Homework 3

Anishka Chauhan

2022-07-18

```
if (!require(pacman)) {install.packages(pacman)}
```

```
## Loading required package: pacman
```

```
pacman::p_load(ggplot2,readr,tidyr,dplyr)
```

1a)

```
x = rnorm(50)
range_x = range(x)
a = range_x[1]
b = range_x[2]
paste0("[a,b] = ", a, ",", " ", b)
```

```
## [1] "[a,b] = -2.67700571627587, 2.98186298109166"
```

```
paste0("Mean of values is ", mean(x))
```

```
## [1] "Mean of values is 0.163248368307548"
```

```
paste0("Median of values is ", median(x))
```

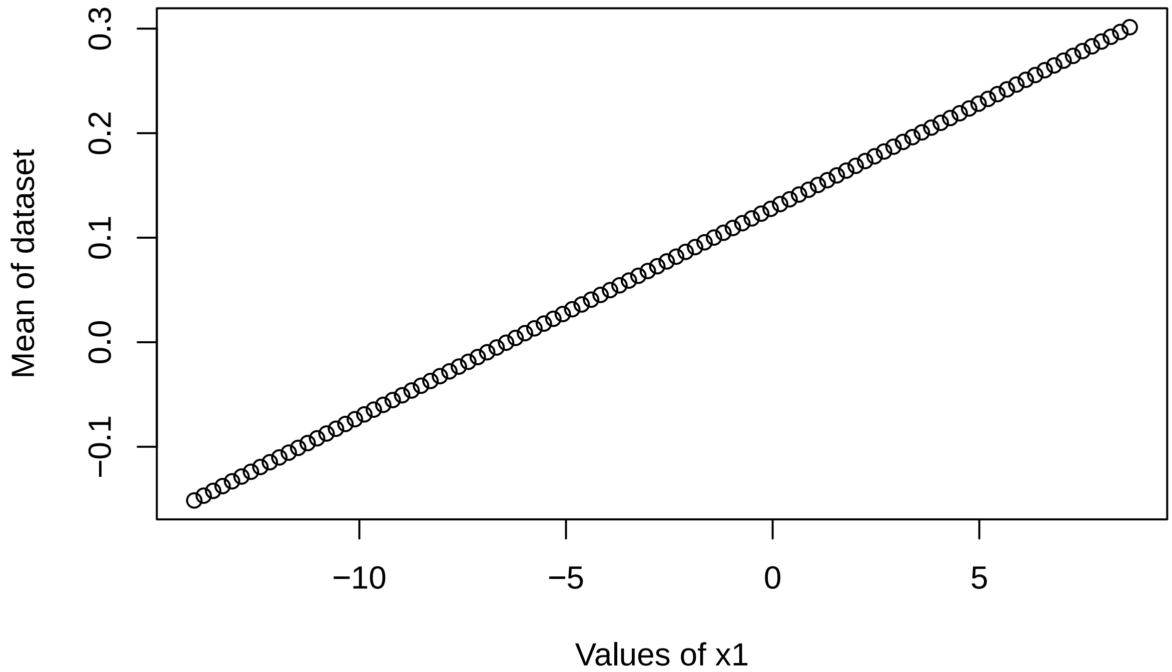
```
## [1] "Median of values is 0.0943558396662256"
```

1b)

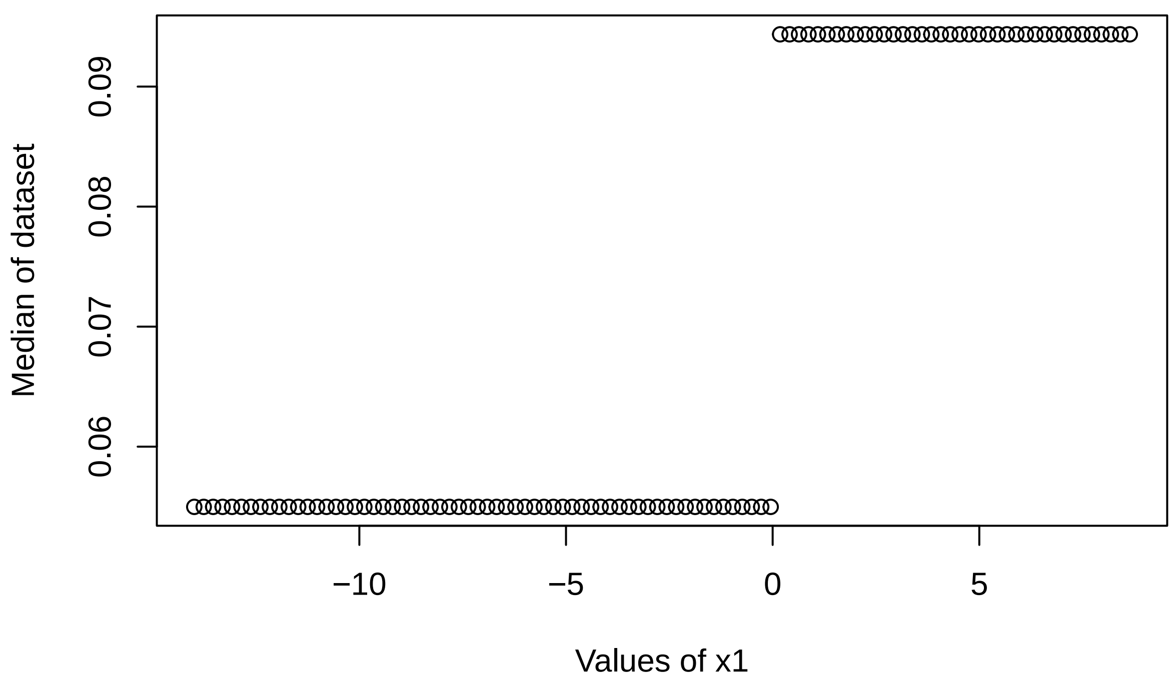
```
y1 = a- (2*(b - a))
y100 = a + (2*(b - a))
y = seq(y1, y100, length.out = 100)
z = seq(100)

loo.mean <- function(z,y,x) {
  x = replace(x, 1, y[z])
  return(mean(x))
}

loo.median <- function (z, y, x) {
  x = replace(x, 1, y[z])
  return(median(x))
}
mns = sapply(z, loo.mean, y, x)
meds = sapply(z, loo.median, y,x)
plot(y, mns, xlab = "Values of x1", ylab = "Mean of dataset")
```




```
plot(y, meds, xlab = "Values of x1", ylab = "Median of dataset")
```



1c) Changing 1 value in a dataset has a greater impact on the mean than it does on the median. The graph showing the mean vs different values of x1 shows that the mean changed at a constant rate, indicating any changes in x1 change the mean. The graph showing median vs different values of x1 shows that the median stays constant as x1 is less than 0, then when x1 is greater than 0, jumps to a positive value and stays constant at that value as x1 stays positive. This indicates that median is only drastically affected by changes in sign of a value; otherwise it stays constant.

