Applied Data Science II - Homework 2

Phileas Dazeley Gaist

16/01/2021

ISLR 2.4: Questions 1, 2, 8, 10

Libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ISLR2)
```

1. ISLR 3.7 - 8

a)

```
auto <- read_csv("Homework 2 data/Auto.csv")  # load the data
auto$horsepower <- as.numeric(auto$horsepower)  # auto$horsepower is registered as
# a character variable, this must be changed before running the linear
# regression in order to get correct results.
auto <- auto %>%
    drop_na()  # required for regression and cor() later in exercise

lm_fit = lm(data = auto, mpg ~ horsepower)
summary(lm_fit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

i

Yes, there is a significant relationship between horsepower and mpg. Since the F-statistic $F > 1$ and the p-value of the F-statistic $p < 0.001$, we can reject the null hypothesis.

ii

With an $R^2 = 0.6049$ for our linear regression, we know that 60.49 of the variance in mpg is explained by the horsepower variable.

iii

The relationship is negative: For higher values of horsepower, we expect lower values of mpg.

iv

```r
predict(lm_fit, data.frame(horsepower = c(98)), interval = "confidence")
```

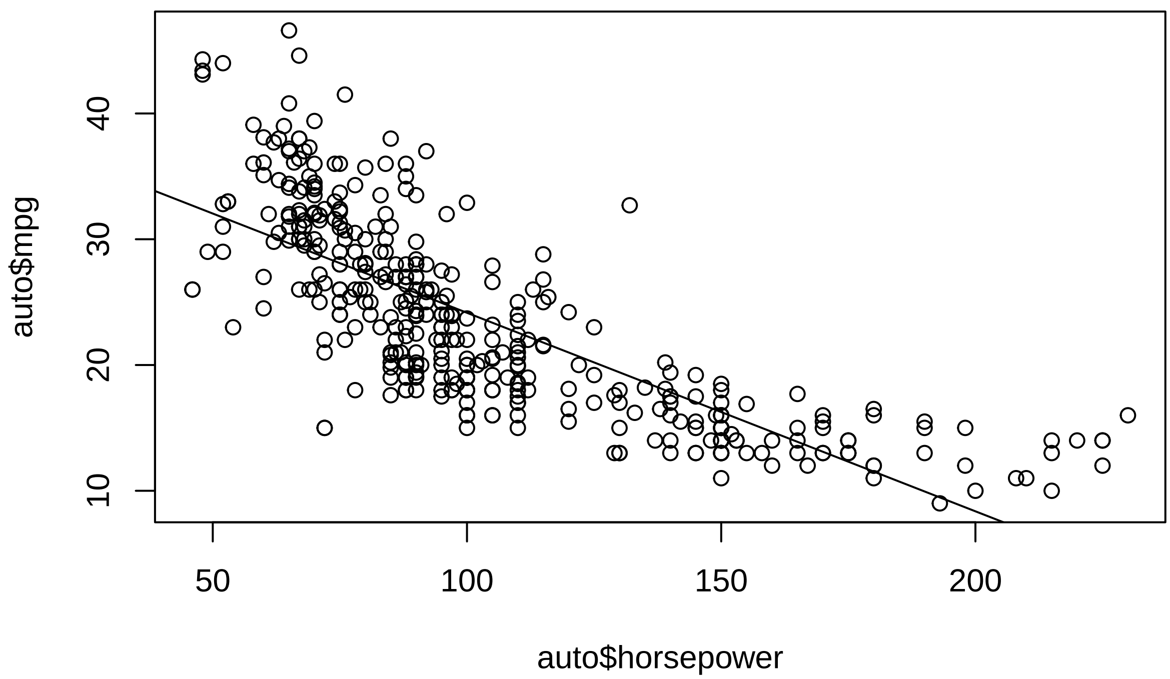
```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

```r
predict(lm_fit, data.frame(horsepower = c(98)), interval = "prediction")
```

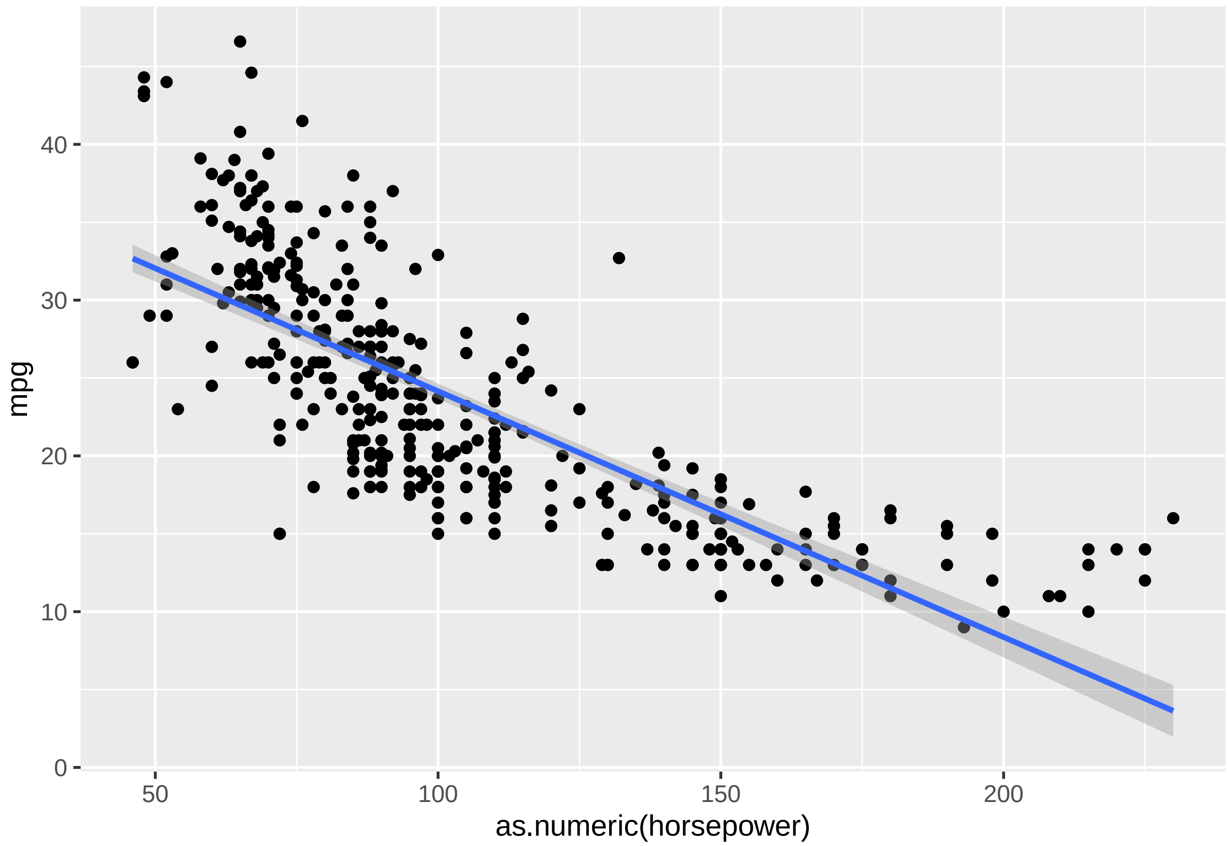
```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

b)

```r
# plot
plot(auto$horsepower, auto$mpg)
abline(lm_fit)
```



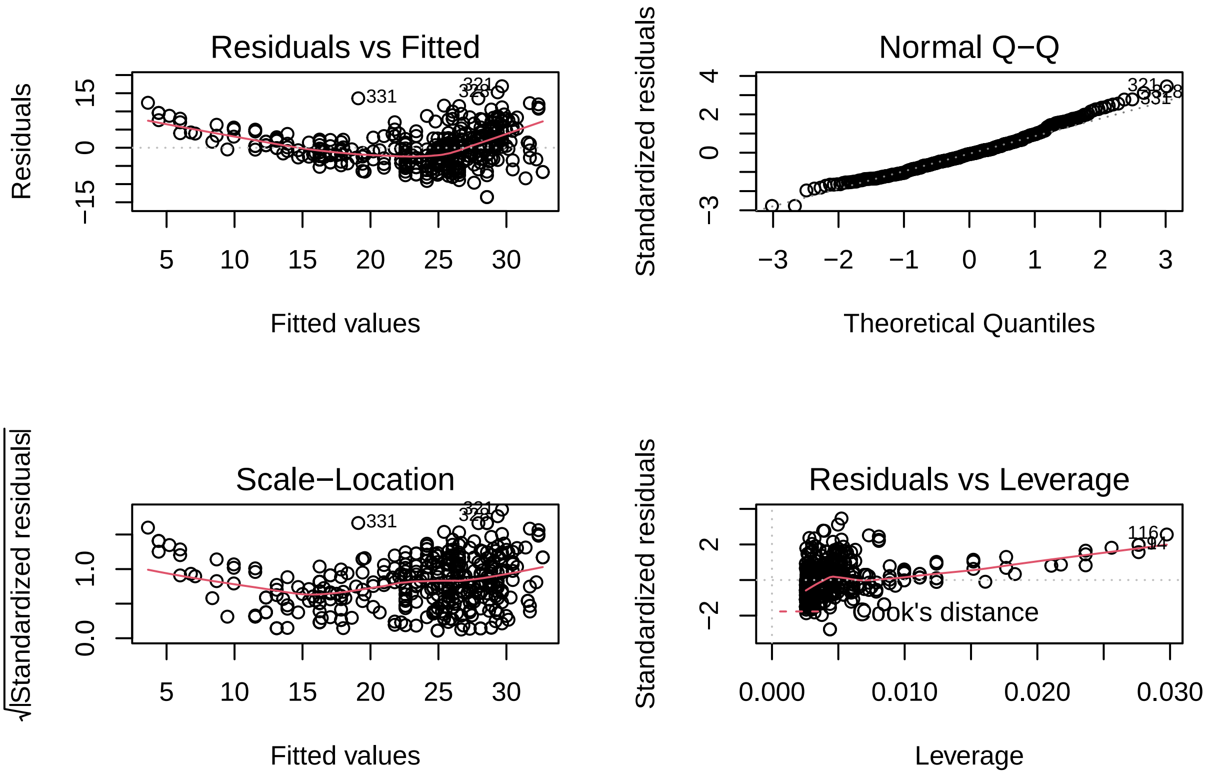

```r
# alternative ggplot version
auto %>%
    ggplot(aes(as.numeric(horsepower), mpg)) + geom_point() + geom_smooth(method = "lm",
    formula = y ~ x)
```



