

Statistical Learning - Homework 2

James Hahn

Chapter 3 - Exercise 3

- a) The regression line for Males is $\hat{y} = 50 + 20 \times GPA + 0.07 \times IQ + 0.01 \times GPA \times IQ$. Meanwhile, for Females, the line is $\hat{y} = 85 + 10 \times GPA + 0.07 \times IQ + 0.01 \times GPA \times IQ$. We know i) is false since a GPA of 0 means females make more than males. ii) is false since a GPA of 3.5 means the females make the same as males. iv) is false since the higher the GPA, the closer the starting salary of males is to the starting salary of females, eventually leading males to surpass females if the GPA is high enough. Therefore, iii) is the correct option.
- b) $y = 85 + 10 \times GPA + 0.07 \times IQ + 0.01 \times GPA \times IQ = 85 + 40 + 7.7 + 4.4 = 137.1$. Therefore, the starting salary is \$137,100.
- c) False. To verify the lack of effectiveness on the model, we need to test the $H_o : \beta_4 = 0$, and then look at the p-value associated with the t or F statistic to draw a conclusion on whether or not to reject the null hypothesis.

Chapter 3 - Exercise 4

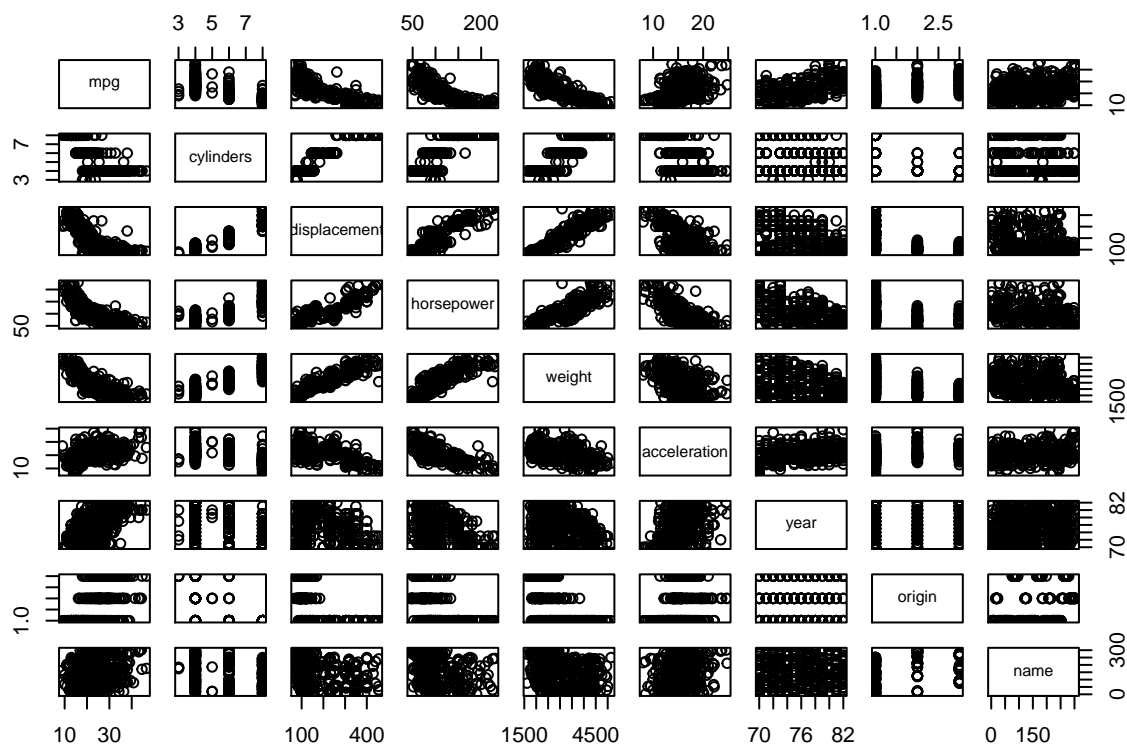
- a) Since the relationship between X and Y is linear, we expect the linear training RSS for linear regression will be lower than the cubic regression.
- b) The polynomial regression fitting will be prone to overfitting, so the cubic regression test RSS is expected to be higher than the linear regression test RSS. Therefore, the linear regression RSS is expected to be lower than cubic regression RSS.
- c) Polynomial regression has more flexibility, so since the relationship is not linear, we expect the cubic regression training RSS to be lower than the linear regression training RSS.
- d) The only information we are given is the relationship is not linear. This can mean the relationship is very close to being linear or very far from being linear. As such, we do not know enough about the test dataset to assess the situation and cannot draw conclusions on whether linear or cubic regression test RSS is lower.

Chapter 3 - Exercise 9

a)

```
library(ISLR)
```

```
pairs(Auto)
```



b)

```
names(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"
## [5] "weight"      "acceleration" "year"         "origin"
## [9] "name"
```

```
cor(Auto[1:8])
```

```
##           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) i.

```
fit2 <- lm(mpg ~ . - name, data = Auto)
summary(fit2)
```

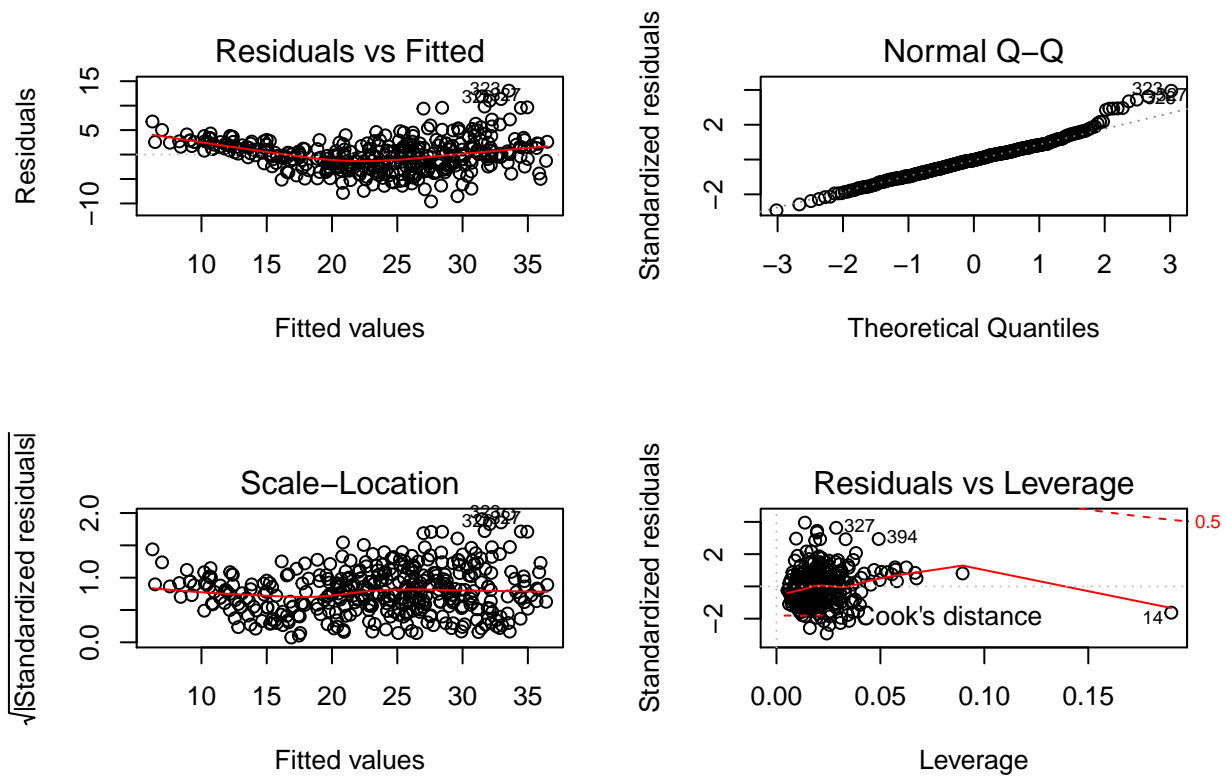
```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.218435   4.644294  -3.707  0.00024 ***
##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

ii. From part i), we can see cylinders, horsepower, and acceleration all have p-values in the range of

iii. It suggests that the every year, mpg of cars increase by 0.750773, which means cars become more fu

d)

```
par(mfrow = c(2, 2))
plot(fit2)
```



The plot of residuals vs. fitted values shows some non-linearity in the data. The plot of standardized residuals vs. leverage shows presence of a few outliers above +2 and below -2. There is one high leverage point at point 14.

e)

```
fit3 <- lm(mpg ~ cylinders * displacement+displacement * weight, data = Auto[, 1:8])
summary(fit3)
```