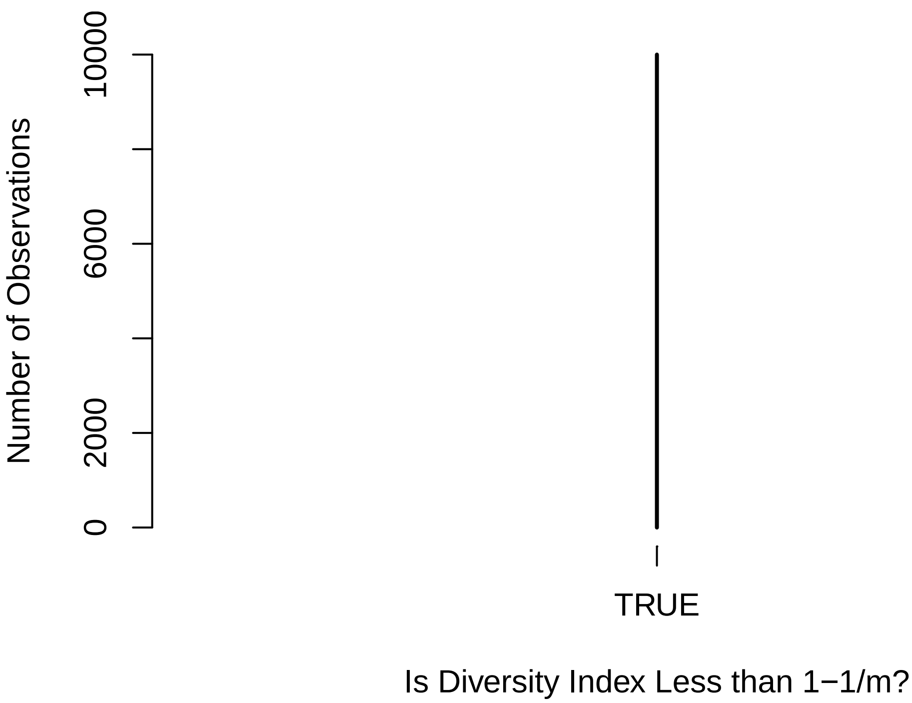
2)

```r
result = c()
B = 10000
for(i in 1:B) {
  rel_freq = sample(1:100, 50, replace = TRUE)

  rel_freq = rel_freq/sum(rel_freq)

  d_index = 1 - sum(rel_freq^2)
  result = c(result, d_index <= 1-(1/length(rel_freq)))
}
result = table(result)

plot(result, xlab = "Is Diversity Index Less than 1-1/m?", ylab = "Number of Observations")
```



Simulating the diversity index of B random datasets containing 50 relative frequency values and evaluating whether the index is less than 1-1/length of the dataset shows that all B simulations returned a true value, indicating the diversity index of a dataset pi. . . pm is always less than 1-1/m.

3a) For the training set I would expect the cubic regression to have a lower RSS because it has more flexibility. For the test set, I would expect the linear RSS to be lower than the cubic RSS because the cubic RSS would most likely be overfitted to the training set and therefore have higher error with a test set.

3b) The cubic regression will have more flexibiity than the linear regression, so the cubic RSS will have lower training RSS than the linear regression RSS. There is not enough information to know if the cubic regression test RSS will be higher or lower than the linear regression test RSS because we don't know how far the true relationship is from linear, so if its closer to linear the linear test RSS would be lower and if its closer to cubic the cubic test RSS would be lower instead.

```r
pacman::p_load(ISLR)
data(Auto)
```

4a)

(c)

$$\hat{y} = x_i \hat{\beta}$$

$$\hat{y} = \frac{x_i \sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$$

$$\hat{y} = \frac{\sum_{i=1}^{n} x_i^2 y_i}{\sum_{i=1}^{n} x_i^2} \qquad \hat{y} = \sum_{i=1}^{n} \left(\frac{x_i^2}{\sum_{i=1}^{n} x_i^2} \right) y_i$$

$$a_{i,j} = \left(\frac{x_i^2}{\sum_{i=1}^{n} x_i^2} \right)$$

(d)

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

$$\hat{y} = (\bar{y} - \hat{\beta}_1 \bar{x}) + \hat{\beta}_1 x \qquad \text{substitute } \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\hat{y} = \bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 (\bar{x}) \qquad \text{plug in } \bar{x}\text{, solve for } \hat{y}$$

$$\hat{y} = \bar{y}$$

Therefor model passes through $(\bar{x}, \bar{y})$

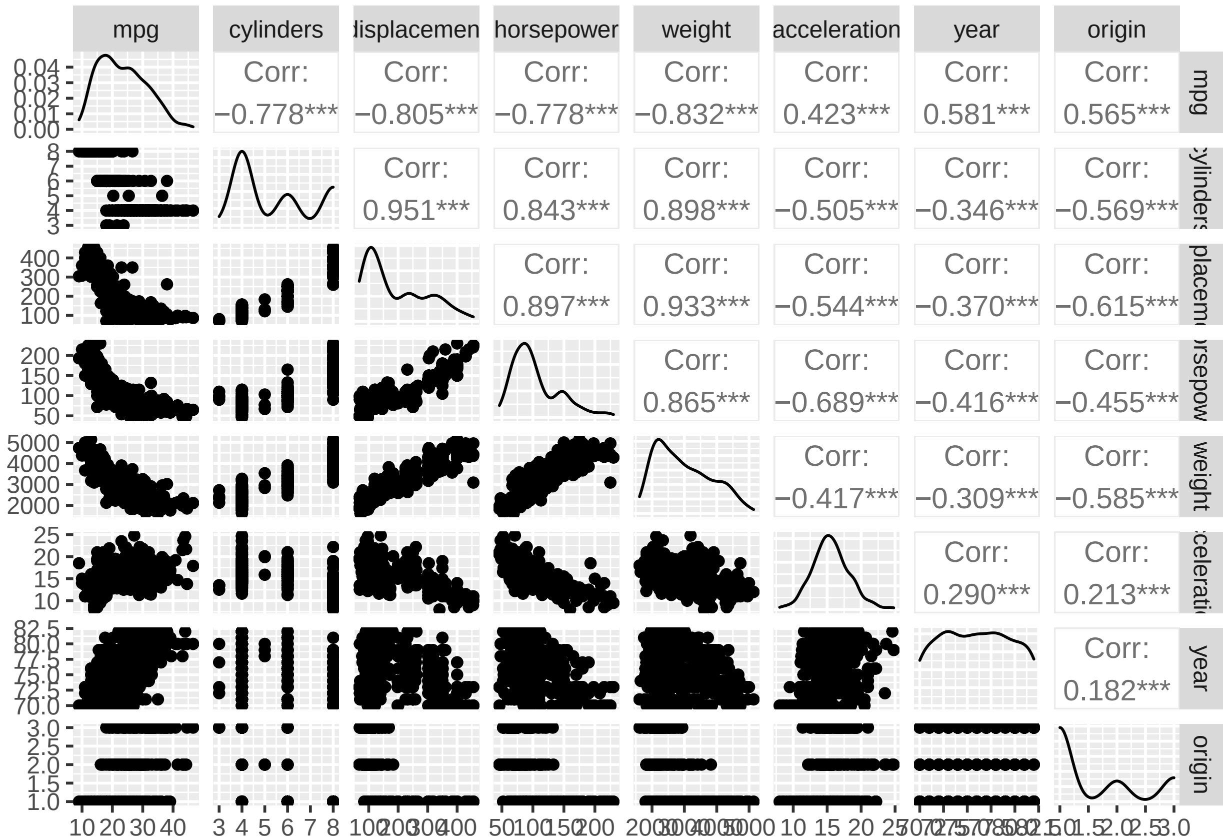
The model doesn't hold if there is no intercept as the intercept is needed to cancel out the $\hat{\beta}_1 \bar{x}$ term and set the $\bar{y}$ term equal to $\hat{y}$

```r
pacman::p_load(dplyr)
if (!require(GGally)) {install.packages(GGally,type='source')}
```

```
## Loading required package: GGally
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
X <- select_if(Auto, is.numeric)    # drop other qualitative variables
ggpairs(X)
```



```r
correlations = cor(X)
correlations
```

```
##                     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
```

4b)