c)

The residuals vs fitted plot indicates some nonlinearity. The scale-location plot indicates heteroscedasticity.

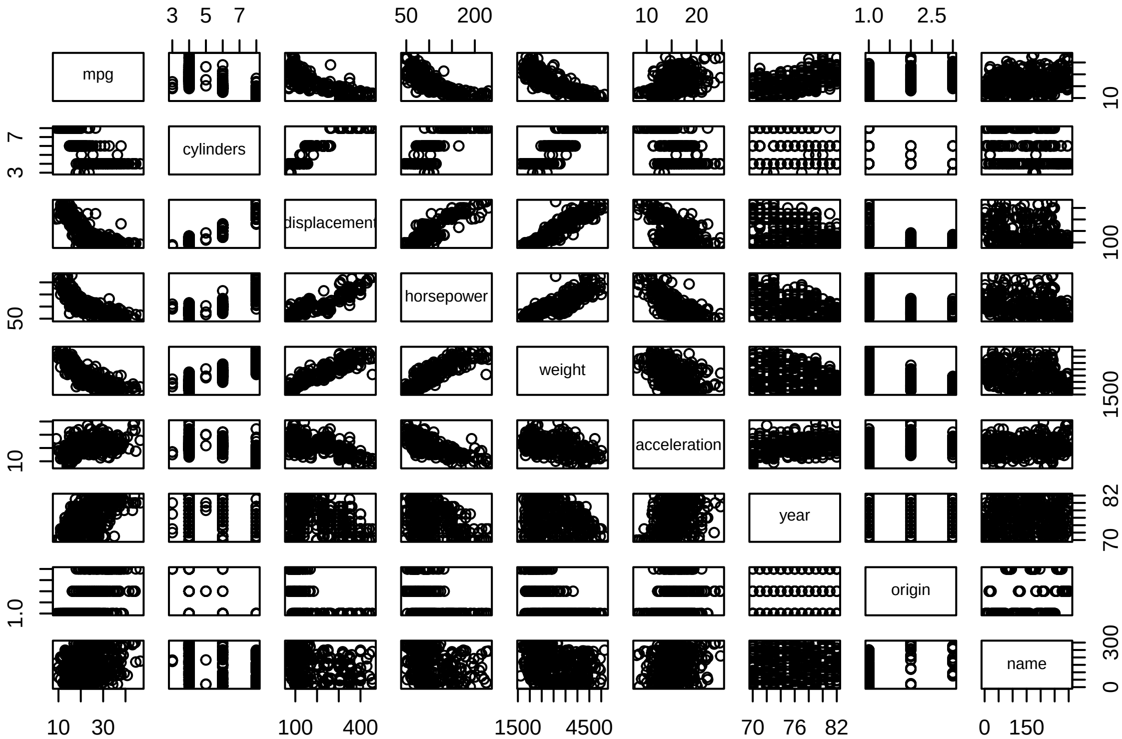
```
# diagnostics plots
par(mfrow = c(2, 2))
plot(lm_fit)
```



1. ISLR 3.7 - 9

a)

```
plot(auto)
```



b)

```r
cor(subset(auto, select = -name))
```

```
##                     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)

```r
lm_fit_multiple <- lm(mpg ~ . - name, data = auto)
summary(lm_fit_multiple)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

i

There is a relationship between the predictors and the response. We reject the null hypothesis: $F = 252.4$, $p < 0.001$

ii

Displacement, weight, year, and origin were statistically significantly related to the response variable mpg.

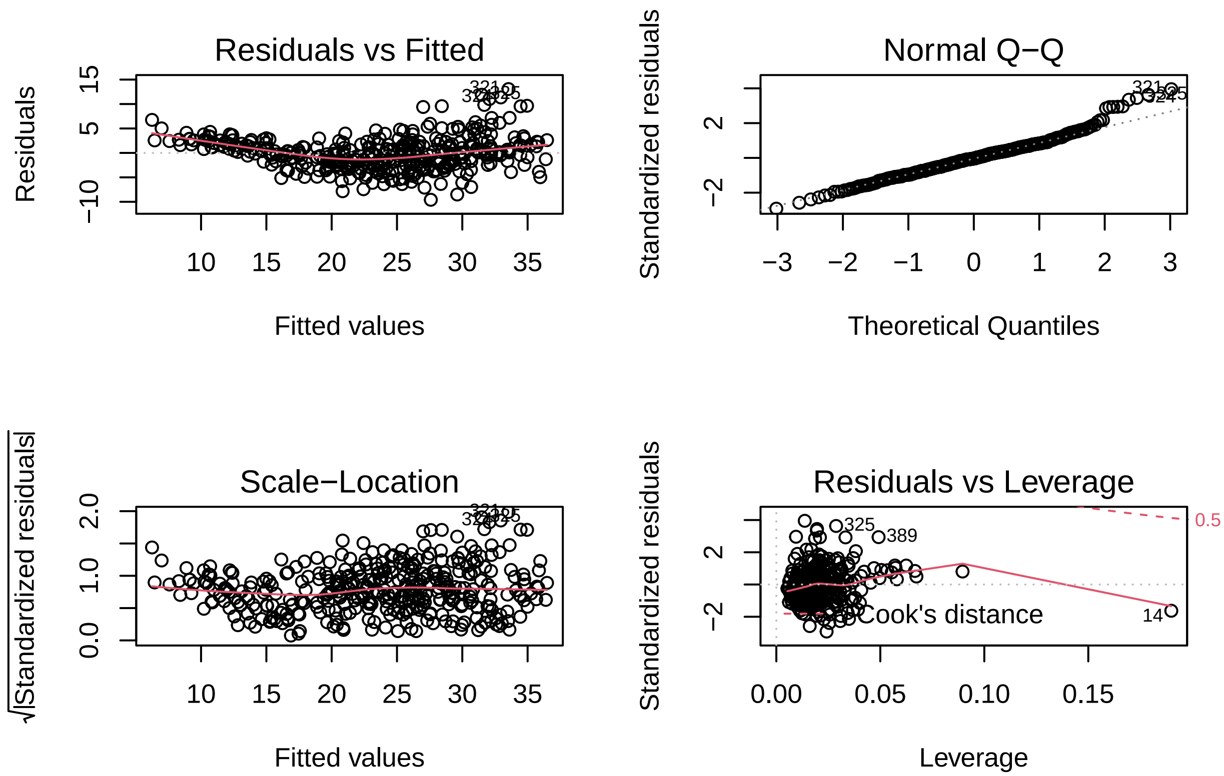
iii

The coefficient of regression for the year variable shows that every year, the mpg variable increases by an average of 0.750773.

d)

Linearity and assumption of normality seem to hold, as well as homoscedasticity. The residuals vs leverage plots indicates that point 14 might be worth looking at further as an outlier in the data, which could bear influence on the regression.

```r
par(mfrow = c(2, 2))
plot(lm_fit_multiple)
```



e)

I could have checked the correlation matrix to decide on which variables to use as interaction terms, but I decided to just pick two variables which I imagined would have an interaction instead.