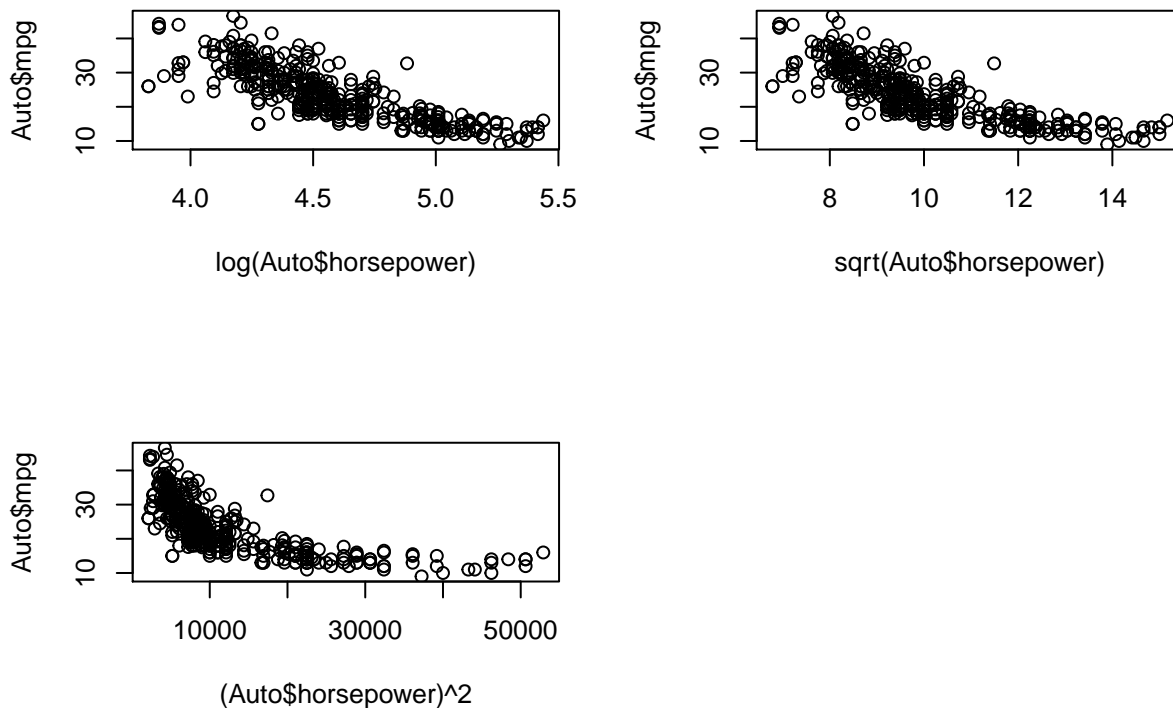
```
##
## Call:
## lm(formula = mpg ~ cylinders * displacement + displacement *
##     weight, data = Auto[, 1:8])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.2934  -2.5184  -0.3476   1.8399  17.7723
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.262e+01  2.237e+00  23.519  < 2e-16 ***
## cylinders       7.606e-01  7.669e-01   0.992   0.322
## displacement  -7.351e-02  1.669e-02  -4.403 1.38e-05 ***
## weight        -9.888e-03  1.329e-03  -7.438 6.69e-13 ***
## cylinders:displacement -2.986e-03  3.426e-03  -0.872   0.384
## displacement:weight  2.128e-05  5.002e-06   4.254 2.64e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.103 on 386 degrees of freedom
## Multiple R-squared:  0.7272, Adjusted R-squared:  0.7237
## F-statistic: 205.8 on 5 and 386 DF,  p-value: < 2.2e-16
```

From above, after observing the p-values, we can see the interaction between cylinders and displacement is not statistically significant, while the interaction between displacement and weight is statistically significant.

f)

```
par(mfrow = c(2, 2))
plot(log(Auto$horsepower), Auto$mpg)
plot(sqrt(Auto$horsepower), Auto$mpg)
plot((Auto$horsepower)^2, Auto$mpg)
```



In this sample, we observed horsepower in the three requested transformations. We can clearly see the log transformation provides the most linear-looking scatterplot.

Chapter 3 - Exercise 10

a)

```
data(Carseats)
fit3 <- lm(Sales ~ Price + Urban + US, data = Carseats)
summary(fit3)
```

```
##
```

```
## 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)

The US variable indicates a US store sells 1200.573 more units than non-US stores with all other predictors fixed. The Urban variable indicates an urban store sells 21.916 less units than rural stores with all other predictors fixed. The Price variable indicates for every dollar increase, 54.459 fewer units are sold with all other predictors fixed.

c)

The model is written as $Sales = 13.043469 - 0.054459 \times Price - 0.021916 \times Urban + 1.200573 \times US + \epsilon$. It is important to note in this equation, Urban = 1 if the store is in an urban location, and 0 otherwise. Also, US = 1 if the store is in the US and 0 otherwise.

d)

We can reject the null for Price and US since their p-values are < 0.025 .

e)

```
fit4 <- lm(Sales ~ Price + US, data = Carseats)
summary(fit4)
```

```
##
## 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)

The correlation, or R^2 value, for the smaller model is only slightly higher than the original model. It indicates 23.93% of the variability is explained by this model. They perform pretty much just as well as each other. They are very mediocre.

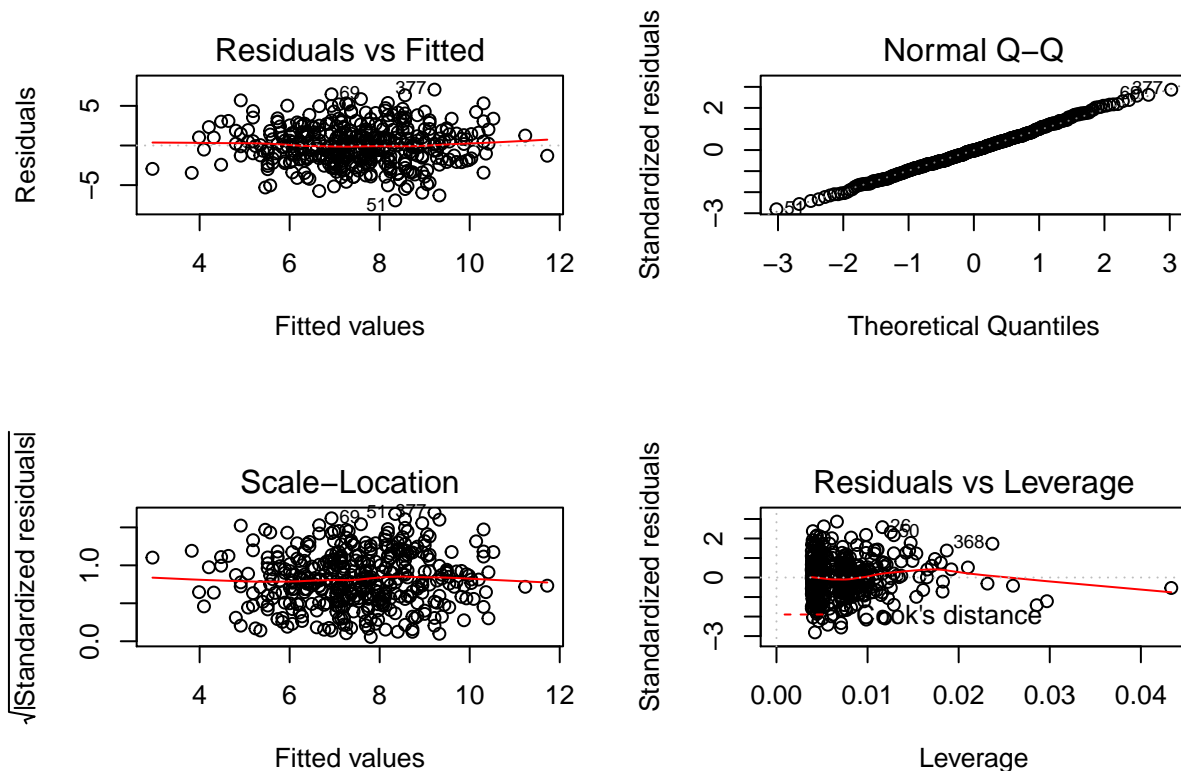
g)

```
confint(fit4)
```

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

h)

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



The standardized results vs. leverage plot shows a few outliers with standardized residuals lower than -2 and higher than +2. Also, there are many high leverage points that exceed 0.01.