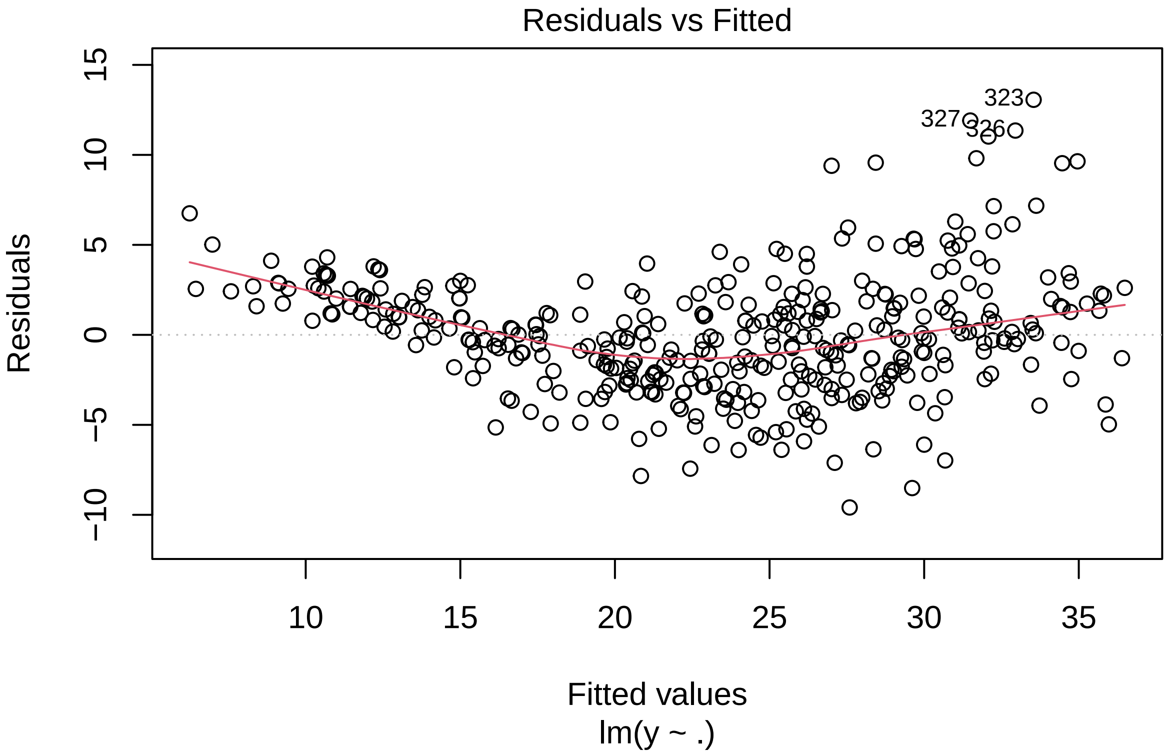
```
y = Auto$mpg
X <- dplyr::select(Auto, -mpg, -name)
mpg_reg = lm(y ~., X)
summary(mpg_reg)
```

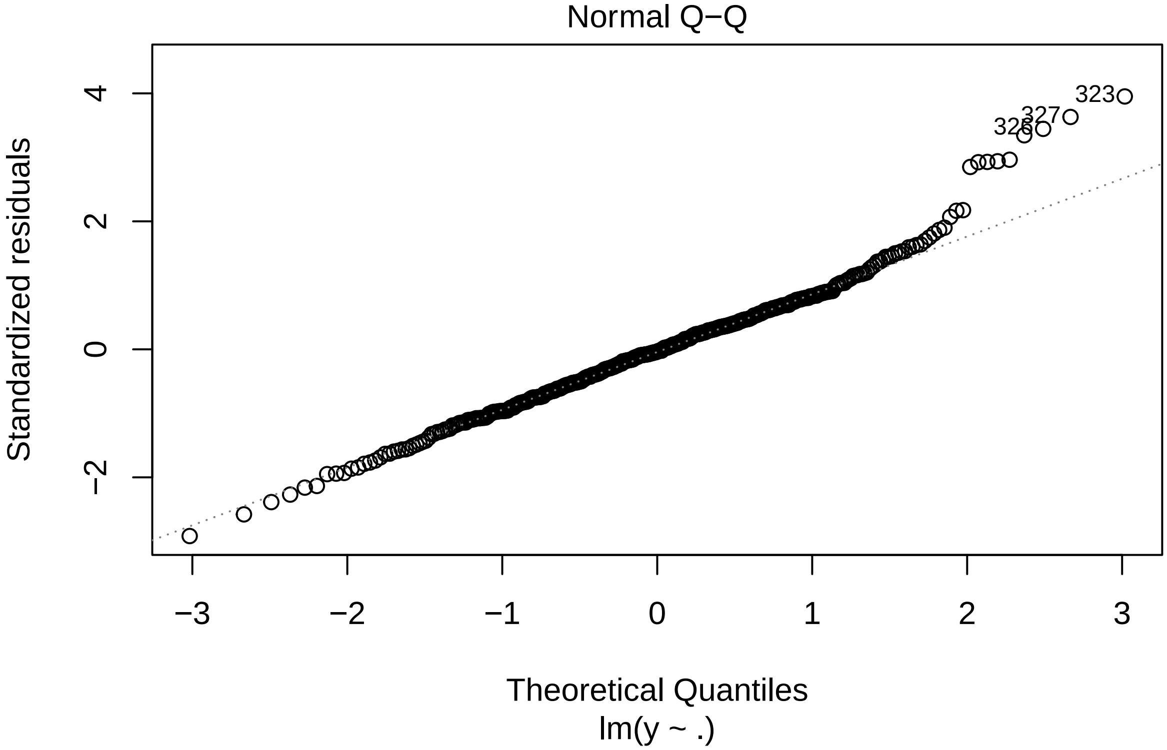
```
##
## Call:
## lm(formula = y ~ ., data = X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders   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

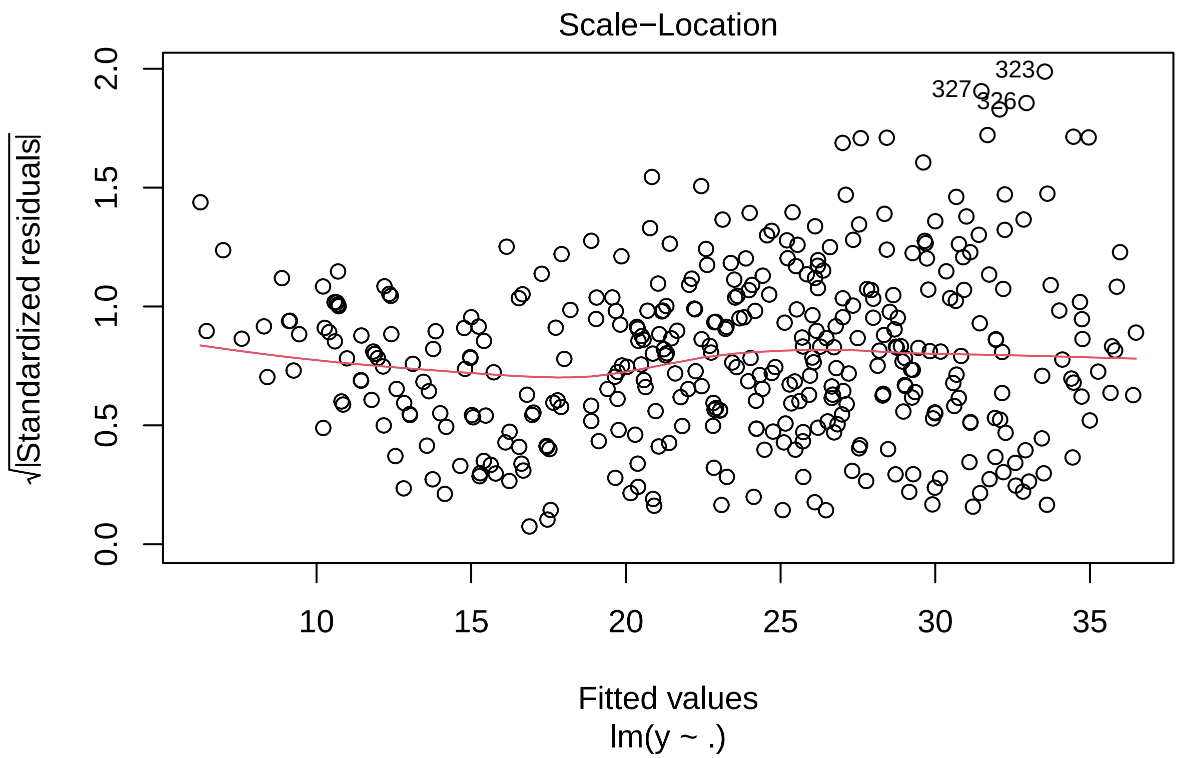
There seems to be a relationship between the predictors and the