Linear Regression exercises

Lucy L.

2024-01-31

```
library(MASS)
library(ISLR2)
```

```
##
## Attaching package: 'ISLR2'
```

```
## The following object is masked from 'package:MASS':
##
##     Boston
```

```
library(car)
```

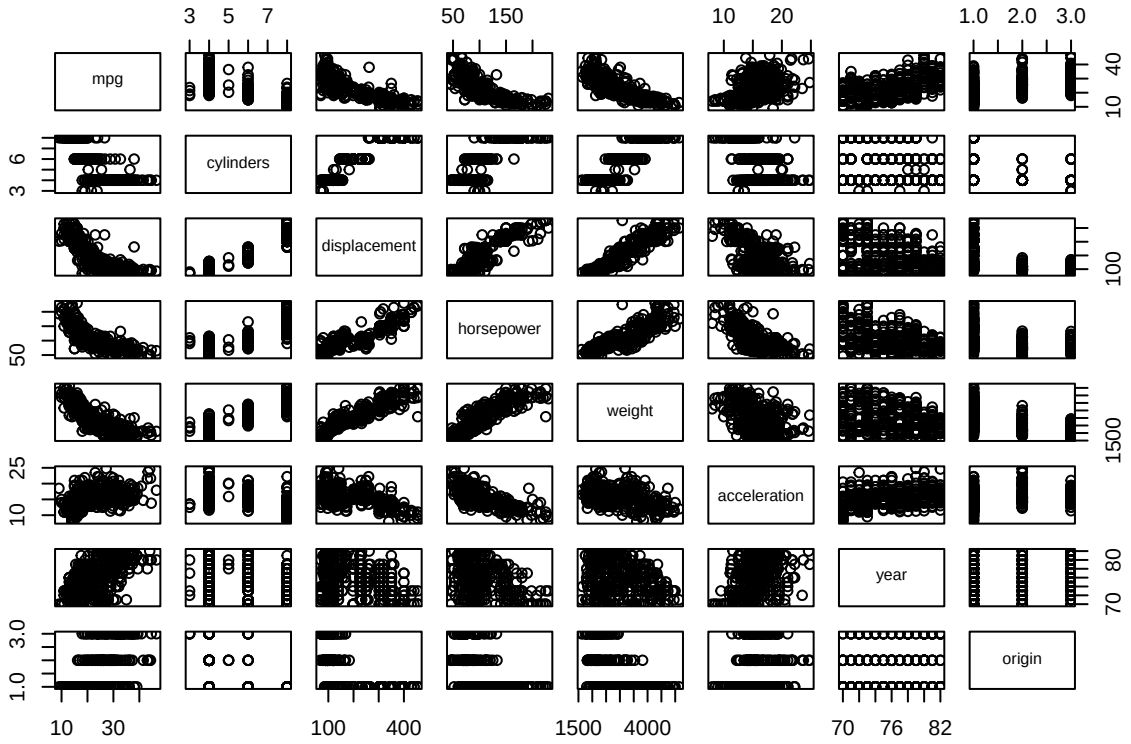
```
## Loading required package: carData
```

Question 9

This question involves the use of multiple linear regression on the `Auto` data set.

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

```
auto = Auto
name_exclude = Auto[,-9]
pairs(name_exclude)
```



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

```
cor(name_exclude)
```

```
##                     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) 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. For instance:

```
model1 = lm(mpg ~ . - name, data = Auto)
summary(model1)
```

```
##
## 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
```

(a) Is there a relationship between the predictors and the response? \

Pretty much yes for all predictors except for `name` based on the correlation matrix.

(b) Which predictors appear to have a statistically signifcant relationship to the response? \

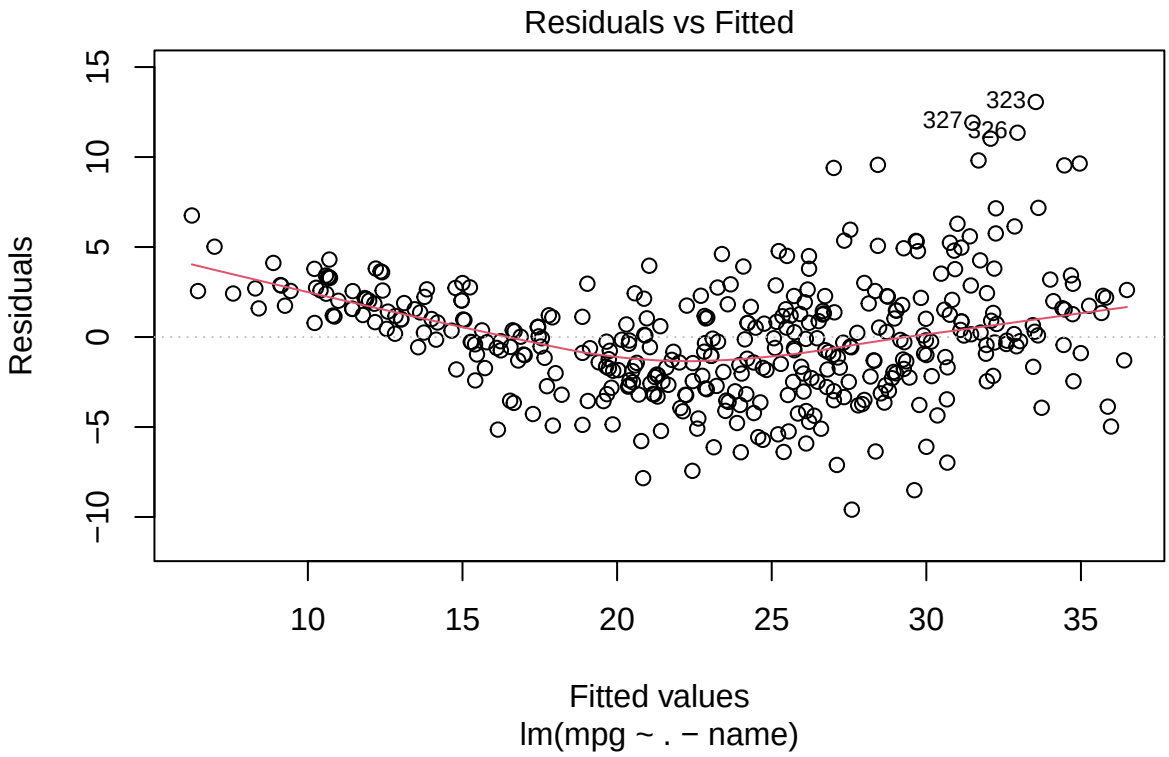
`Displacement`, `Weight`, `year` and `origin` are statistically significant for predicting `mpg`.

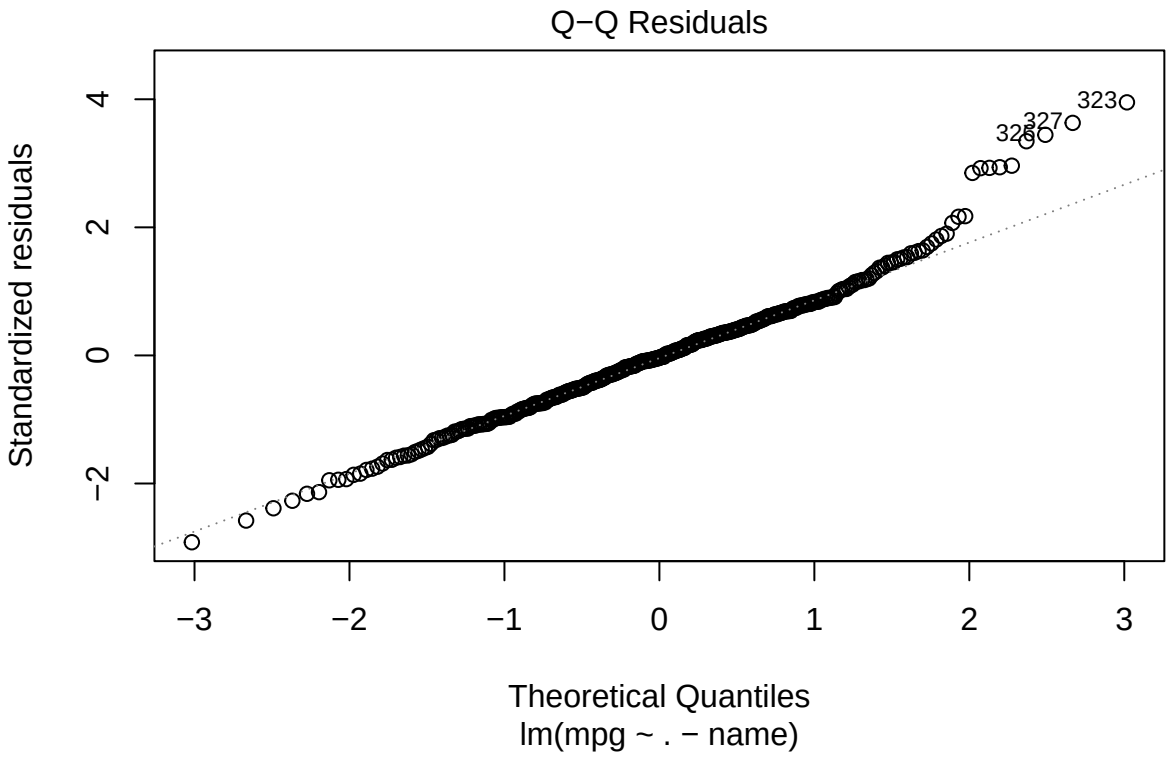
(c) What does the coefficient for the year variable suggest?

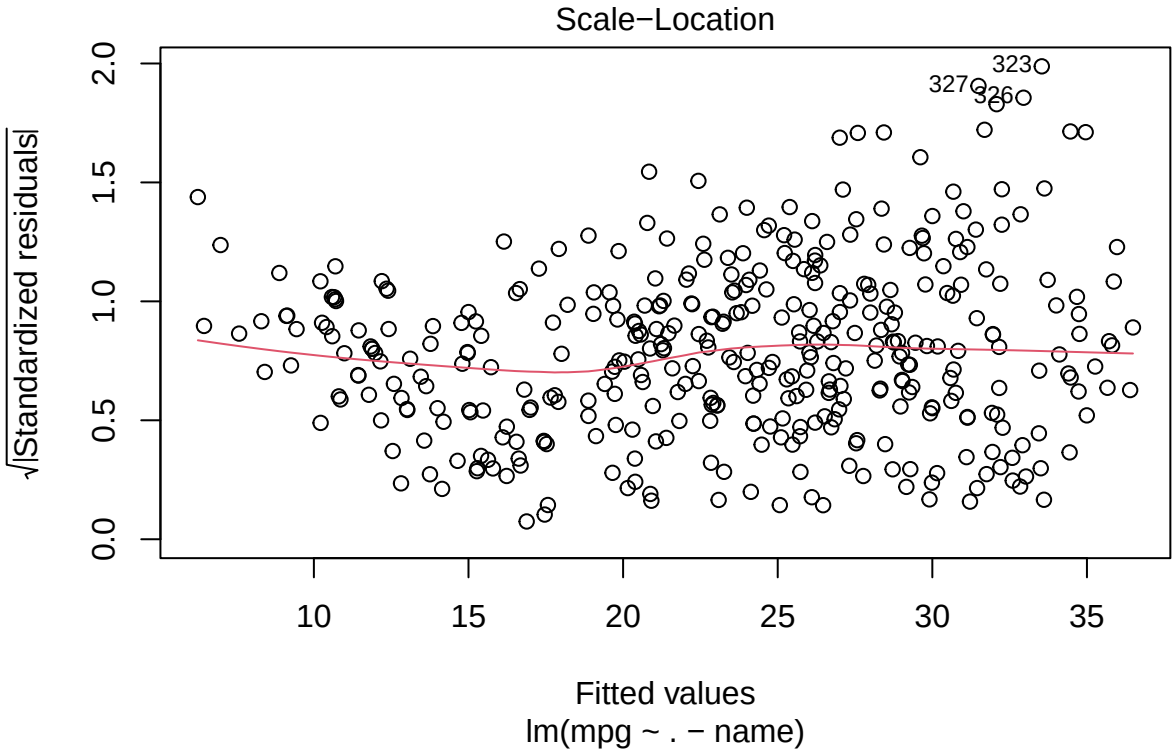
Holding all other variables constant, a one-unit increase in `year` is predicted to change `mpg` by _____ on average.