Chapter 3 - Exercise 13

a)

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

b)

```
eps <- rnorm(100, sd = sqrt(0.25))
```

c)

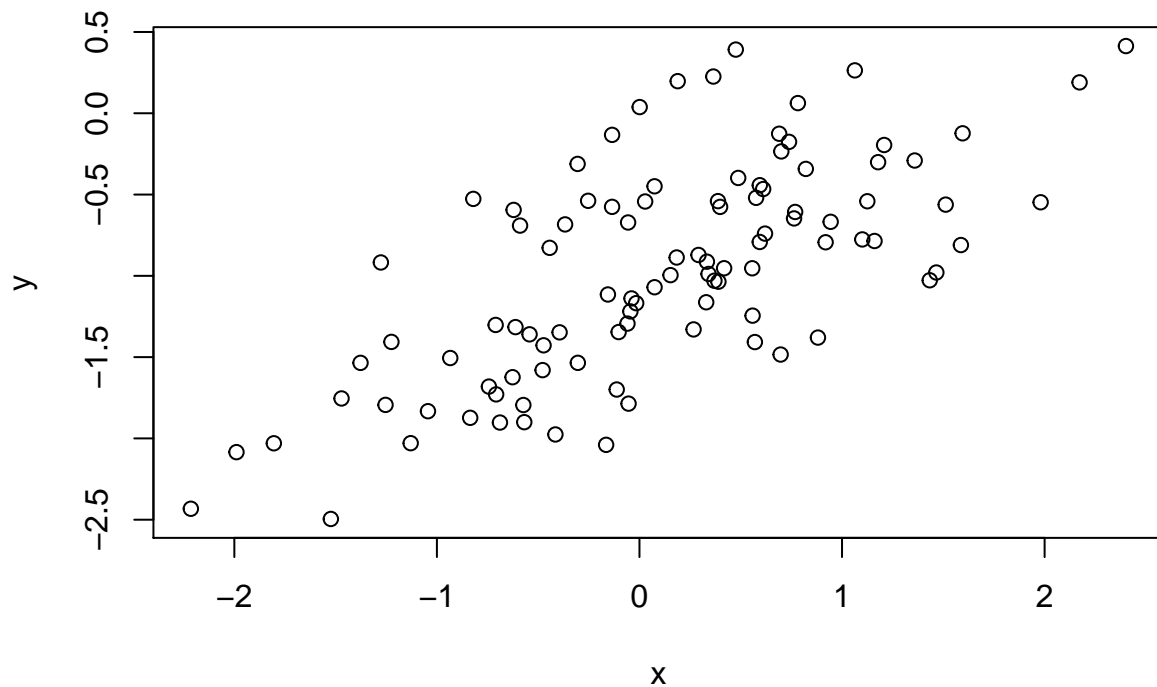
```
y <- -1 + 0.5 * x + eps
length(y)
```

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

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

d)

```
plot(x, y)
```



The relationship between x and y is moderately strong and positively linear. Because of eps, there is some noise introduced, which might be throwing the model off.

e)

```
fit9 <- lm(y ~ x)
summary(fit9)
```

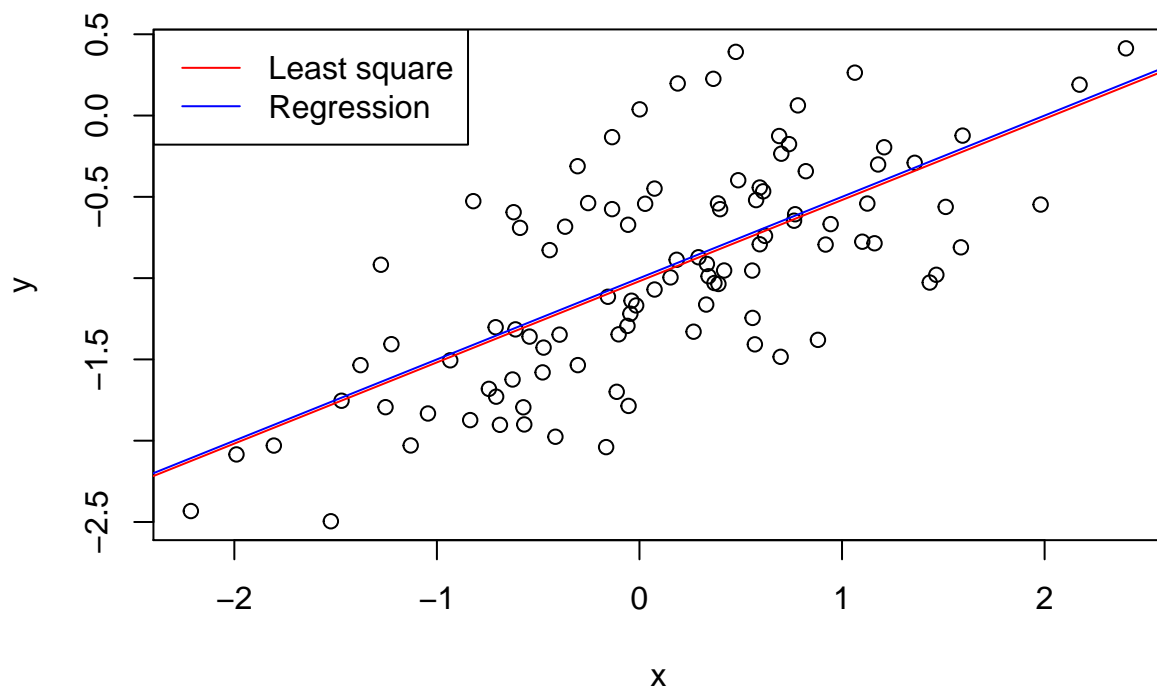


```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93842 -0.30688 -0.06975  0.26970  1.17309
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01885     0.04849  -21.010  < 2e-16 ***
## x              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 -1.02753 and 0.50698 respectively, which is pretty close to the true underlying values of -1 and 0.5. Since the p-value is significantly lower than 0.025, the null hypothesis can be rejected.

f)

```
plot(x, y)
abline(fit9, col = "red")
abline(-1, 0.5, col = "blue")
legend("topleft", c("Least square", "Regression"), col = c("red", "blue"), lty = c(1, 1))
```



g)

```
fit10 <- lm(y ~ x + I(x^2))
summary(fit10)
```

```
##
## 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
```

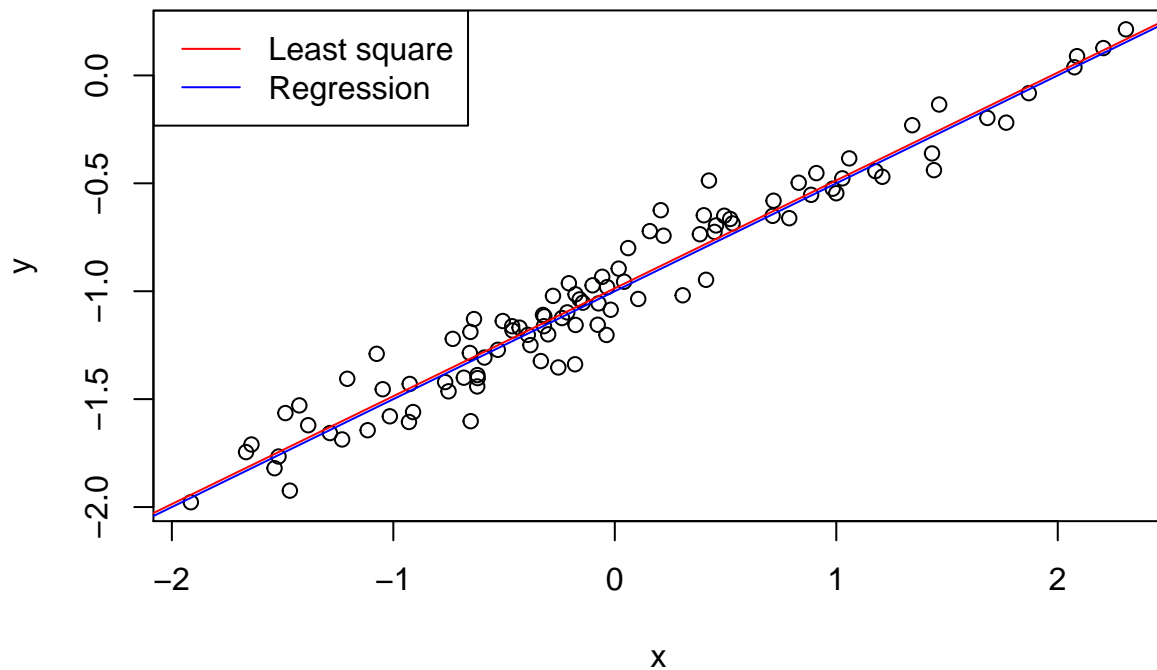
The p-value for x^2 is higher than 0.05, so it is not statistically significant. Therefore, we do not have enough evidence to show that the quadratic term improves the model fit.

h)

```
set.seed(1)
eps <- rnorm(100, sd = 0.125)
x <- rnorm(100)
y <- -1 + 0.5 * x + eps
plot(x, y)
fit11 <- lm(y ~ x)
summary(fit11)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29052 -0.07545  0.00067  0.07288  0.28664
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.98639    0.01129  -87.34  <2e-16 ***
## x            0.49988    0.01184   42.22  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1128 on 98 degrees of freedom
## Multiple R-squared:  0.9479, Adjusted R-squared:  0.9474
## F-statistic: 1782 on 1 and 98 DF,  p-value: < 2.2e-16

abline(fit11, col = "red")
abline(-1, 0.5, col = "blue")
legend("topleft", c("Least square", "Regression"), col = c("red", "blue"), lty = c(1, 1))
```



Since we decreased the variance of the normal distribution where the noise is drawn from, we have reduced the amount of noise in the data. The coefficients are still very close to the previous ones, but the relationship is almost perfectly linear, the R^2 correlation is a lot higher (up to 0.9479), and the RSE is much lower. Since there is less noise, the least squares and regression line also overlap a lot more.

i)