response. The year, origin, and cylinders variables have statistically significant relationship. The year coefficient suggests that it is the variable that has the strongest/most plausible correlation with mpg. For the weight coefficient, the sign indicates that cylinders and mpg are negatively corellated, and the magnitude indicates for every increase of 1 mpg, the weight of the car goes down by 0.006574 lbs.

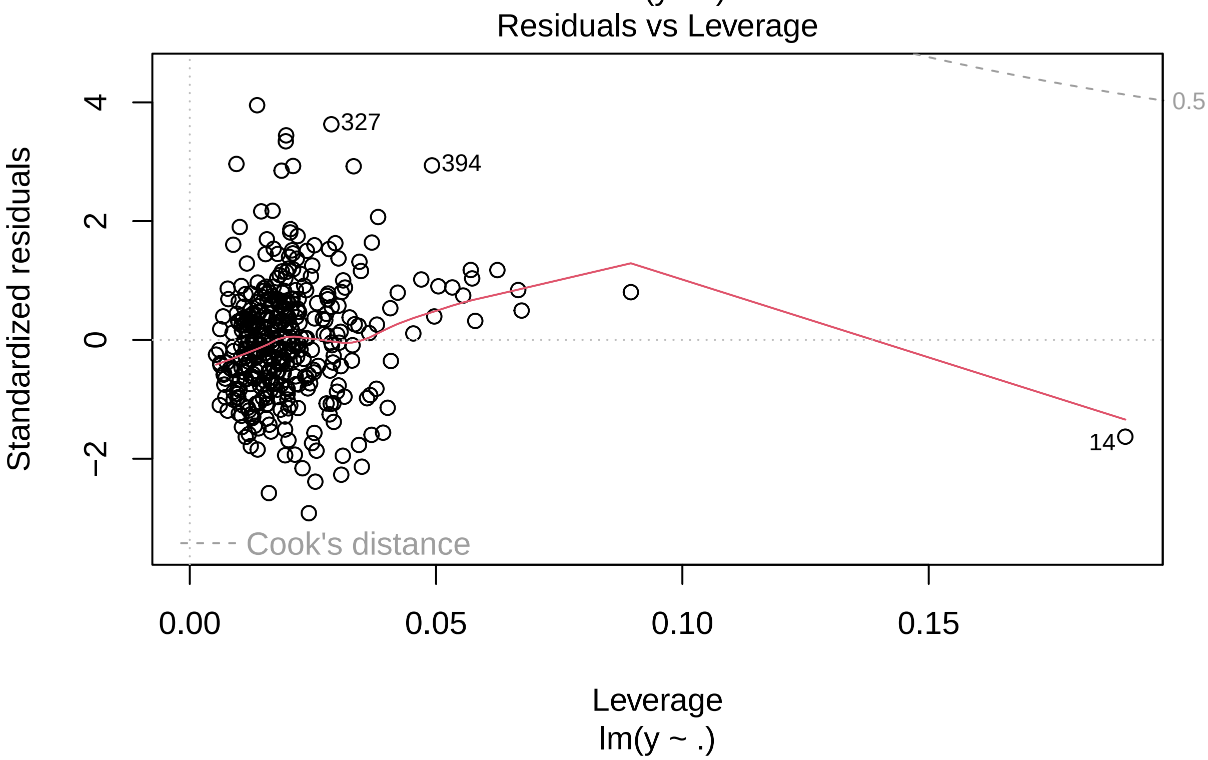
4c)