There is a significant interaction between the acceleration and weight variables.

```r
summary(lm(mpg ~ weight * acceleration, data = auto))  # there is a significant interaction be
```

```
##
## Call:
## lm(formula = mpg ~ weight * acceleration, data = auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5823  -2.6411  -0.3517   2.2611  15.6704
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.814e+01  4.872e+00   5.776 1.57e-08 ***
## weight              -3.168e-03  1.461e-03  -2.168  0.03076 *
## acceleration         1.117e+00  3.097e-01   3.608  0.00035 ***
## weight:acceleration -2.787e-04  9.694e-05  -2.875  0.00426 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.249 on 388 degrees of freedom
## Multiple R-squared:  0.706,  Adjusted R-squared:  0.7037
## F-statistic: 310.5 on 3 and 388 DF,  p-value: < 2.2e-16
```

```
summary(lm(mpg ~ weight:acceleration, data = auto))  # checking the interaction by itself
```

```
##
## Call:
## lm(formula = mpg ~ weight:acceleration, data = auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6424  -4.1342  -0.5959   3.8714  23.7401
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          4.053e+01  1.245e+00   32.55   <2e-16 ***
## weight:acceleration -3.772e-04  2.656e-05  -14.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.345 on 390 degrees of freedom
## Multiple R-squared:  0.3407, Adjusted R-squared:  0.3391
## F-statistic: 201.6 on 1 and 390 DF,  p-value: < 2.2e-16
```

f)

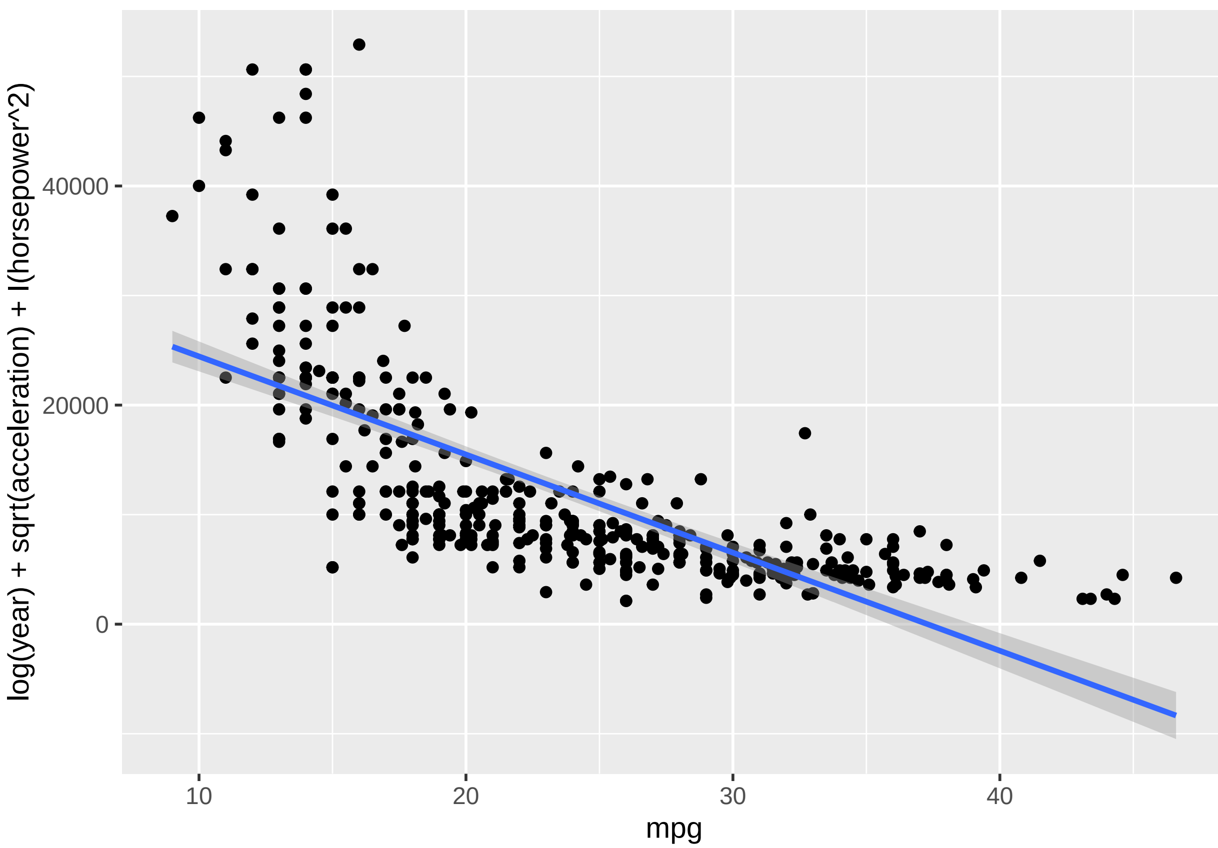
I messed around completely at random and everything is still significantly related to everything else. Perhaps a sign of autocorrelated variables in the data set? Although the assumptions of linear regression are not met. There is indication of nonlinearity in the residuals vs fitted plot, evidence of outliers in the residuals vs leverage plot, and evidence of heteroscedasticity in the scale-location plot.

```
summary(lm(data = auto, mpg ~ log(year) + sqrt(acceleration) + I(horsepower^2)))
```

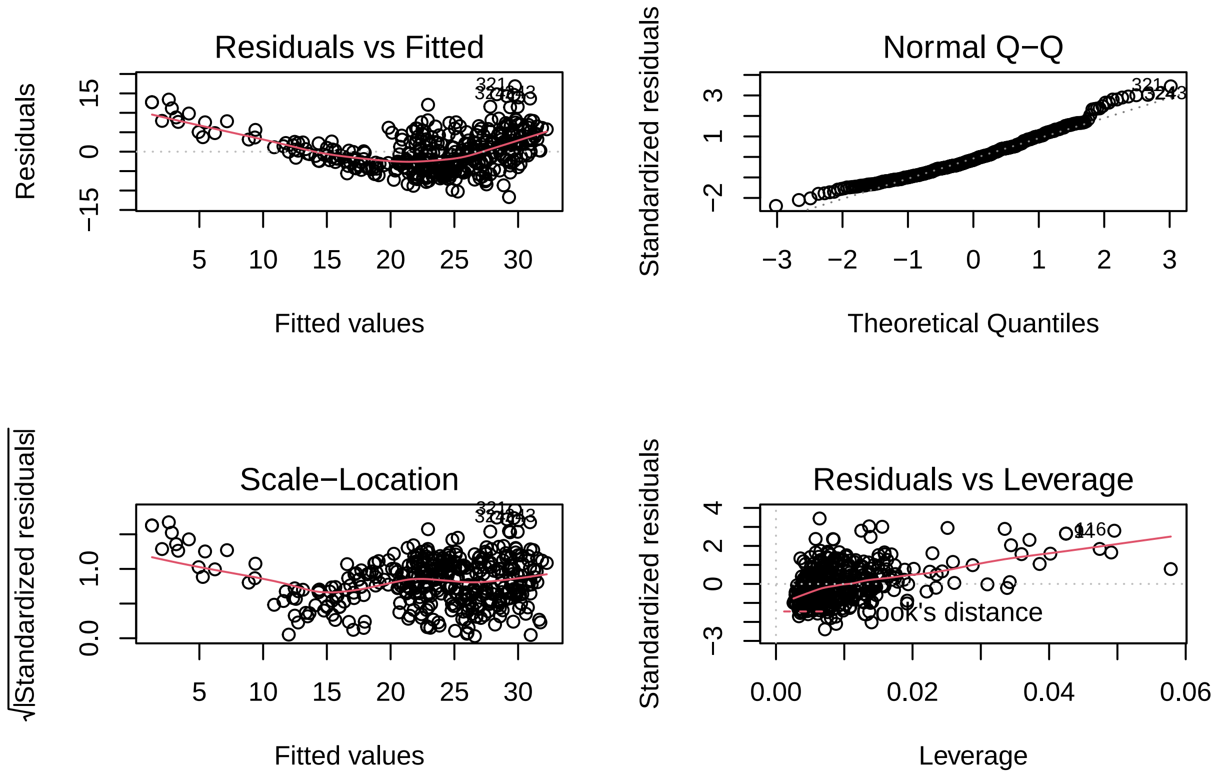
```
##
## Call:
## lm(formula = mpg ~ log(year) + sqrt(acceleration) + I(horsepower^2),
##     data = auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.688  -3.615  -0.702   2.881  16.843
##
```

```
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)         -1.964e+02  2.477e+01  -7.930 2.35e-14 ***
## log(year)            5.457e+01  5.634e+00   9.686  < 2e-16 ***
## sqrt(acceleration) -2.548e+00  9.674e-01  -2.634  0.00877 ** 
## I(horsepower^2)     -5.153e-04  3.629e-05 -14.199  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 388 degrees of freedom
## Multiple R-squared:  0.6079, Adjusted R-squared:  0.6048 
## F-statistic: 200.5 on 3 and 388 DF,  p-value: < 2.2e-16
```

```r
auto %>%
    ggplot(aes(mpg, log(year) + sqrt(acceleration) + I(horsepower^2))) + geom_point() +
    geom_smooth(method = "lm")
```




```r
par(mfrow = c(2, 2))
plot(lm(data = auto, mpg ~ log(year) + sqrt(acceleration) + I(horsepower^2)))
```