```
set.seed(1)
eps <- rnorm(100, sd = 0.5)
x <- rnorm(100)
y <- -1 + 0.5 * x + eps
plot(x, y)
fit12 <- lm(y ~ x)
summary(fit12)
```

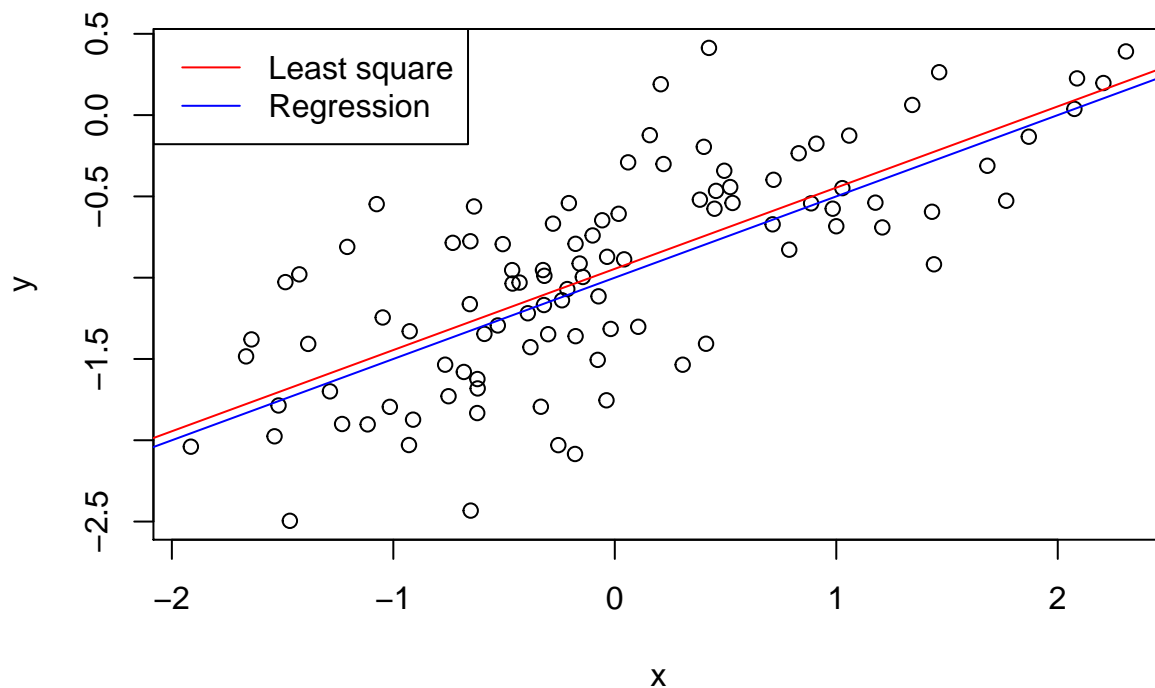
```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-1.16208	-0.30181	0.00268	0.29152	1.14658

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

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.94557	0.04517	-20.93	<2e-16 ***
x	0.49953	0.04736	10.55	<2e-16 ***

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4514 on 98 degrees of freedom
## Multiple R-squared:  0.5317, Adjusted R-squared:  0.5269
## F-statistic: 111.2 on 1 and 98 DF,  p-value: < 2.2e-16
abline(fit12, col = "red")
abline(-1, 0.5, col = "blue")
legend("topleft", c("Least square", "Regression"), col = c("red", "blue"), lty = c(1, 1))
```



After increasing the variance of the normal distribution where the noise is drawn from, we have increased the amount of noise in the data. The coefficients are still pretty close to their previous values, but the relationship between the data points isn't as linear as we just saw in h). The R^2 correlation is a decent amount lower (0.5317), and the RSE is a lot higher. The two lines, least square and regression, are pretty far apart compared to previous iterations, but are still relatively close.

j)

```
confint(fit9)
```

```
##                2.5 %    97.5 %
## (Intercept) -1.1150804 -0.9226122
## x           0.3925794  0.6063602
```

```
confint(fit11)
```

```
##                2.5 %    97.5 %
## (Intercept) -1.008805 -0.9639819
```

```
## x          0.476387  0.5233799
```

```
confint(fit12)
```

```
##              2.5 %      97.5 %  
## (Intercept) -1.0352203 -0.8559276  
## x           0.4055479  0.5935197
```

The confidence intervals for β_0 are pretty strongly centered around 0.0, while the intervals for β_1 are all still centered around 0.5. One thing to note is as noise increases, the confidence intervals widen, indicating we are less confident on the specific value of the intercept and coefficient. Less noise indicates more predictability.

Chapter 3 - Exercise 14

a)

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

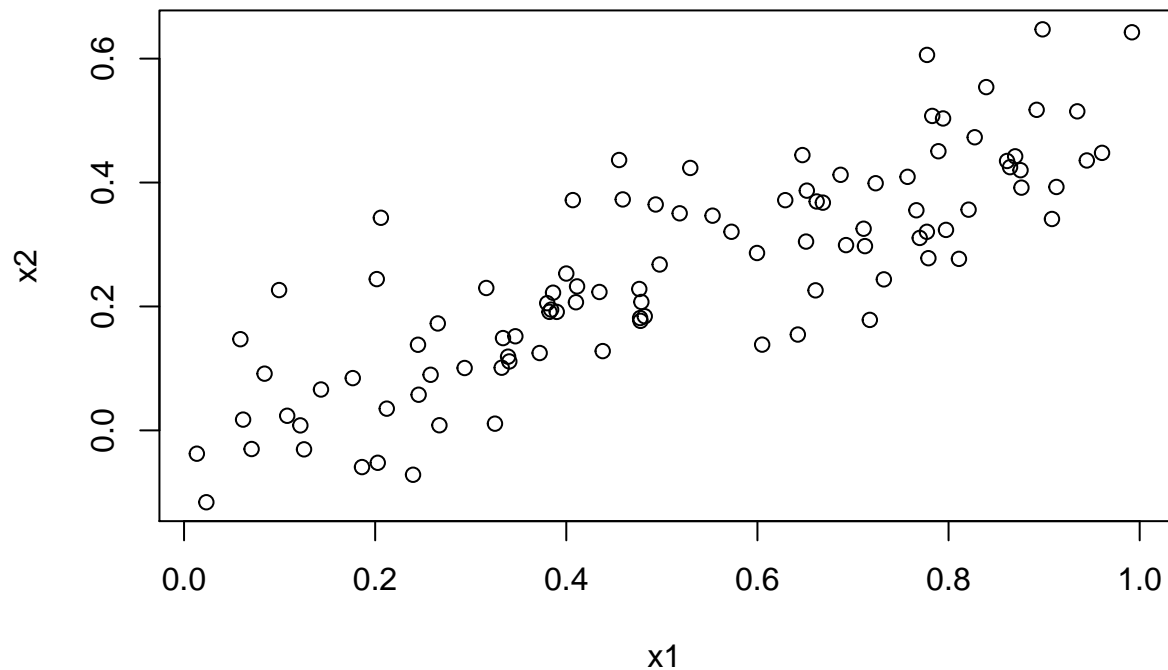
The linear model is $Y = 2 + 2X_1 + 0.3X_2 + \epsilon$. The regression coefficients are 2 for the intercept, 2 for X_1 , and 0.3 for X_2 .

b)

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

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

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



c)

```
fit13 <- lm(y ~ x1 + x2)
summary(fit13)
```

```
##
## 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
```

The coefficients are $\hat{\beta}_0 = 2.1305$, $\hat{\beta}_1 = 1.4396$, $\hat{\beta}_2 = 1.0097$. The only coefficient close to its underlying true value is $\hat{\beta}_0$. We can reject the null for $H_o : \hat{\beta}_1 = 0$ since the p-value is less than 0.05. We fail to reject the

null for $H_o : \hat{\beta}_2 = 0$ since the p-value is higher than 0.05.

d)

```
fit14 <- lm(y ~ x1)
summary(fit14)
```

```
##
## 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 for x1 has changed a lot compared to the x1 in the previous model containing both x1 and x2. We can reject the null since the p-value is extremely low (< 0.001).

e)

```
fit15 <- lm(y ~ x2)
summary(fit15)
```

```
##
## 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 for x2 is very different from its value in the original model containing both x1 and x2 (x2 was almost 1.0). We can reject the null since the p-value is extremely low (< 0.001).

f)

No, there is no contradiction. Both the x_1 and x_2 variables are highly correlated ($R^2 = 0.8351212$), so we are observing collinearity. As such, it is difficult to reasonably determine how much each predictor influences the response. You may recall we rejected H_o for x_1 , but failed to reject H_o for x_2 . This is partially due to collinearity. Since we failed to reject the null for x_2 , it seems like that variable is not important, but in fact its importance is masked by its collinearity with x_1 .

g)

```
x1 <- c(x1, 0.1)
x2 <- c(x2, 0.8)
y <- c(y, 6)
fit16 <- lm(y ~ x1 + x2)
fit17 <- lm(y ~ x1)
fit18 <- lm(y ~ x2)
summary(fit16)
```