```
plot(mpg_reg)
```


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


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

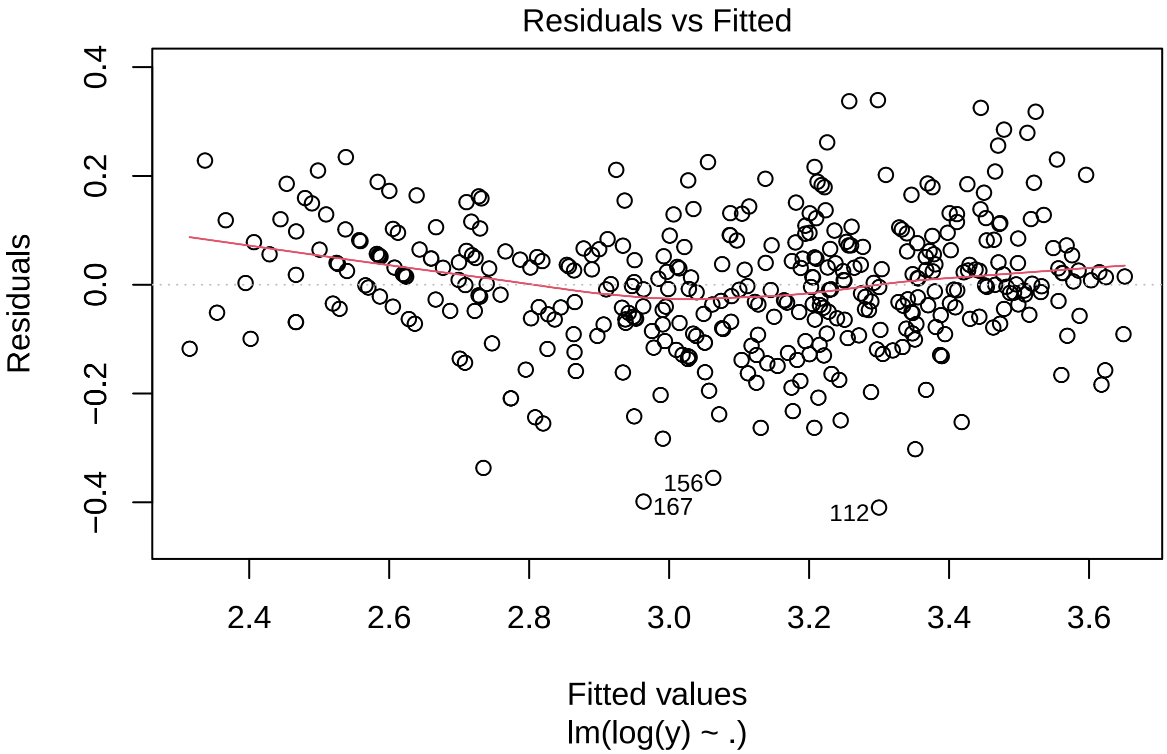


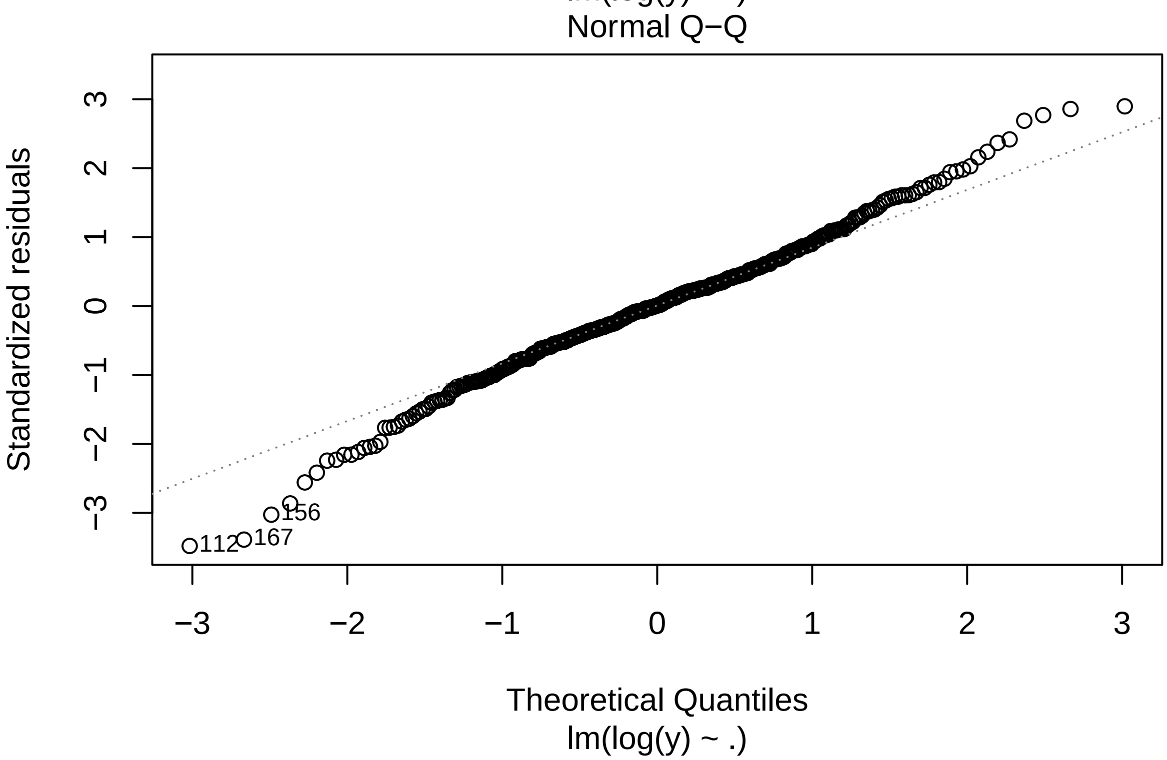


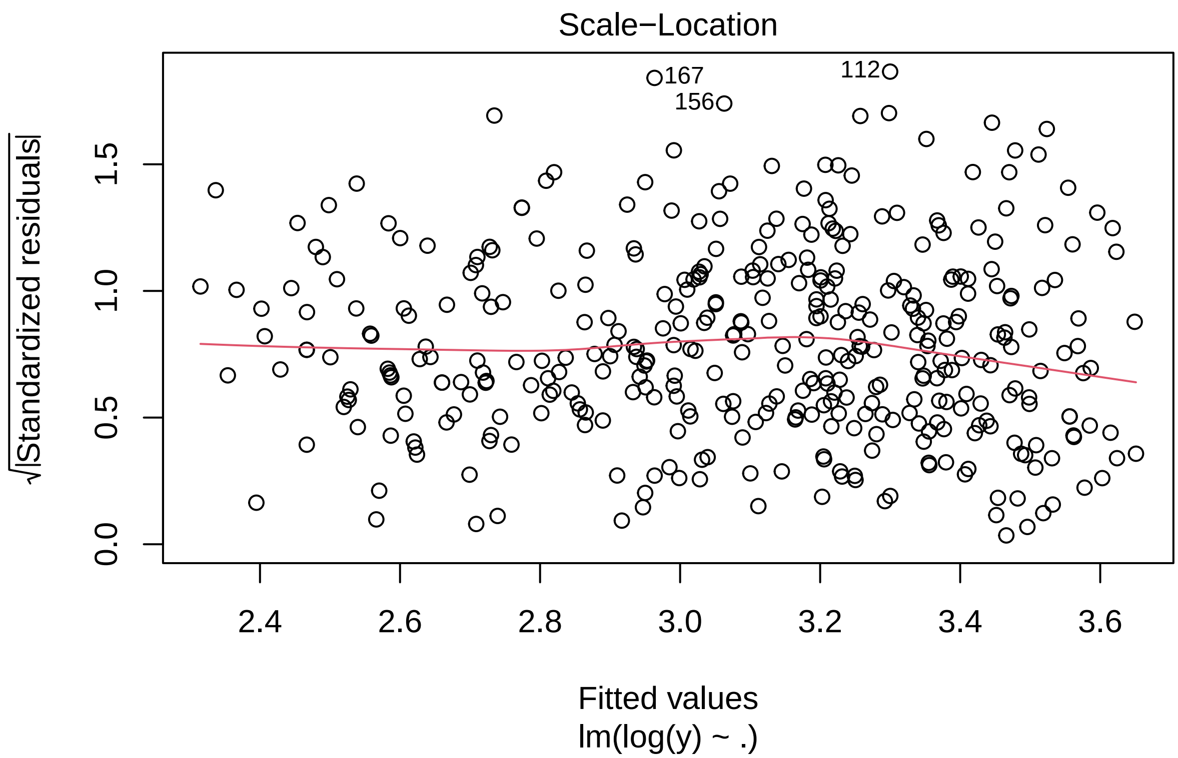
The residuals vs leverage plot indicates that very few points have very high leverage, with point 14 having the highest, making it an outlier. The residual plots do suggest the presense of outliers. The residuals vs fitted shows the red line as being somewhat curved, indicating there are some minor problems with the fit. The residuals vs fitted value graph is also in a somewhat cone shape, indicating that the variance is not constant and therefore there is another problem with the regression model.

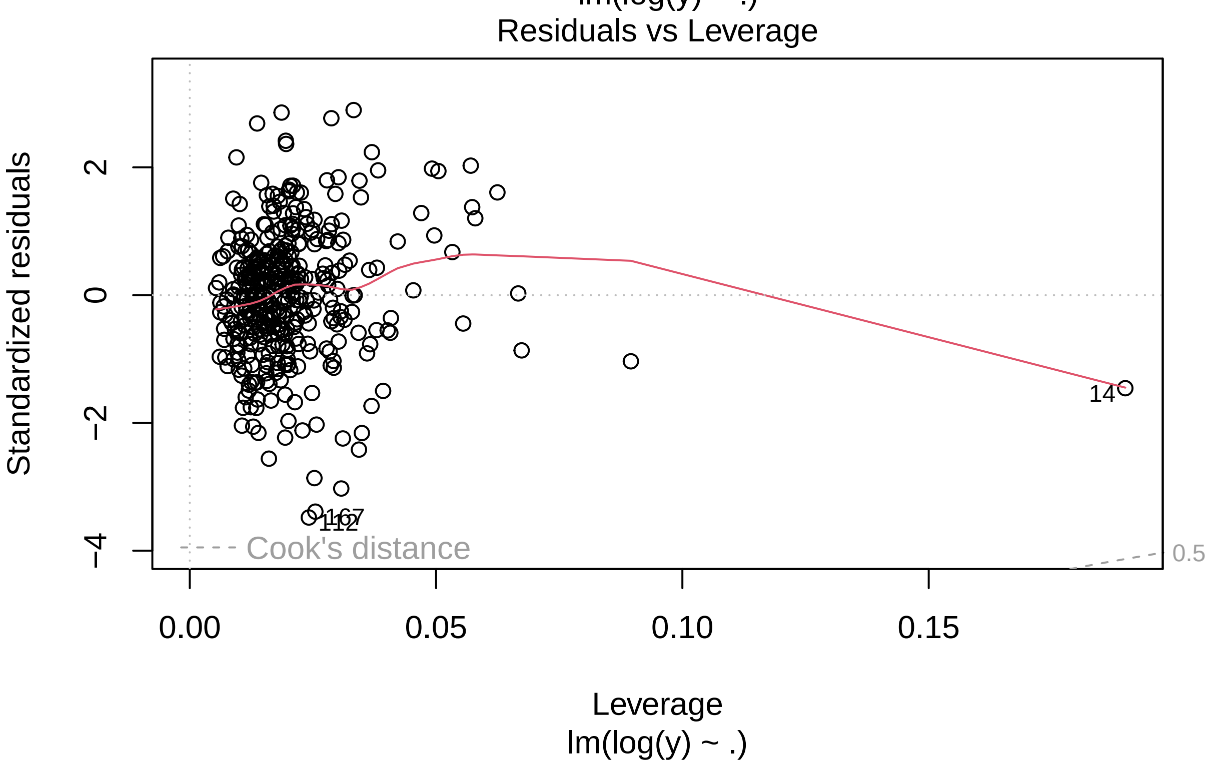
4d)