1. ISLR 3.7 - 10

a)

```
head(Carseats)
```

```
##   Sales CompPrice Income Advertising Population Price ShelveLoc Age Education
## 1  9.50       138     73          11        276   120       Bad  42        17
## 2 11.22       111     48          16        260    83      Good  65        10
## 3 10.06       113     35          10        269    80    Medium  59        12
## 4  7.40       117    100           4        466    97    Medium  55        14
## 5  4.15       141     64           3        340   128       Bad  38        13
## 6 10.81       124    113          13        501    72       Bad  78        16
##   Urban  US
## 1   Yes Yes
## 2   Yes Yes
## 3   Yes Yes
## 4   Yes Yes
## 5   Yes  No
## 6    No Yes
```

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

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

b)

- Price: The regression reports a statistically significant negative relationship between Price and Pales ($p < 0.001$). For every unit Price increase, Sales decrease by approximately 0.05.
- UrbanYes: There is no relationship between UrbanYes and Sales ($p > 0.05$).
- USYes: There is a statistically significant positive relationship between USYes and Sales ($p < 0.001$).

c)

$Sales = coef_1 \cdot Intercept + coef_2 \cdot coefPrice + coef_3 \cdot UrbanYes + coef_4 \cdot USYes$ $Sales = 13.043469 - 0.054459xPrice - 0.02191xUrban + 1.200573xUS$

d)

We can reject the null hypothesis H_0 for the Price and US variables.

e)

```r
lm_fit_2 <- lm(data = Carseats, Sales ~ Price + US)
summary(lm_fit_2)
```

```
##
## 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)

- a model $R^2 = 0.2335$
- e model $R^2 = 0.2354$

The e model is slightly better than the a model.

g)

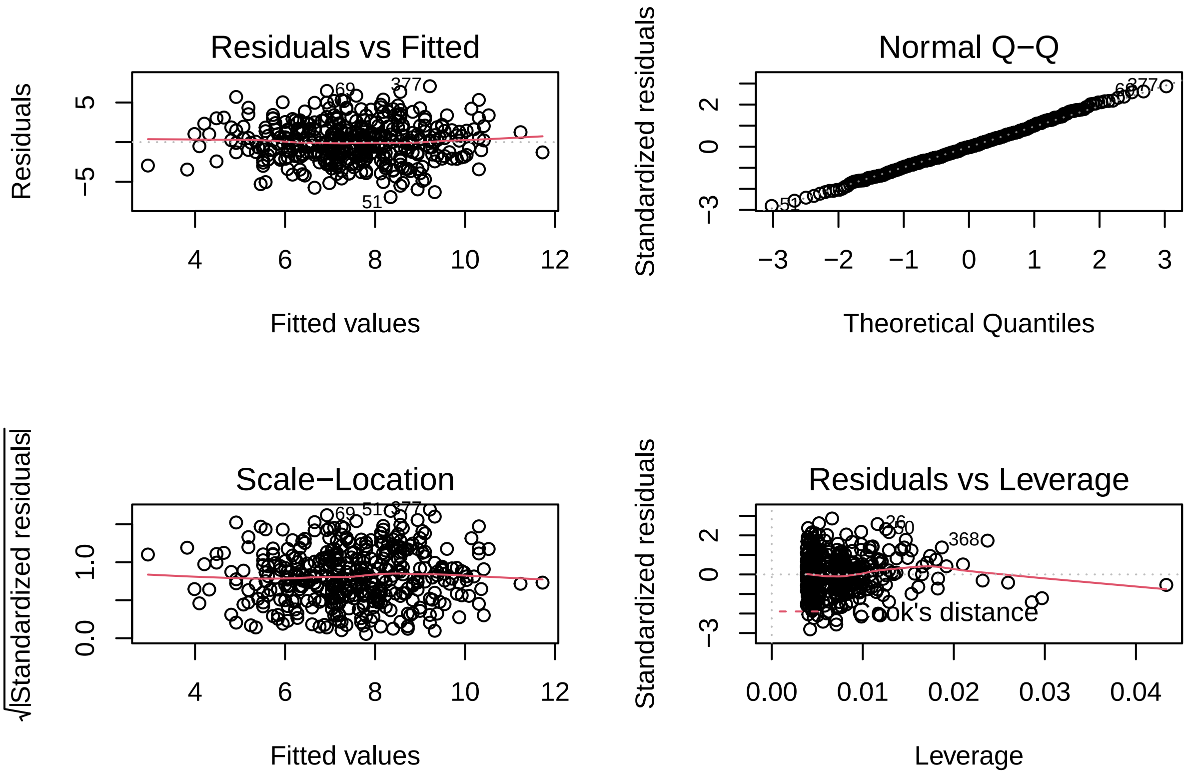
```r
confint(lm_fit_2)  # defaults to 95%
```

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

h)

The residuals vs leverage plot indicates the presence of some outliers, including the presence of points of leverage which merit further investigation.

```r
par(mfrow = c(2, 2))
plot(lm_fit_2)
```



1. ISLR 3.7 - 11

```
set.seed(1)
x <- rnorm(100)
y <- 2 * x + rnorm(100)
```

a)

- The coefficient estimate $\widehat{\beta} = 1.9939$
- Standard error $= 0.1065$
- t-statistic $= 18.73$
- $p < 0.001$

We reject $H_0 : \beta = 0$

```
summary(lm(y ~ x + 0))
```

```
##
## Call:
## lm(formula = y ~ x + 0)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1.9154 -0.6472 -0.1771  0.5056  2.3109
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   1.9939     0.1065   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9586 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

b)

- The coefficient estimate $\widehat{\beta} = 0.39111$
- Standard error $= 0.02089$
- t-statistic $= 18.73$
- $p < 0.001$

We reject $H_0 : \beta = 0$

```
summary(lm(x ~ y + 0))
```

```
##
## Call:
## lm(formula = x ~ y + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8699 -0.2368  0.1030  0.2858  0.8938
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## y  0.39111    0.02089   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4246 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

c)

We get the same t statistic and p-value, which in both cases allows us to reject H_0. This makes sense because we are performing the same regression both times. The equations for both regression lines are equal.

1. ISLR 3.7 - 13

a)

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

b)

```
eps <- rnorm(100, sd = sqrt(0.25))  # (standard deviation is the square root of variance)
```

c)

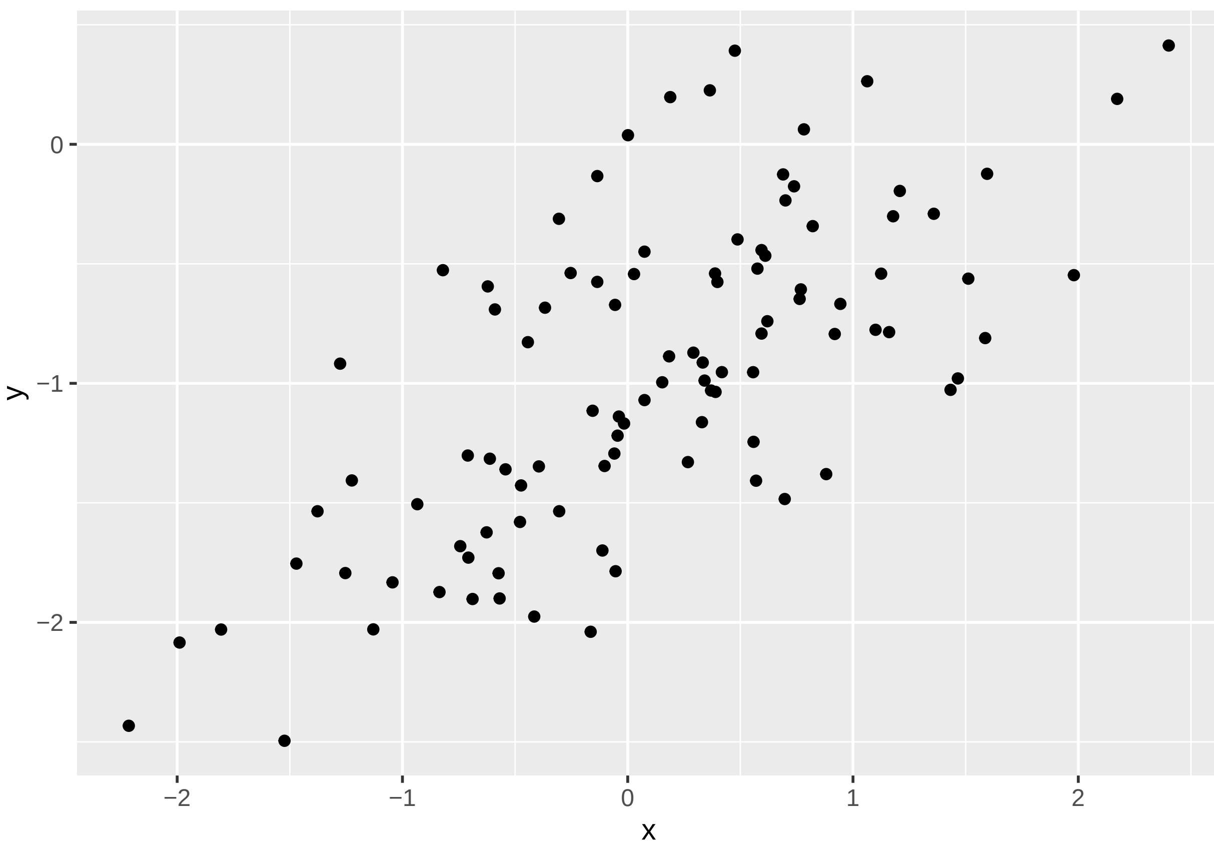
length(y) = 100 (same as x) $\beta_0 = -1$ $\beta_1 = 0.5$

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

d)

The scatter plot indicates a positive linear relationship between x and y (which we know exists since we made up the data). The variance corresponds to the variance introduced using the eps variable.

