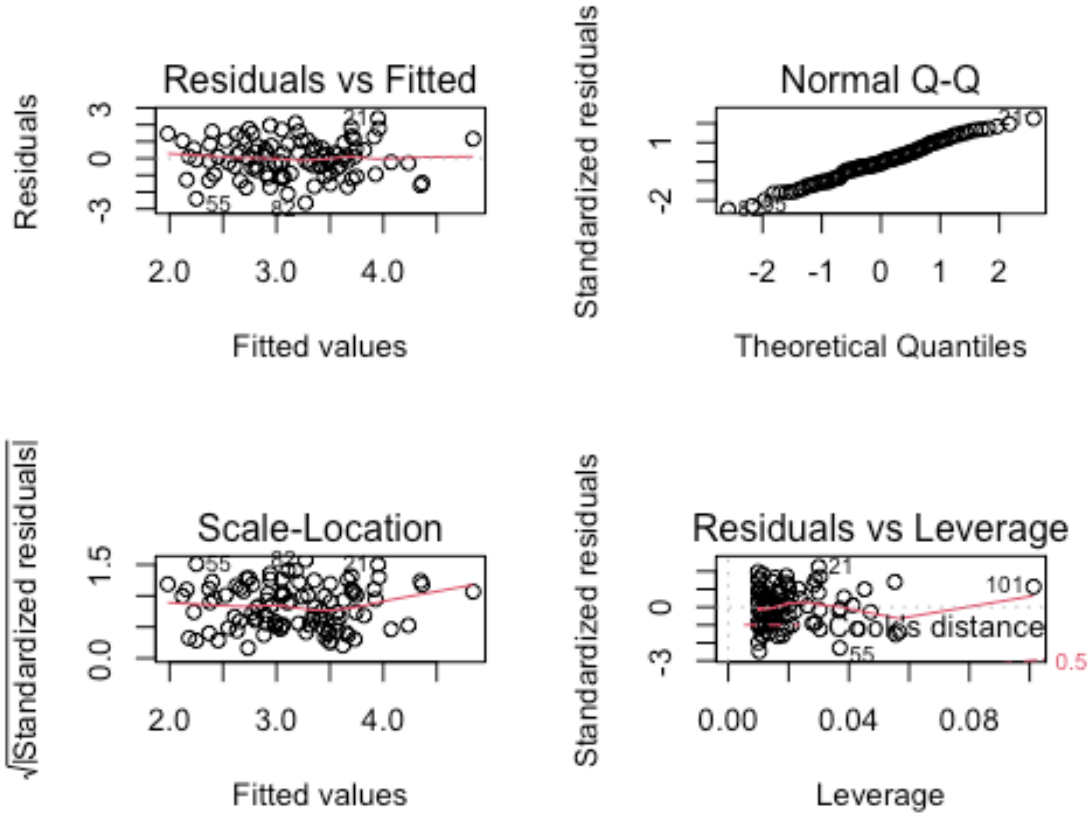
Fitting y using x_2

```
summary(lm.fit6)
##
## 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
```

To predict y using x_2, we can observe that x_2 is statistically significant as the p-value is greater than 0.05.

```
par(mfrow = c(2,2))
plot(lm.fit6)
```



The observation 101 is an outlier in this model as it does not label in the residual plot even though it may not be an outlier in Q-Q plot, but it has the high-leverage point in the plot of residual and leverage.