```
mpg_reg_log = lm(log(y) ~., X)
plot(mpg_reg_log)
```

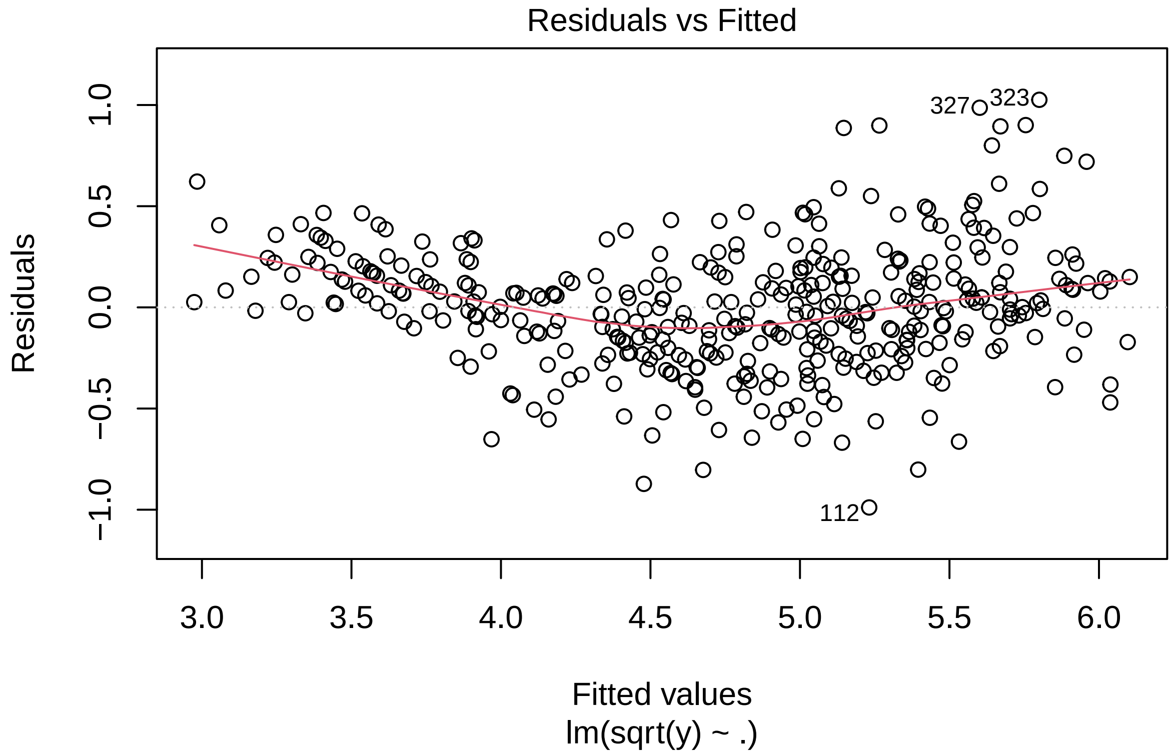


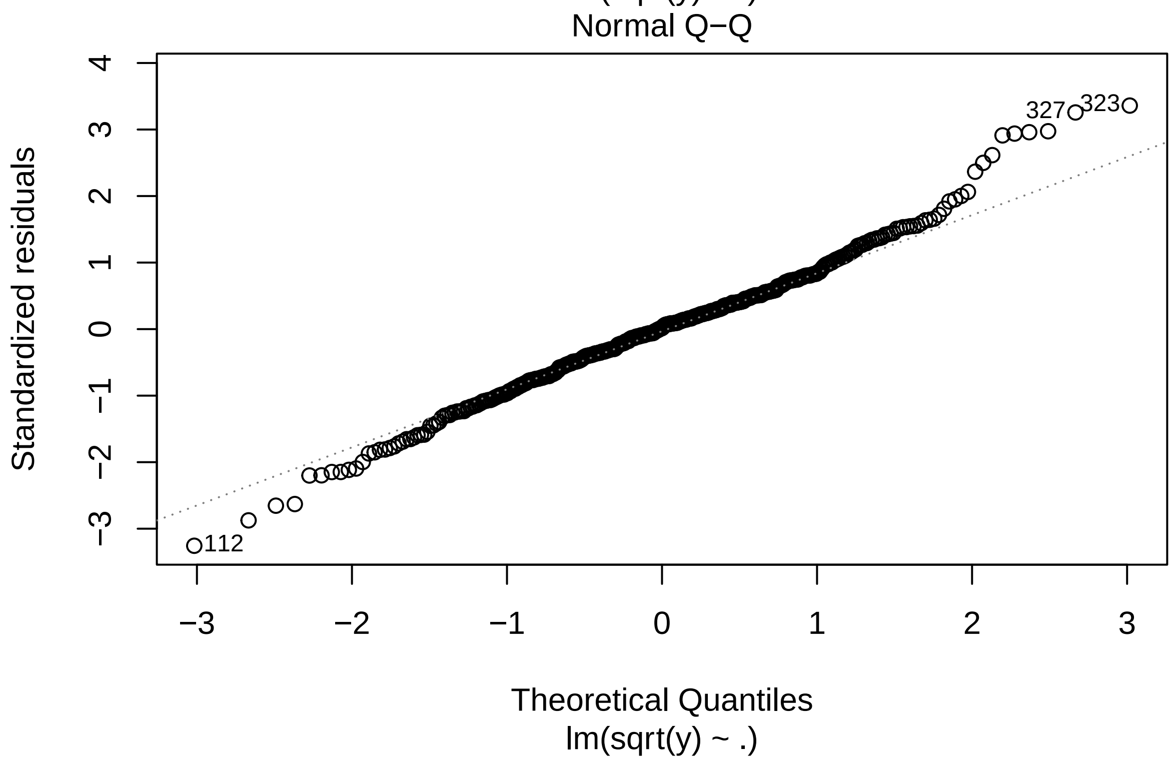


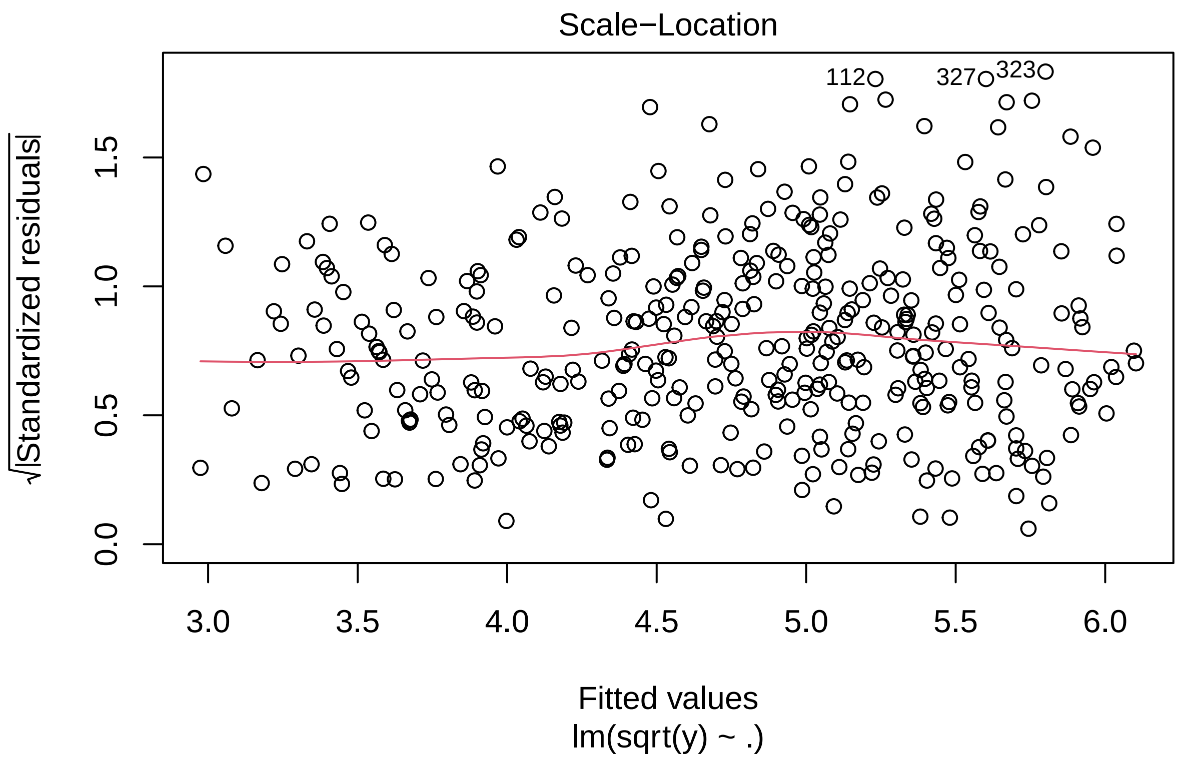


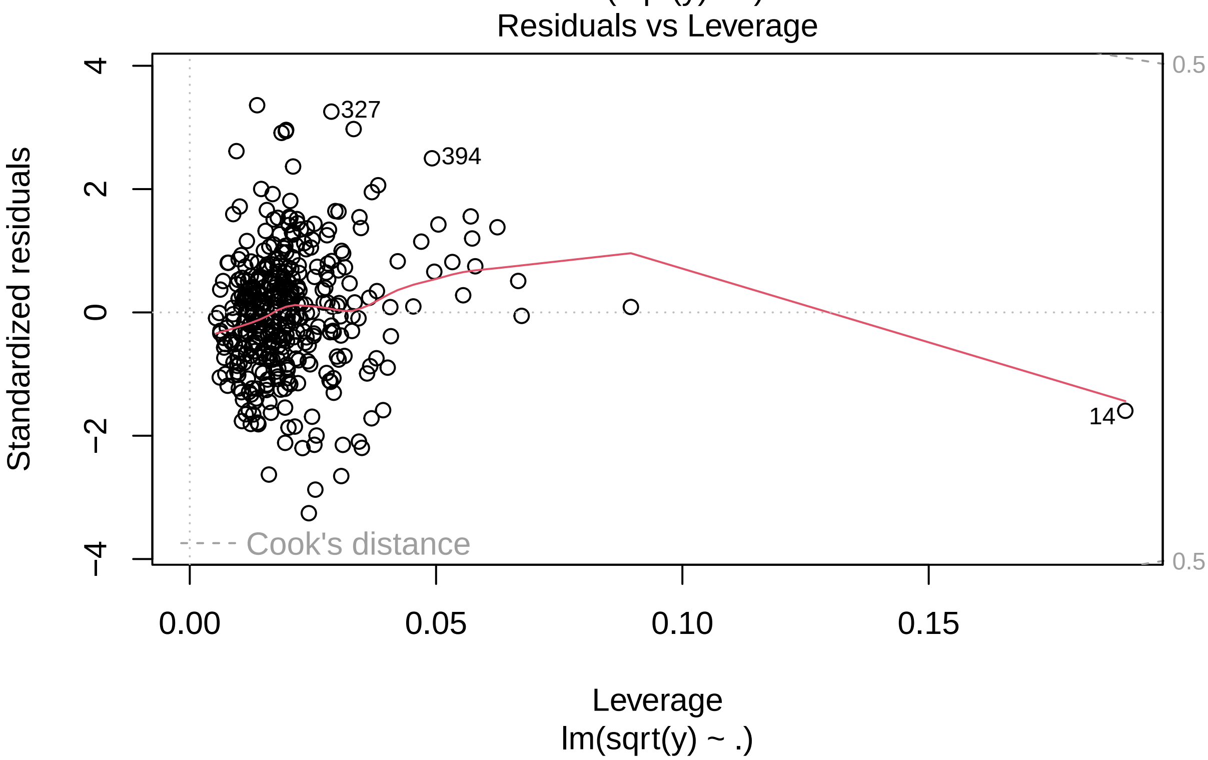