```
# plot(x, y)
tibble(x, y) %>%
    ggplot(aes(x, y)) + geom_point()
```



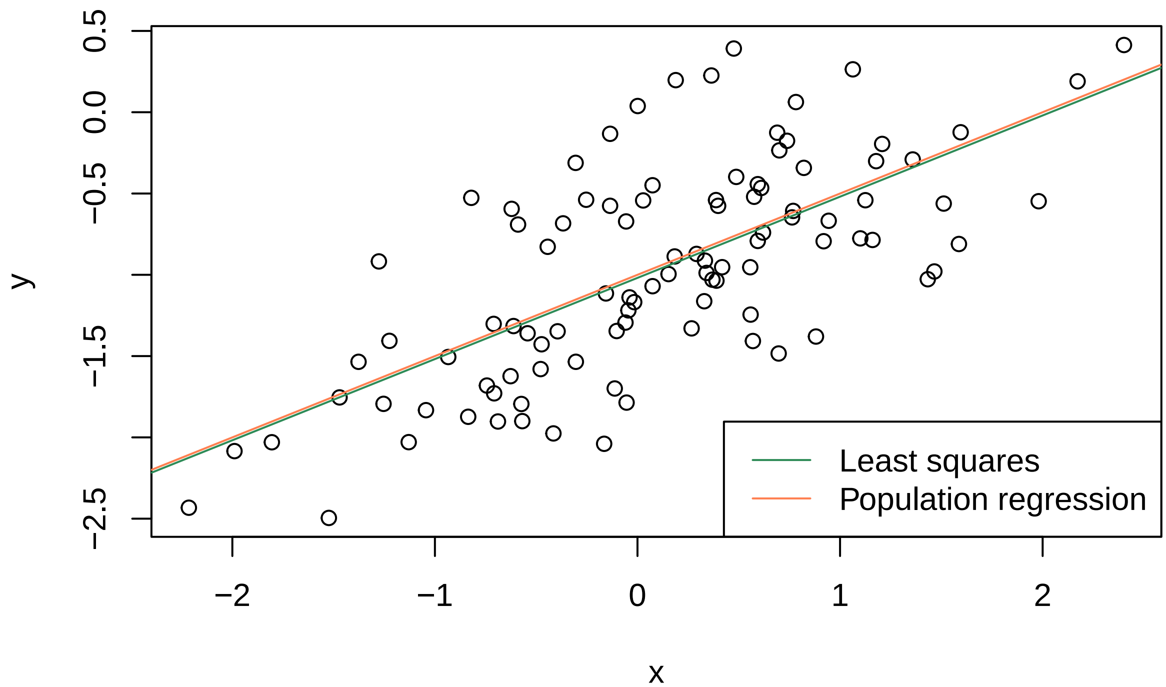
e) The estimated intercept and slope $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are very close to β_0 and β_1 (consistent with the way we generated the data to be linearly related, and introduced noise). With a F-statistic = 132.1 and $p < 0.001$, we reject H_0

```r
original <- lm(y ~ x)
summary(original)
```

```
##
## 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
```

f)

```r
plot(x, y)
abline(original, col = "seagreen")
abline(-1, 0.5, col = "coral")
legend("bottomright", c("Least squares", "Population regression"), col = c("seagreen",
    "coral"), lty = c(1, 1))
```




```
# ggplot version (commented out because I'm a little lost how I would add the
# legend given that it's not about data included in the data frame directly,
# but about the plotted lines; I would love some tips on this!)

# tibble(x, y) %>% ggplot(aes(x, y)) + geom_point() + geom_smooth(method =
# 'lm', colour = 'seagreen') + geom_abline(intercept = -1, slope = 0.5, colour
# = 'coral')
```

g)

There is no evidence that the quadratic term improves the model fit, as the coefficient for the x^2 term is not statistically significant.

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

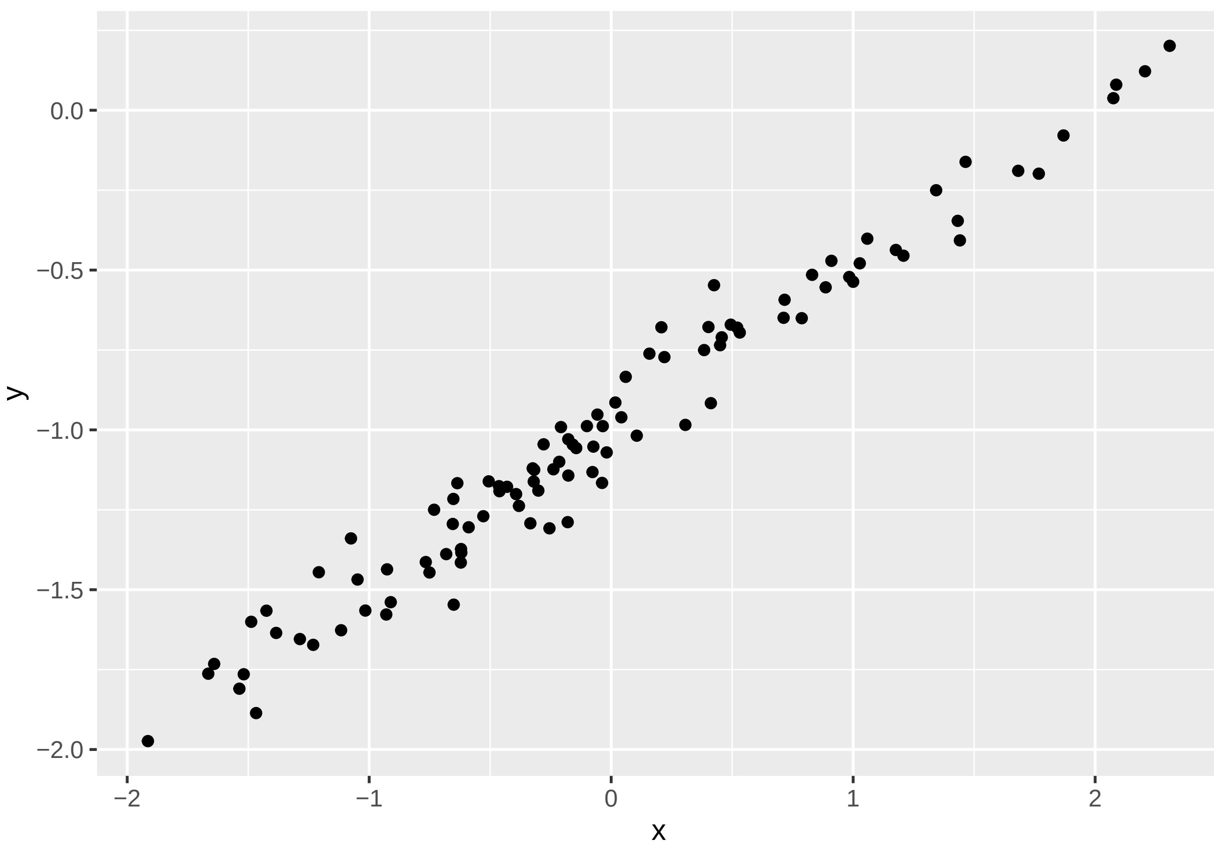
```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98252 -0.31270 -0.06441  0.29014  1.13500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.97164    0.05883 -16.517  < 2e-16 ***
## x            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
```

h)

By decreasing the noise in the y variable with a reduced variance of eps (resulting in a decreased error term ϵ), the same model achieves a better fit. R^2 is higher, RSE is lower. The population regression and least squares line are now much closer, and almost overlap.

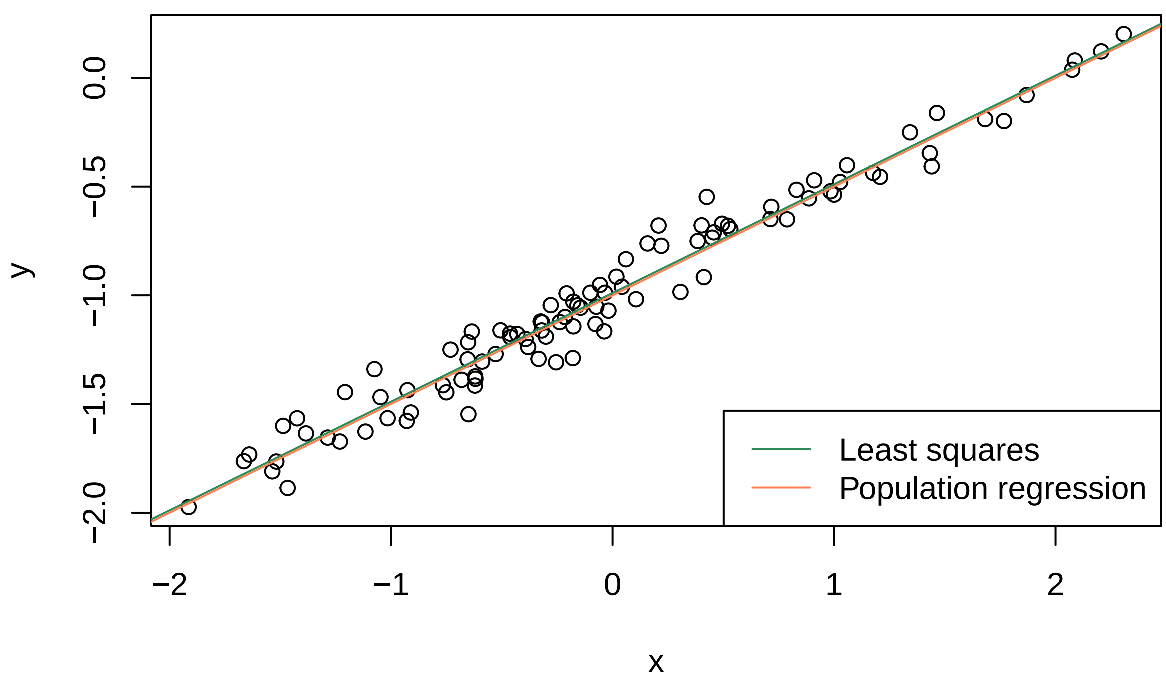
```
set.seed(1)
eps <- rnorm(100, sd = 0.1)
x <- rnorm(100)
y <- -1 + 0.5 * x + eps