```
##
## 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
```

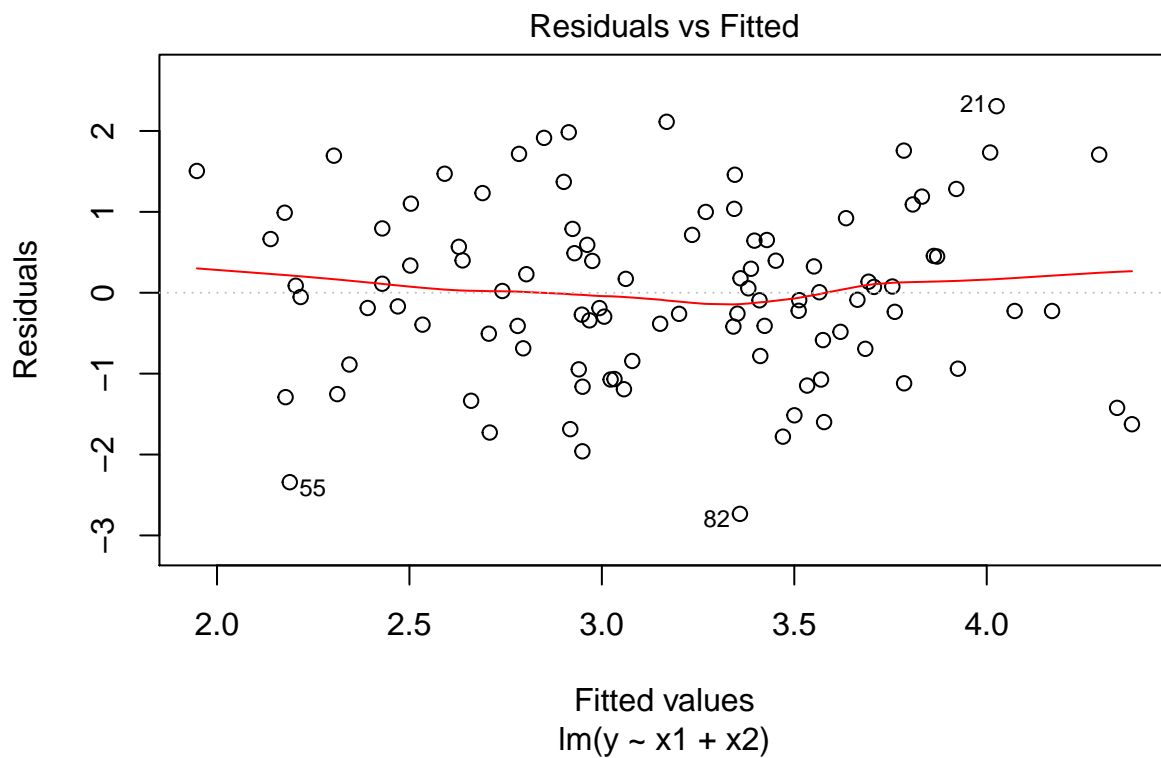
```
summary(fit17)
```

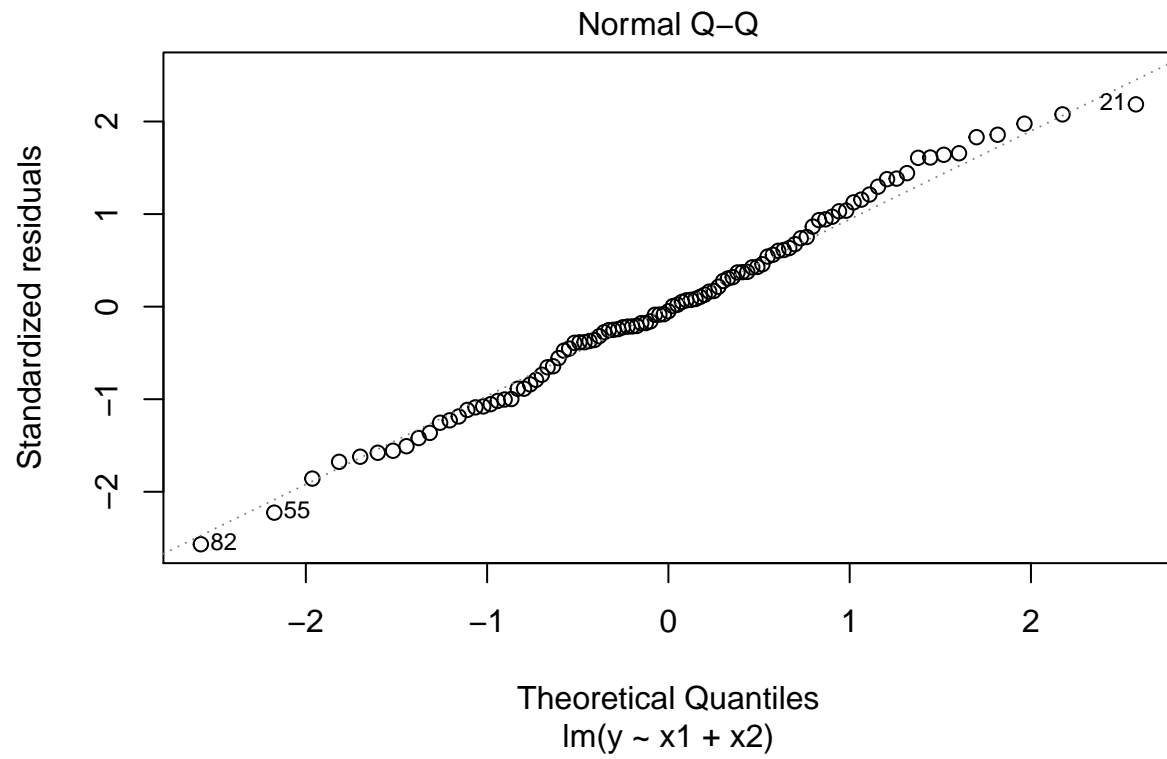
```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1             1.7657     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
```

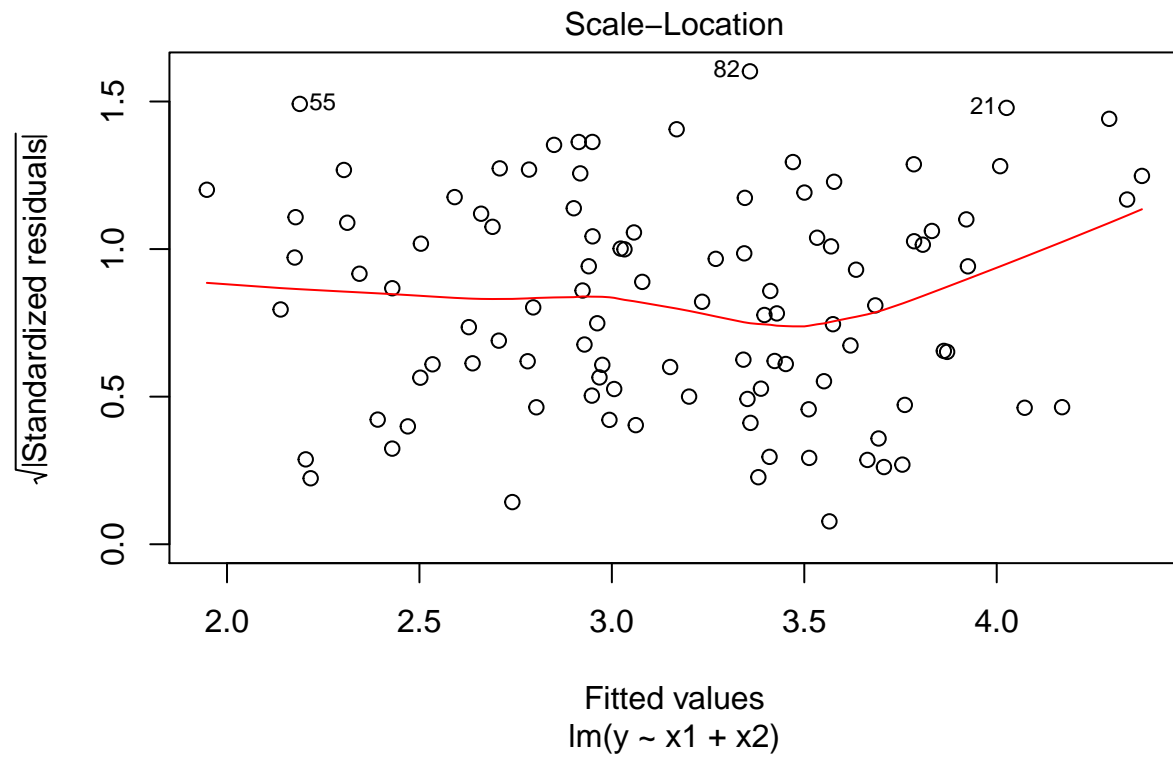
```
summary(fit18)
```

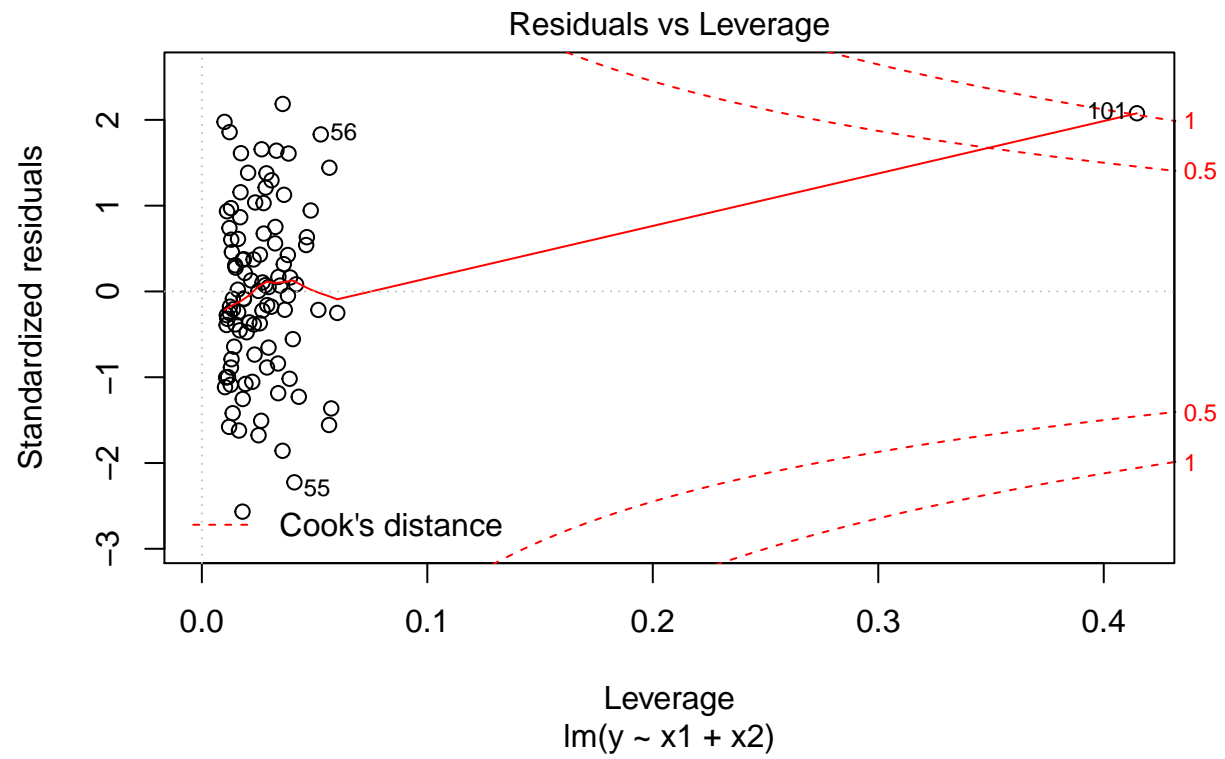
```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264 < 2e-16 ***
## x2            3.1190     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
```