```
mpg_reg_sqrt = lm(sqrt(y) ~., X)
plot(mpg_reg_sqrt)
```

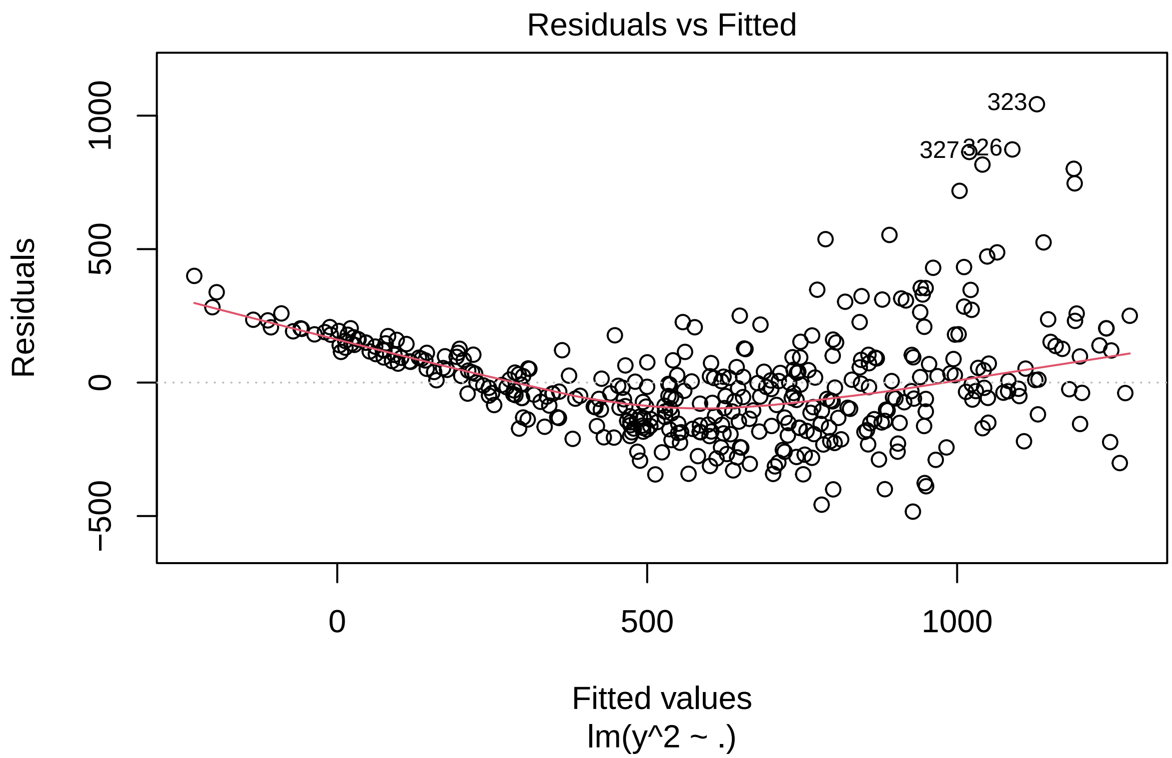

Residuals vs Fitted
Residuals
Fitted values
lm(sqrt(y) ~ .)
327
323
112

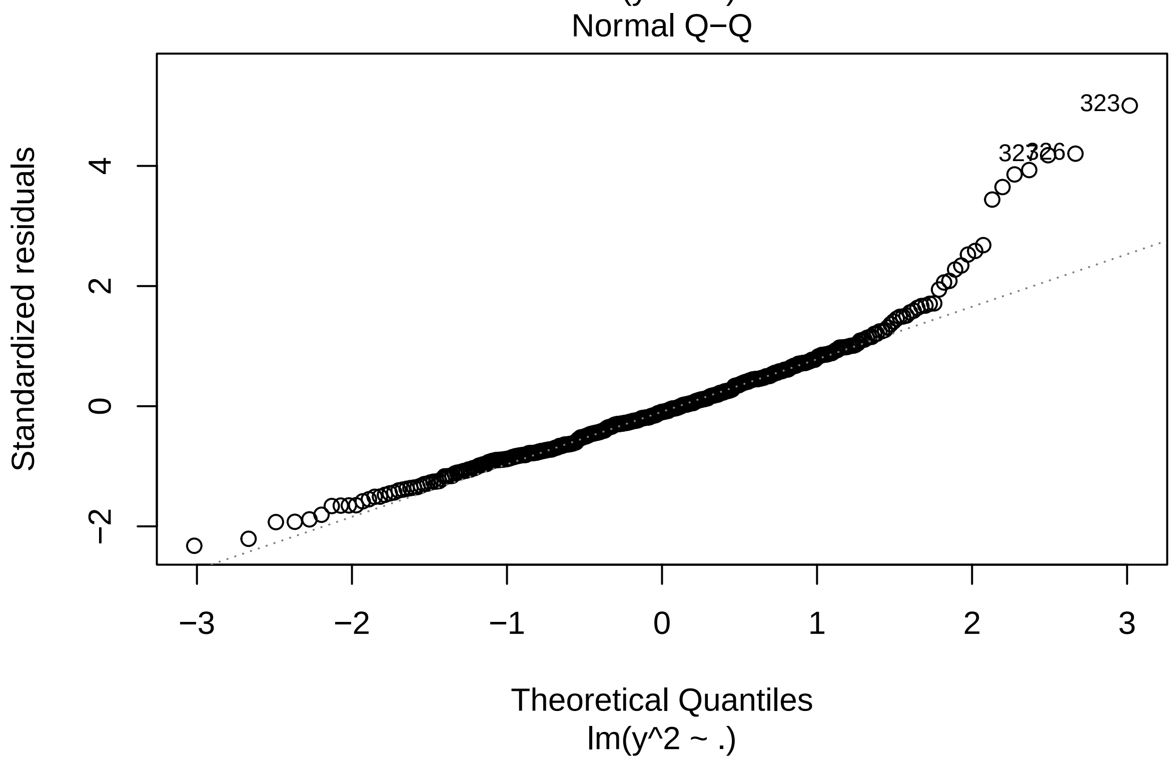

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(sqrt(y) ~ .)
327
323
112

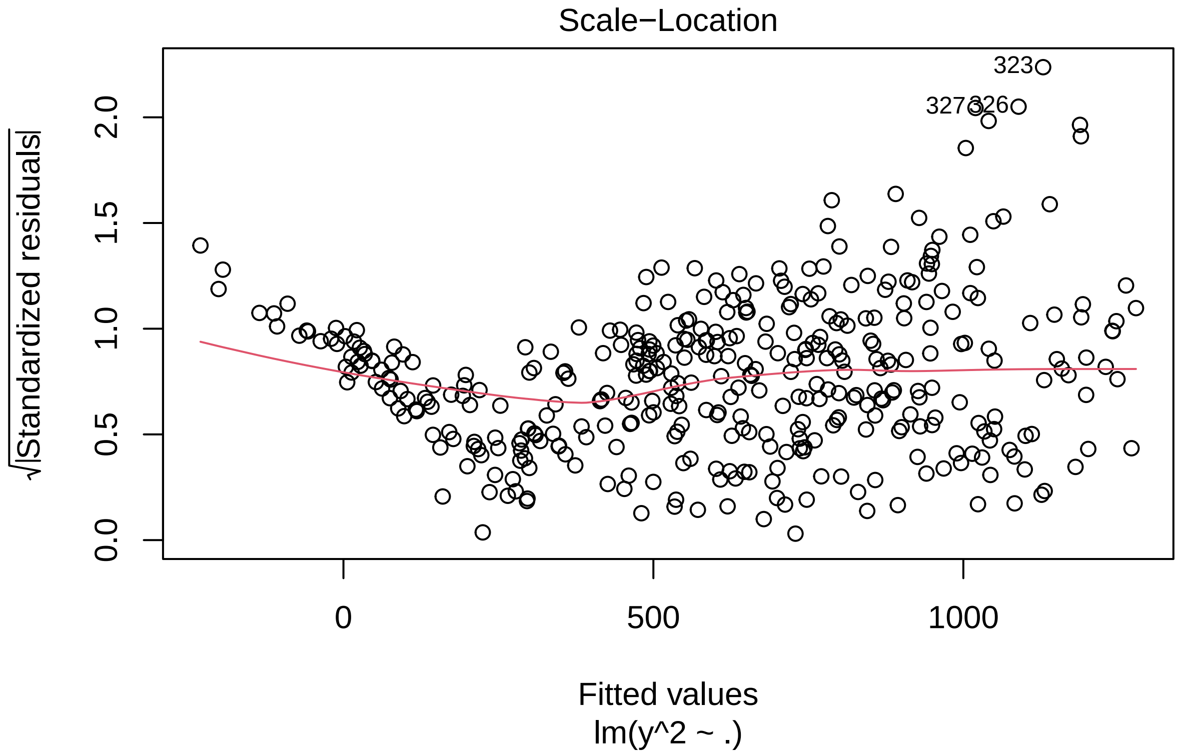


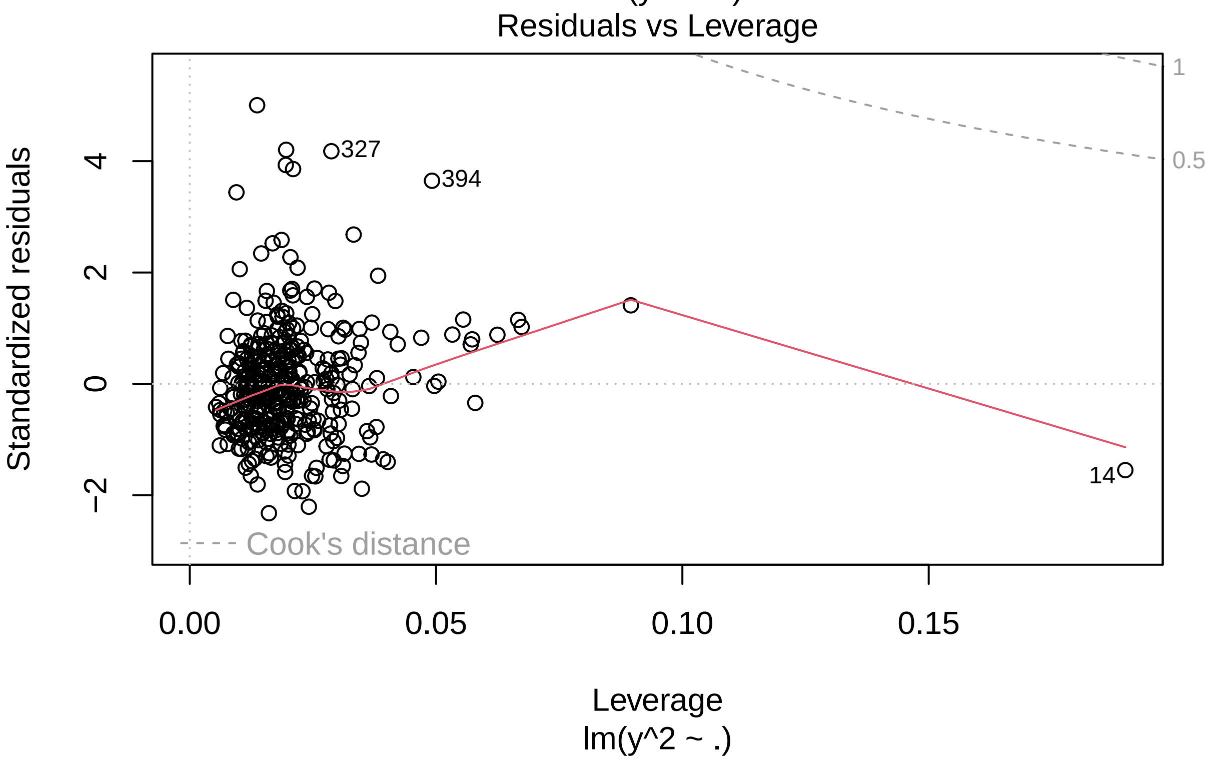



```
mpg_reg_sqr = lm(y^2 ~., X)
plot(mpg_reg_sqr)
```


Residuals vs Fitted
Residuals
1000
500
0
−500
323
327
326
0
500
1000
Fitted values
lm(y^2 ~ .)


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





The logarithmic transformation of y seems to improve the fit of the regression. The scale-location plot for the logarithmic transformation is the flattest out of all the plots, indicating its variance stays constant for the most part. The logarithmic transformation also has the least high leverage points.