# plot(x, y)
tibble(x, y) %>%
    ggplot(aes(x, y)) + geom_point()
```



```r
quiet <- lm(y ~ x)
summary(quiet)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.232416 -0.060361  0.000536  0.058305  0.229316
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.989115   0.009035 -109.48   <2e-16 ***
## x            0.499907   0.009472   52.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09028 on 98 degrees of freedom
## Multiple R-squared:  0.966,  Adjusted R-squared:  0.9657
## F-statistic:  2785 on 1 and 98 DF,  p-value: < 2.2e-16
```

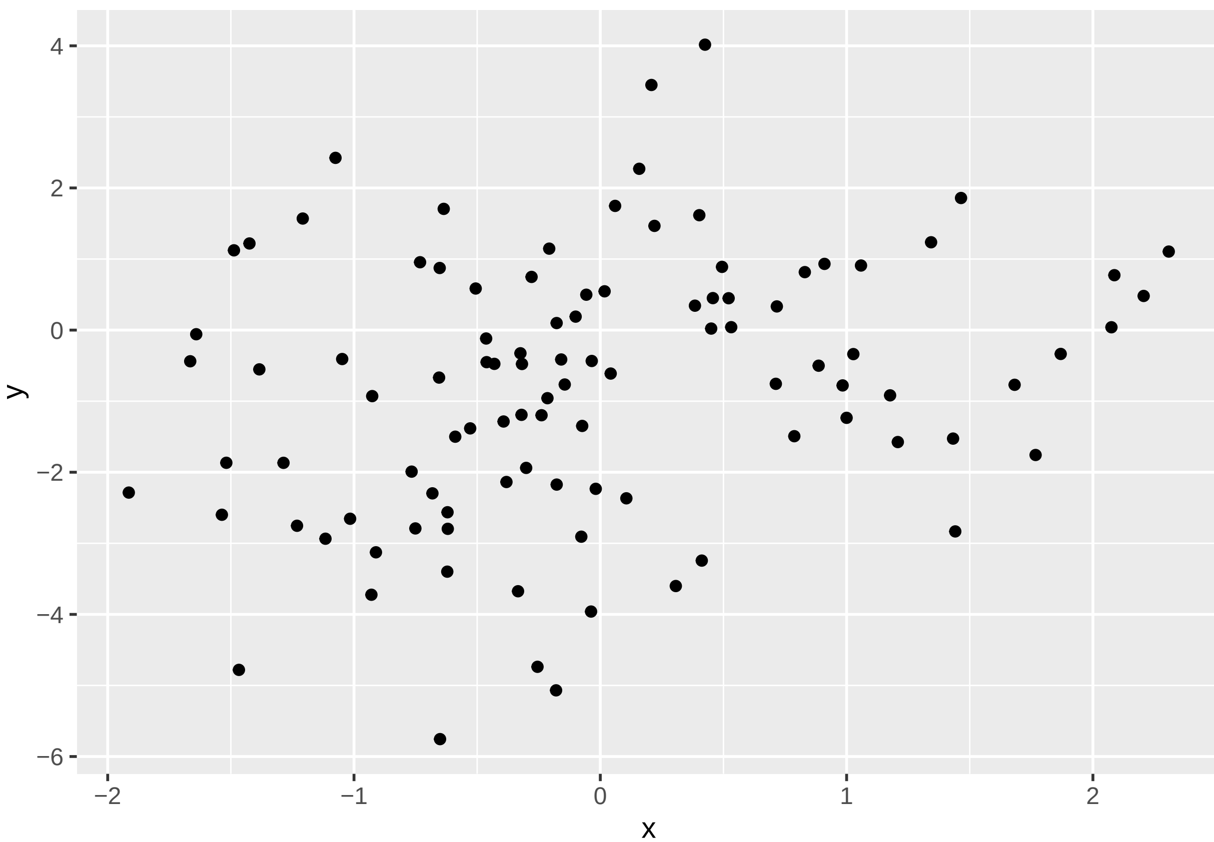
```r
plot(x, y)
abline(quiet, col = "seagreen")
abline(-1, 0.5, col = "coral")
legend("bottomright", c("Least squares", "Population regression"), col = c("seagreen",
    "coral"), lty = c(1, 1))
```



i)

By increasing the noise in the y variable with an increased variance of eps (resulting in an increased error term ϵ), the same model achieves a worse fit. R^2 is lower, RSE is higher The population regression and least squares line now differ more, but are similar enough that we still reject H_0.

```
set.seed(1)
eps <- rnorm(100, sd = 2)
x <- rnorm(100)
y <- -1 + 0.5 * x + eps

# plot(x, y)
tibble(x, y) %>%
    ggplot(aes(x, y)) + geom_point()
```

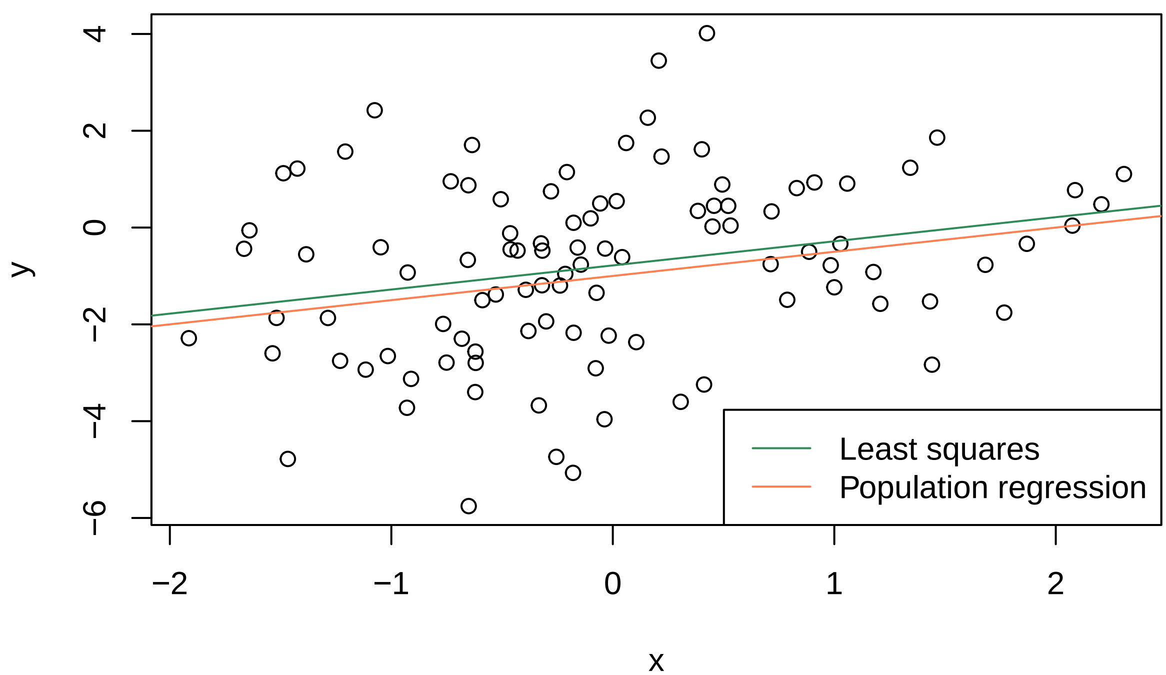



```
noisy <- lm(y ~ x)
summary(noisy)
```

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

```
##      Min       1Q   Median       3Q      Max
## -4.6483 -1.2072  0.0107  1.1661  4.5863
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.7823     0.1807  -4.329 3.61e-05 ***
## x             0.4981     0.1894   2.629  0.00993 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.806 on 98 degrees of freedom
## Multiple R-squared:  0.0659, Adjusted R-squared:  0.05637
## F-statistic: 6.914 on 1 and 98 DF,  p-value: 0.009931
```

```r
plot(x, y)
abline(noisy, col = "seagreen")
abline(-1, 0.5, col = "coral")
legend("bottomright", c("Least squares", "Population regression"), col = c("seagreen",
    "coral"), lty = c(1, 1))
```



j)

As the noise in the data increases, so do the confidence intervals (they widen), as the noise decreases, the confidence interval constricts.