```
plot(fit16)
```

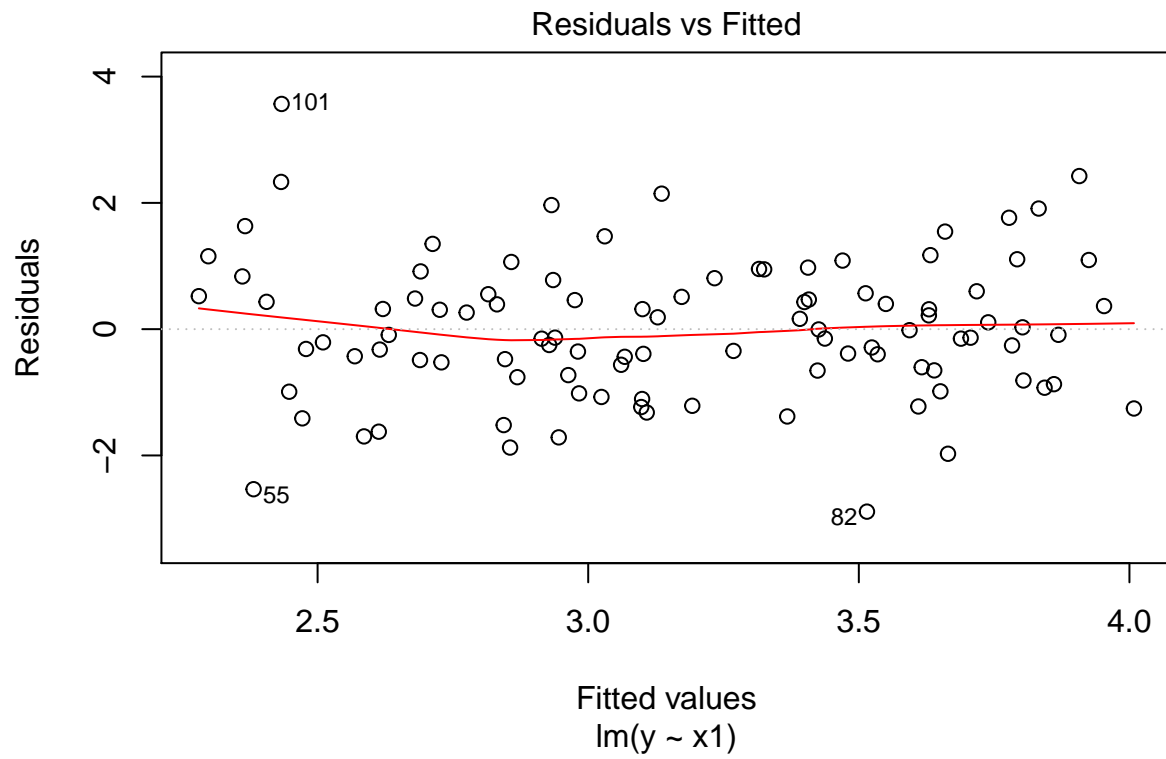


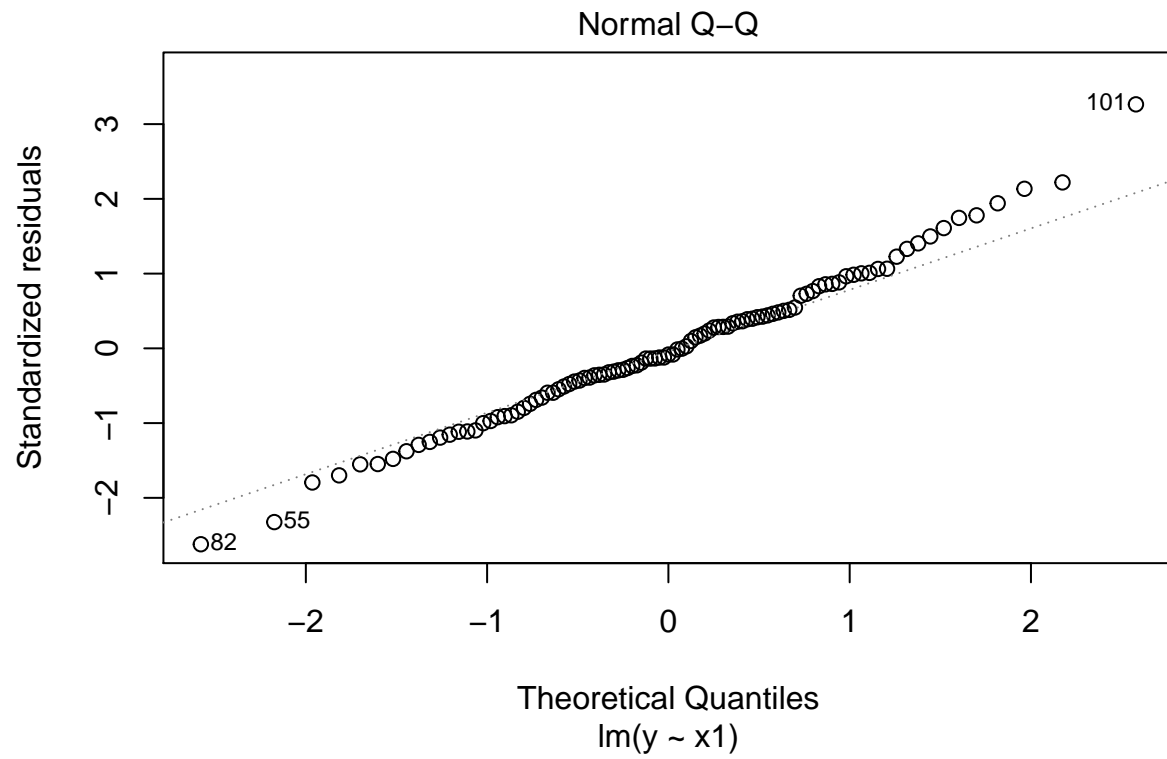


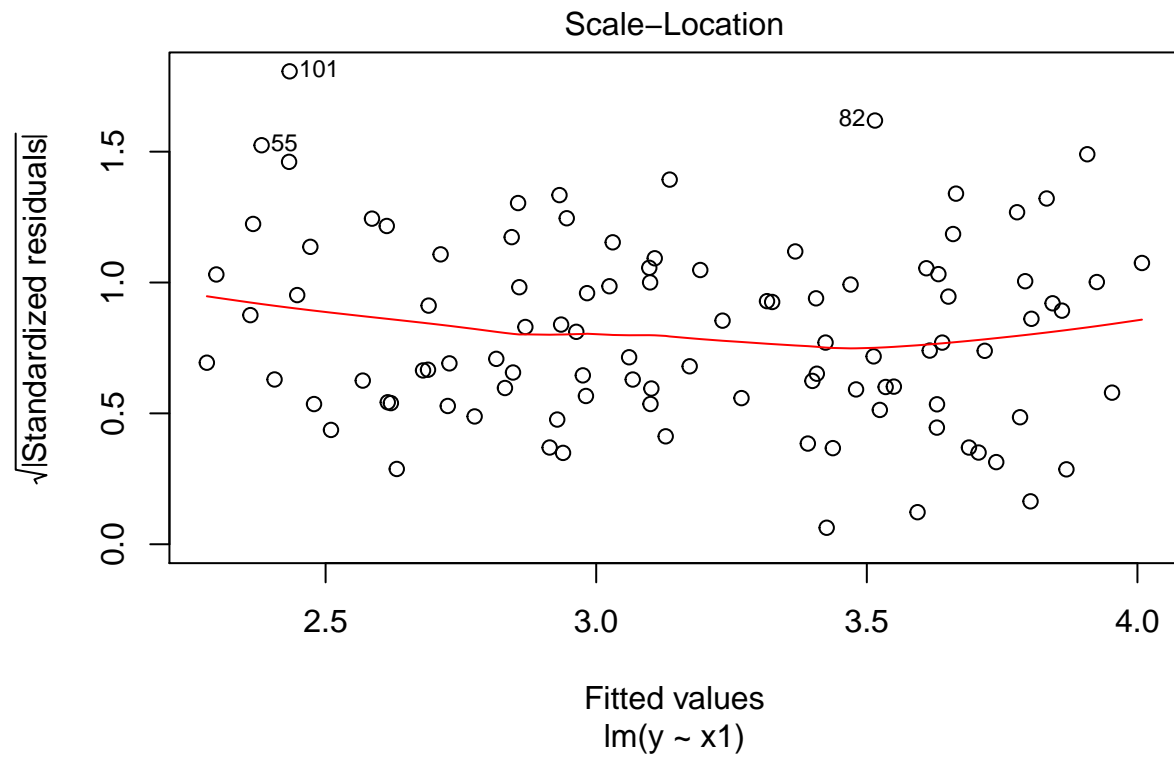


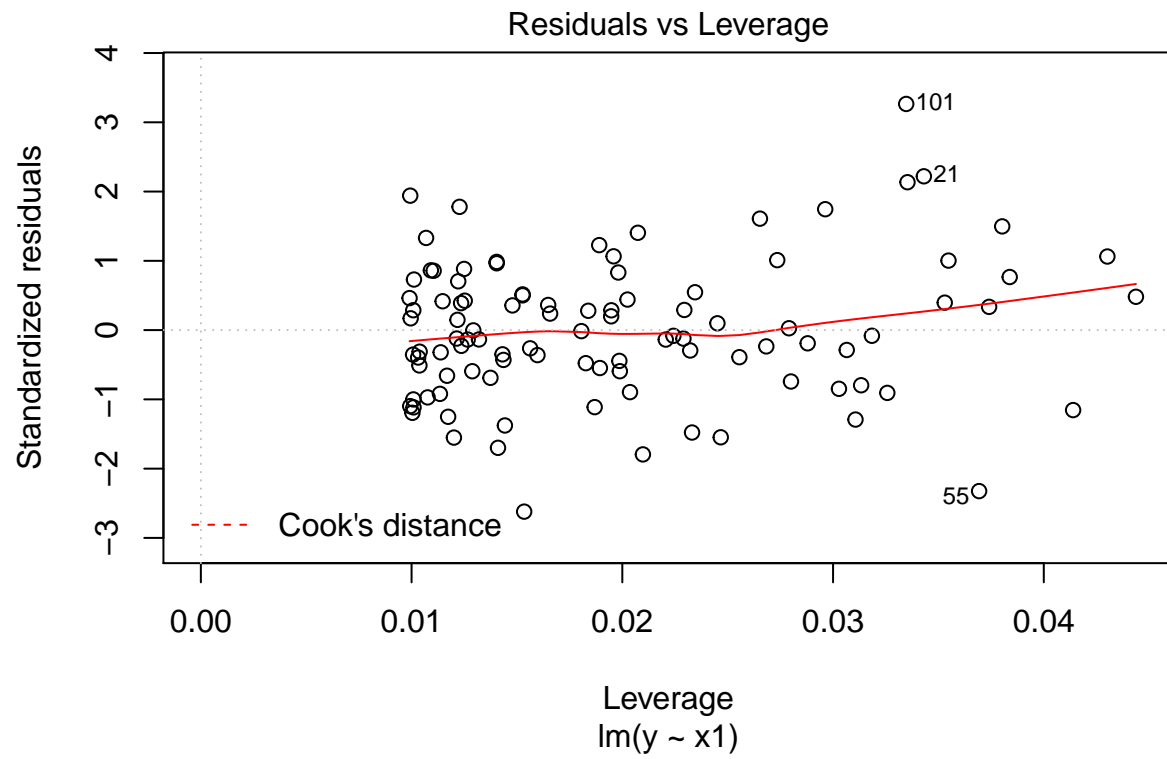