```r
confint(original)
```

```
##                  2.5 %     97.5 %
```

```
## (Intercept) -1.1150804 -0.9226122
## x            0.3925794  0.6063602
```

```
confint(quiet)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0070441 -0.9711855
## x            0.4811096  0.5187039
```

```
confint(noisy)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1408811 -0.4237104
## x            0.1221916  0.8740789
```

1. ISLR 3.7 - 14

a)

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

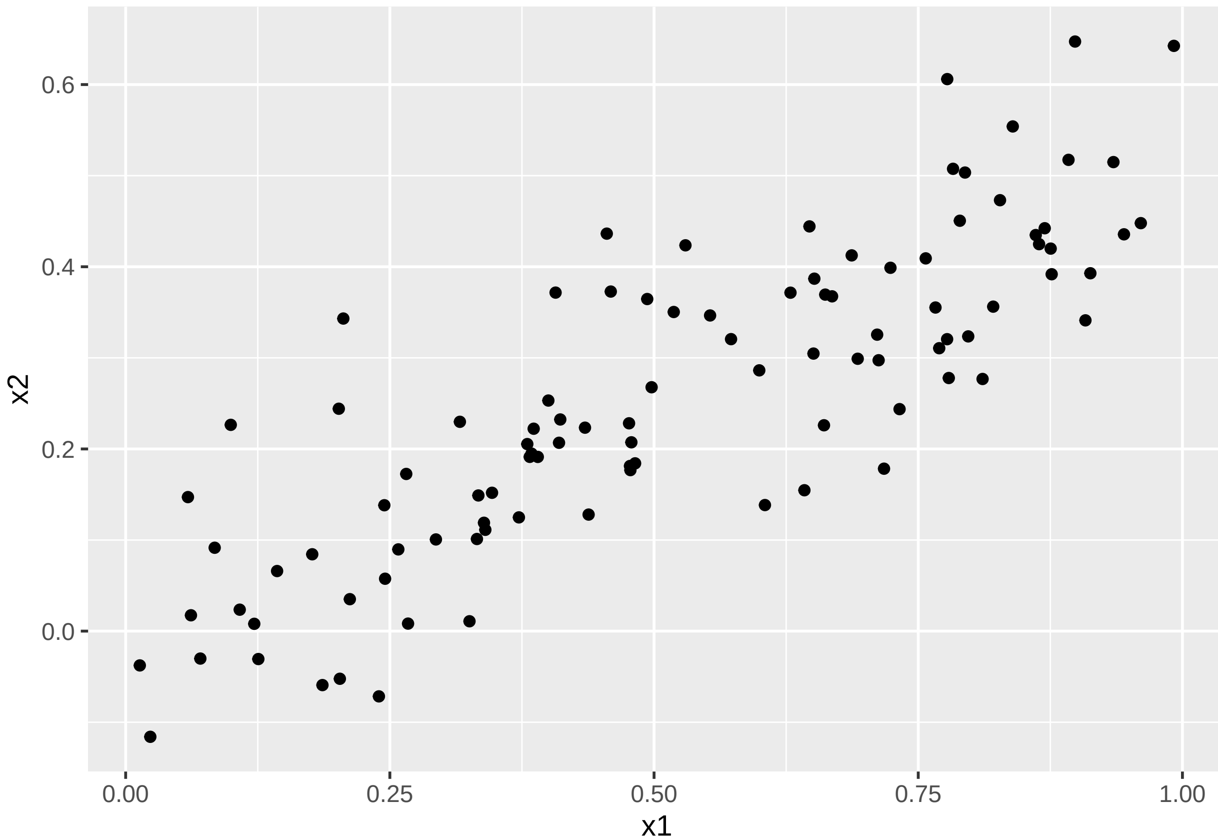
b)

There is a positive correlation between x1 and x2, $r = 0.8351212$.

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

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

```
tibble(x1, x2) %>%
    ggplot(aes(x1, x2)) + geom_point()
```



c)

- $\widehat{\beta}_0$ is close to the true β_0
- $\widehat{\beta}_1$ is (less) close to the true β_1
- $\widehat{\beta}_2$ is not close to the true β_2

We reject $H_0 : \widehat{\beta}_1 = 0$ ($p < 0.05$), but cannot reject $H_0 : \widehat{\beta}_2 = 0$ ($p > 0.05$).

```
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
```

d)

- $\widehat{\beta}_0$ is very close to the true β_0
- $\widehat{\beta}_1$ is very close to the true β_1

We reject $H_0 : \widehat{\beta}_1 = 0$ ($p < 0.05$).

```r
lm_fit = lm(y ~ x1)
summary(lm_fit)
```

```
##
## 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
```

e)

- $\widehat{\beta}_0$ is very close to the true β_0
- $\widehat{\beta}_1$ is very close to the true β_1

We reject $H_0 : \widehat{\beta}_1 = 0$ ($p < 0.05$).

```r
lm_fit = lm(y ~ x2)
summary(lm_fit)
```

```
##
## 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
```

f) No, the results obtained in c to e do not contradict each other. The cause for the discrepancies is that x1 and x2 are strongly correlated, resulting in collinearity between the variables, which makes it harder for the model to tell how either predictor affects the response variable. This causes the model to return less accurate regression coefficients, resulting in higher standard error.

g)

- Model 1: the mismeasured point has high leverage.
- Model 2: the mismeasured point is an outlier, but does not have high leverage.
- Model 3: the mismeasured point has high leverage.

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

summary(lm(y ~ x1 + x2))  # model 1
```

```
##
## 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(lm(y ~ x1))  # model 2
```

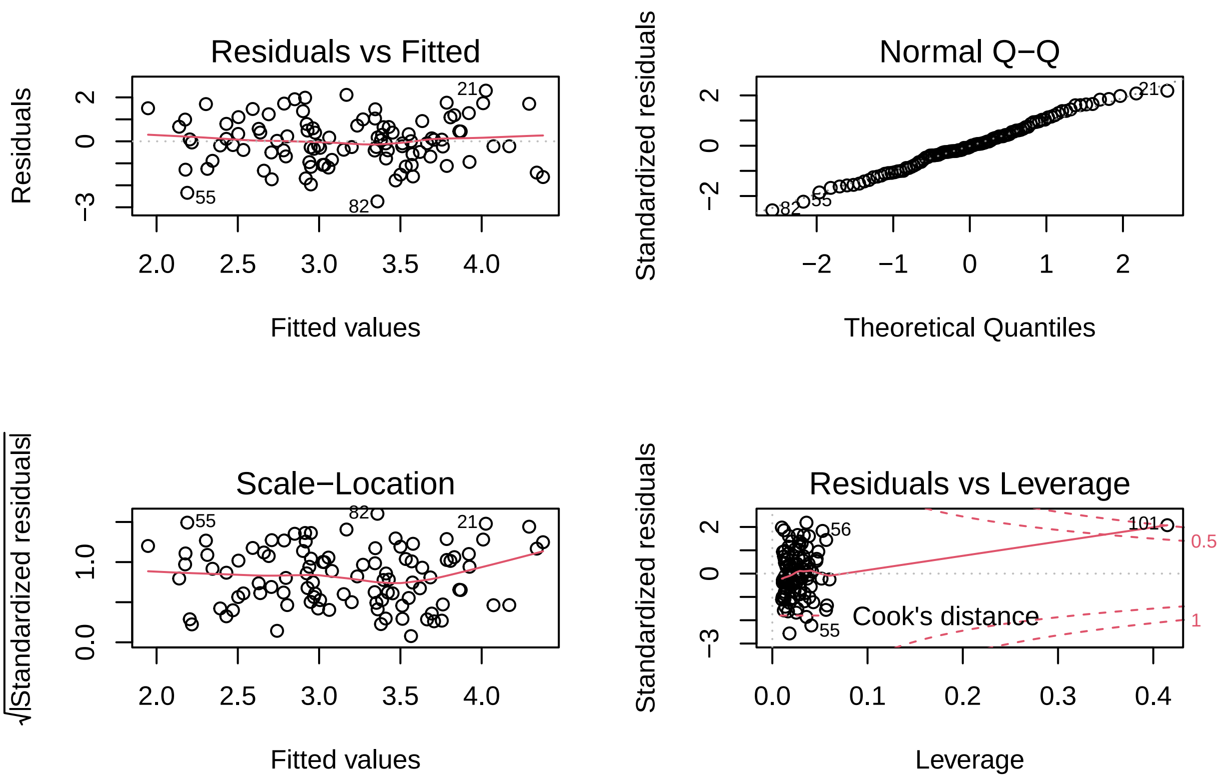
```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1            1.7657     0.4124   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(lm(y ~ x2))  # model 3
```

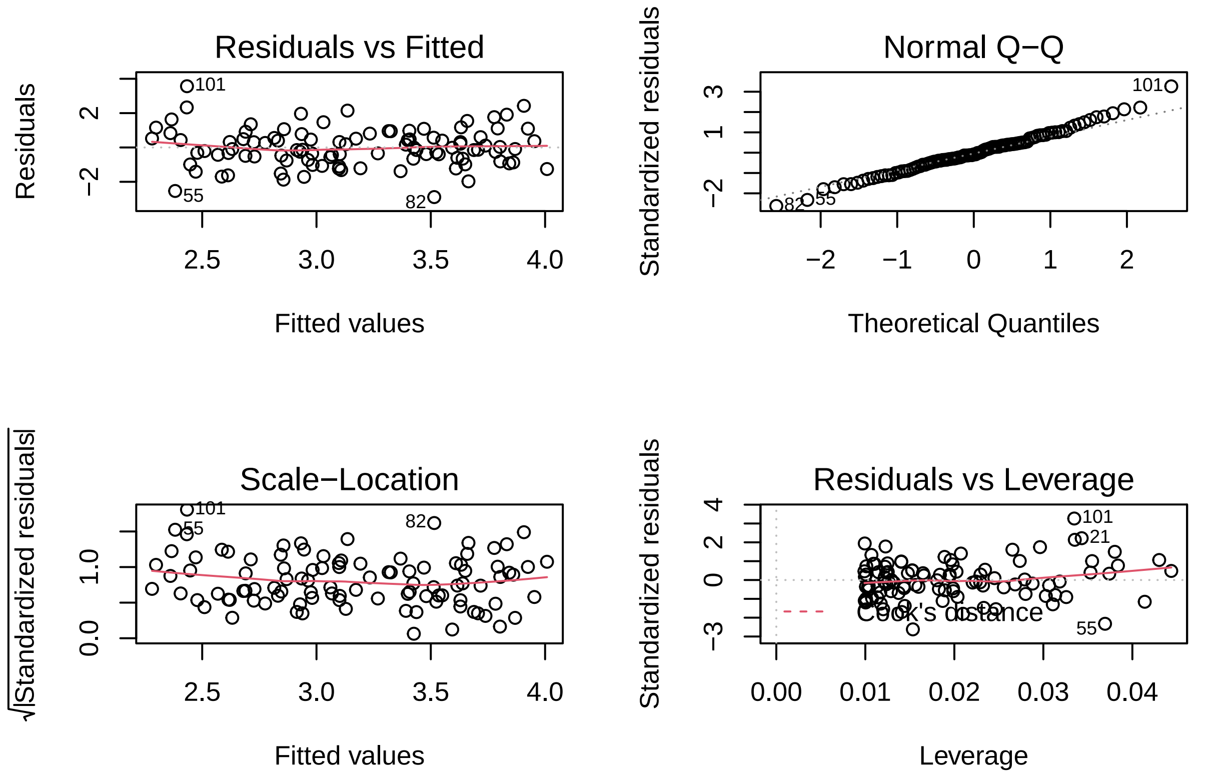
```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (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
```

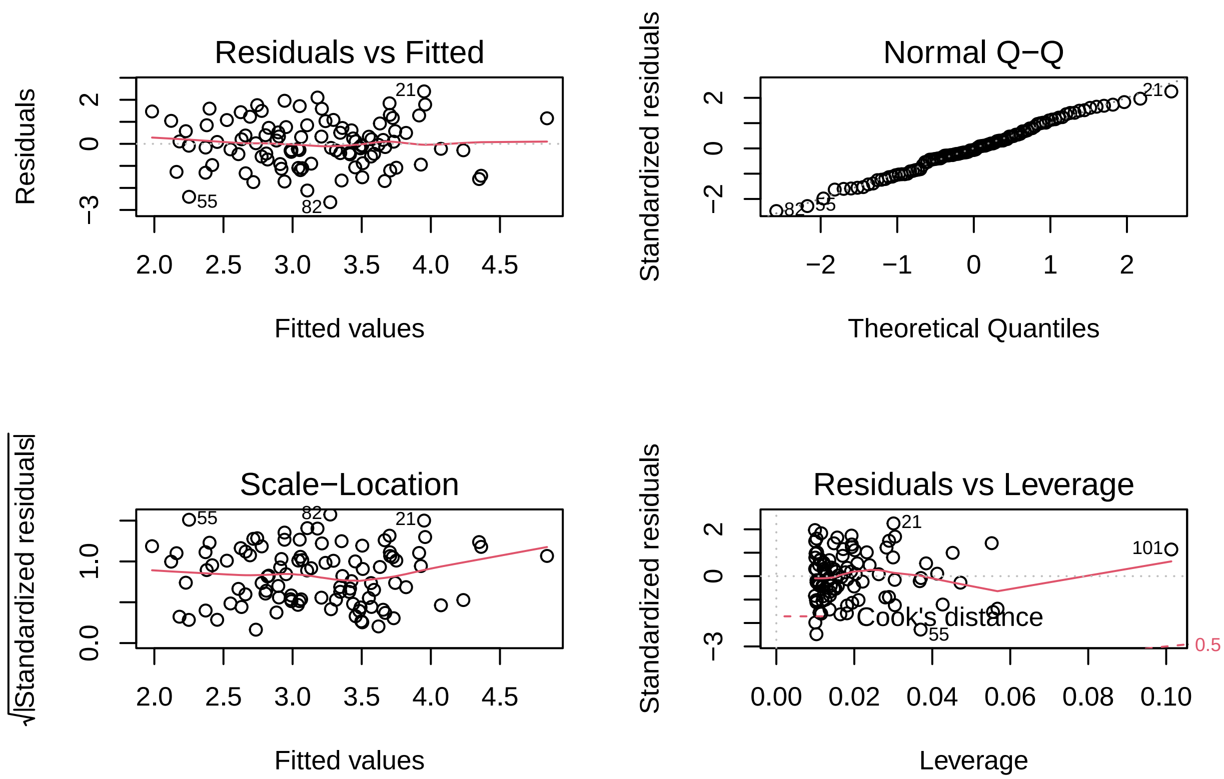
```
par(mfrow = c(2, 2))
plot(lm(y ~ x1 + x2))   # m1
```




```
plot(lm(y ~ x1))   # m2
```




```r
plot(lm(y ~ x2))   # m3
```



Session Info

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ISLR2_1.3-1     forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7
##  [5] purrr_0.3.4     readr_2.1.0     tidyr_1.1.4     tibble_3.1.6
##  [9] ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8        lattice_0.20-45   lubridate_1.8.0   assertthat_0.2.1
##  [5] digest_0.6.29     utf8_1.2.2        R6_2.5.1          cellranger_1.1.0
##  [9] backports_1.4.1   reprex_2.0.1      evaluate_0.14     highr_0.9
## [13] httr_1.4.2        pillar_1.6.4      rlang_0.4.12      readxl_1.3.1
## [17] rstudioapi_0.13   Matrix_1.3-4      rmarkdown_2.11    labeling_0.4.2
## [21] splines_4.1.2     bit_4.0.4         munsell_0.5.0     broom_0.7.11
## [25] compiler_4.1.2    modelr_0.1.8      xfun_0.29         pkgconfig_2.0.3
## [29] mgcv_1.8-38       htmltools_0.5.2   tidyselect_1.1.1 fansi_1.0.0
## [33] crayon_1.4.2      tzdb_0.2.0        dbplyr_2.1.1      withr_2.4.3
## [37] grid_4.1.2        nlme_3.1-153      jsonlite_1.7.2    gtable_0.3.0
## [41] lifecycle_1.0.1   DBI_1.1.1         magrittr_2.0.1    formatR_1.11
## [45] scales_1.1.1      cli_3.1.0         stringi_1.7.6     vroom_1.5.6
## [49] farver_2.1.0      fs_1.5.0          xml2_1.3.2        ellipsis_0.3.2
## [53] generics_0.1.1    vctrs_0.3.8       tools_4.1.2       bit64_4.0.5
## [57] glue_1.6.0        hms_1.1.1         parallel_4.1.2    fastmap_1.1.0
## [61] yaml_2.2.1        colorspace_2.0-2 rvest_1.0.2       knitr_1.37
## [65] haven_2.4.3
```