```
plot(fit17)
```

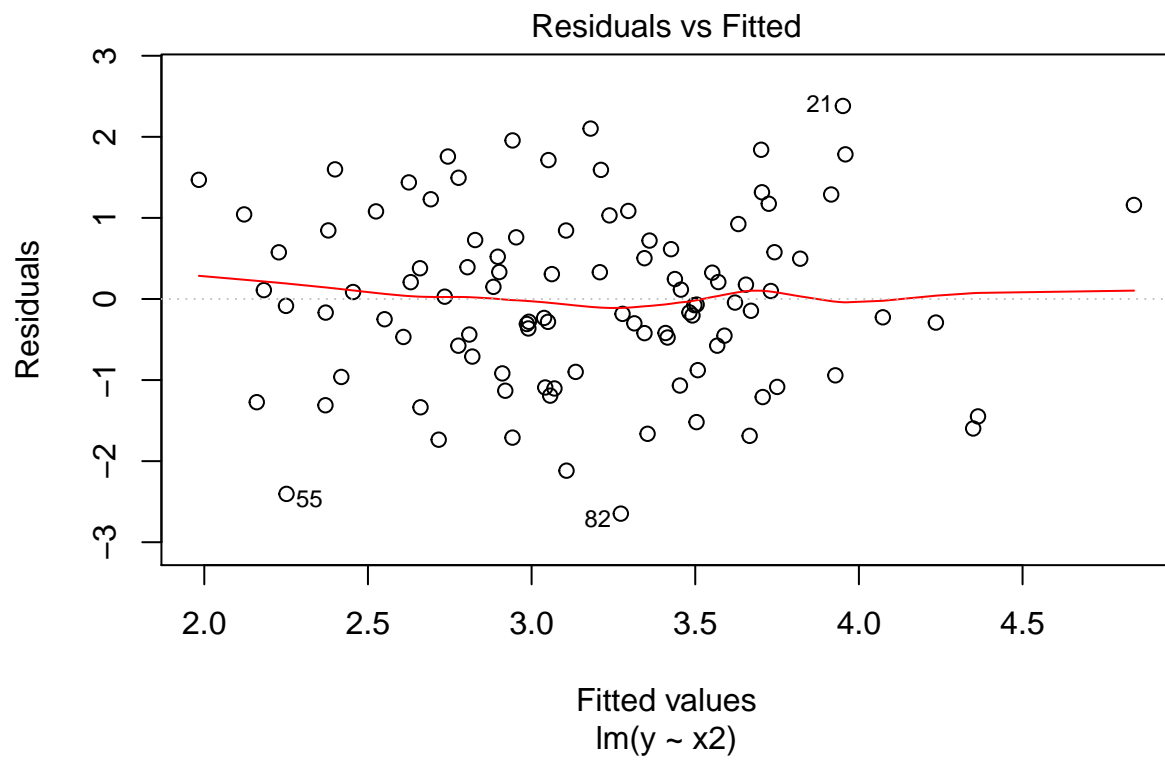


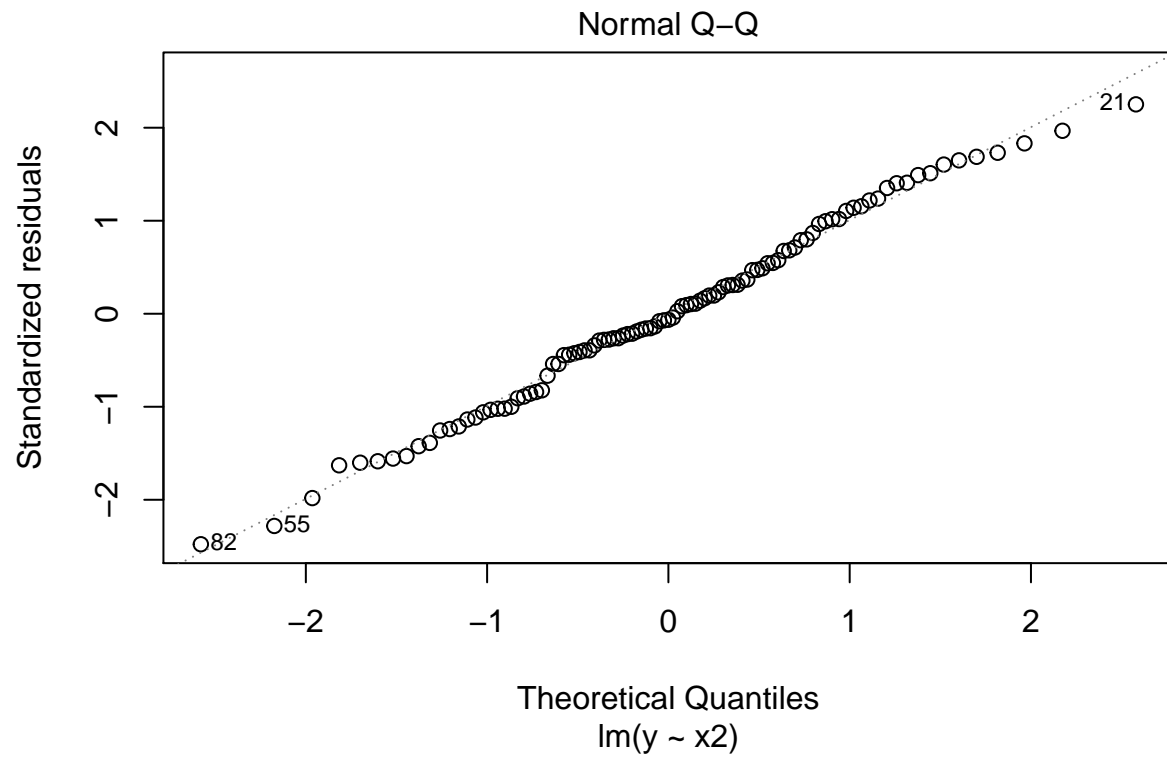


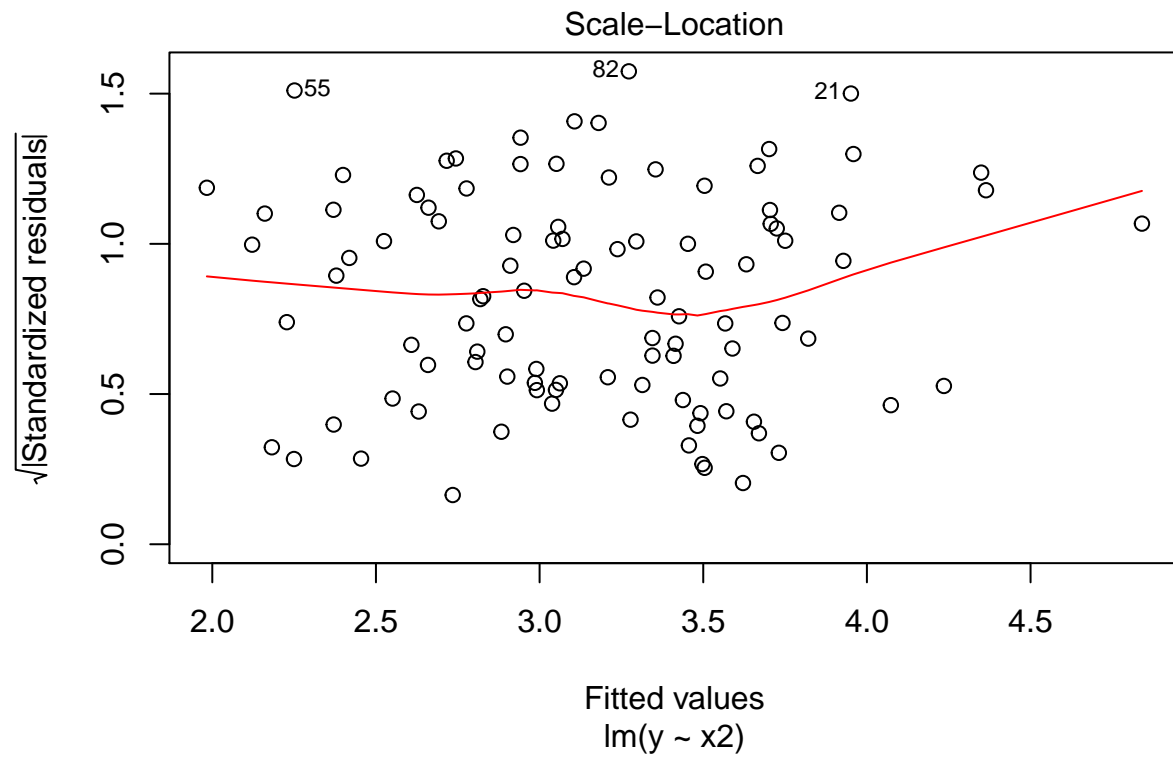


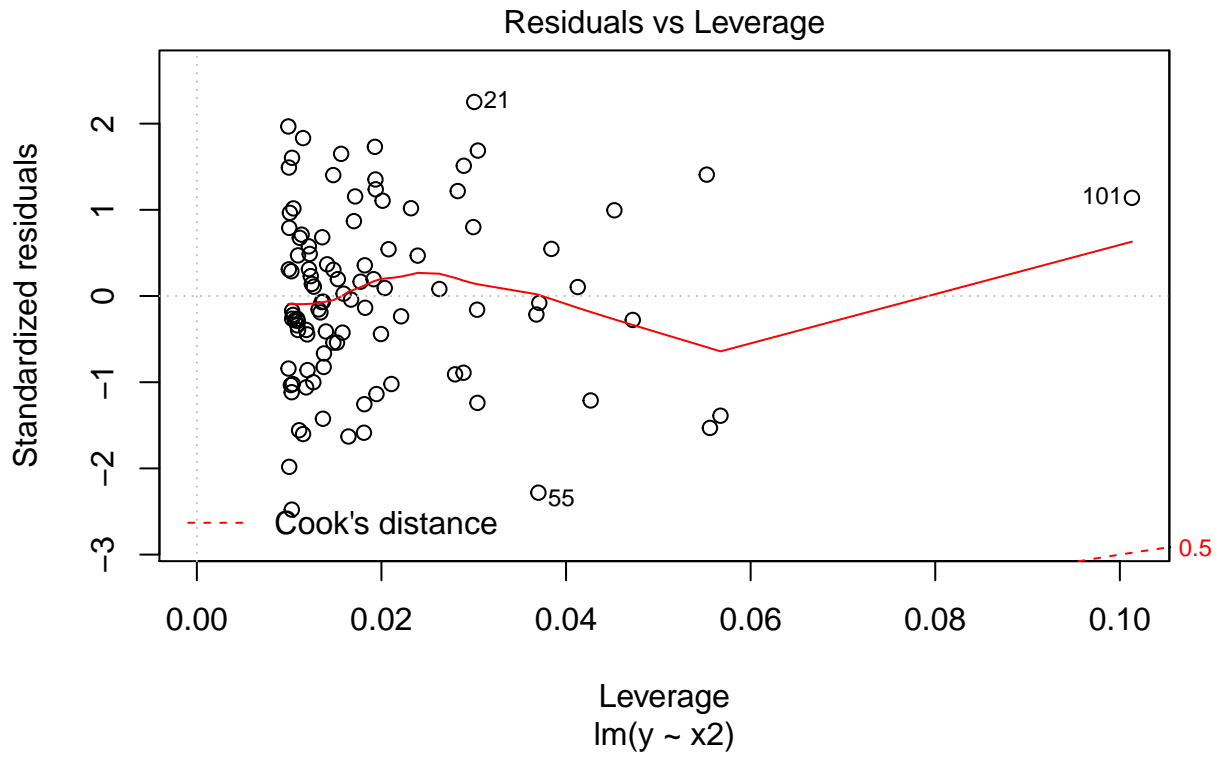


```
plot(fit18)
```









In the first model with both predictors, the last point is a high-leverage point. In the model with just x_1 , the last point is an outlier. In the model with just x_2 , the last point is a high leverage point. This last point increased the x_2 coefficient in the “ x_2 -only” model from 2.8 to 3.3, changed the x_1 coefficient in the “ x_1 -only” model from 2.0 to 1.6, and made x_1 almost non-existent in the “ x_1/x_2 ” model, completely flipping the roles of x_1 and x_2 in the combined model. As such, this new point completely changed the coefficient values of this data’s model.