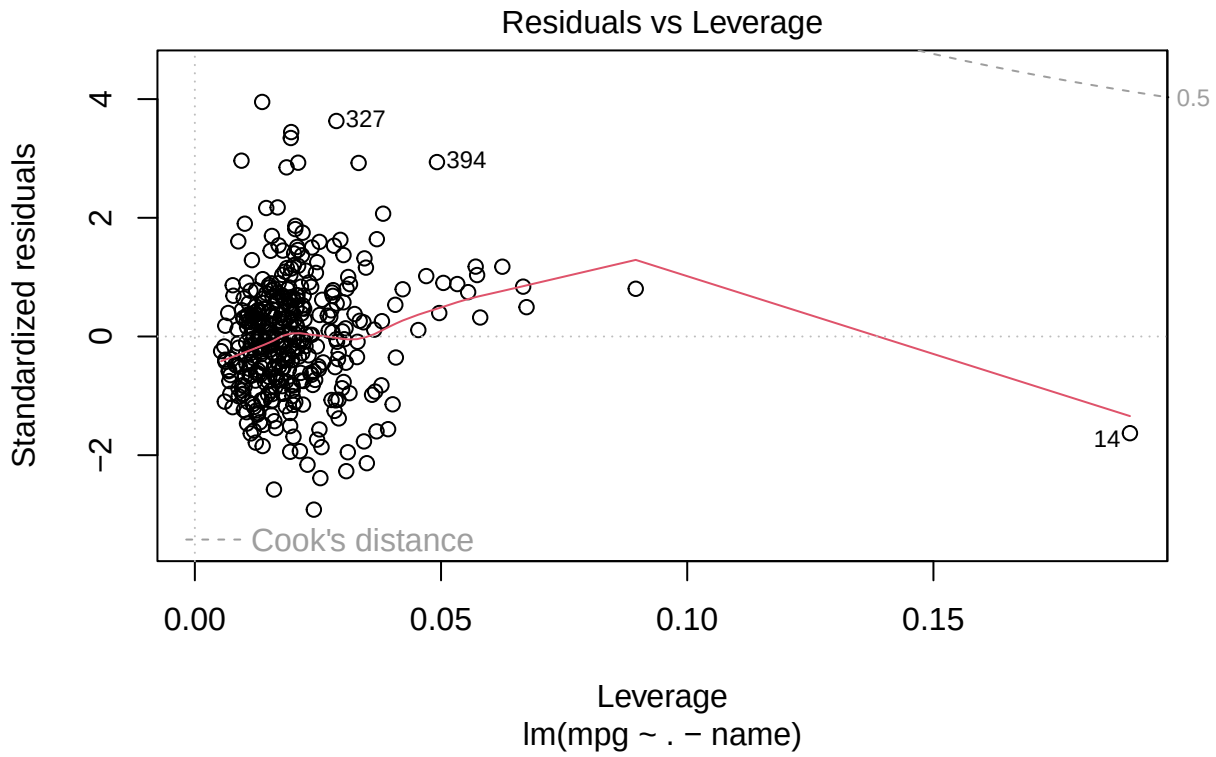
(d) Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

```
plot(model1)
```


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


Q−Q Residuals
Standardized residuals
4
2
0
−2
323
327
326
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(mpg ~ . − name)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
323
327
326
10
15
20
25
30
35
Fitted values
lm(mpg ~ . − name)



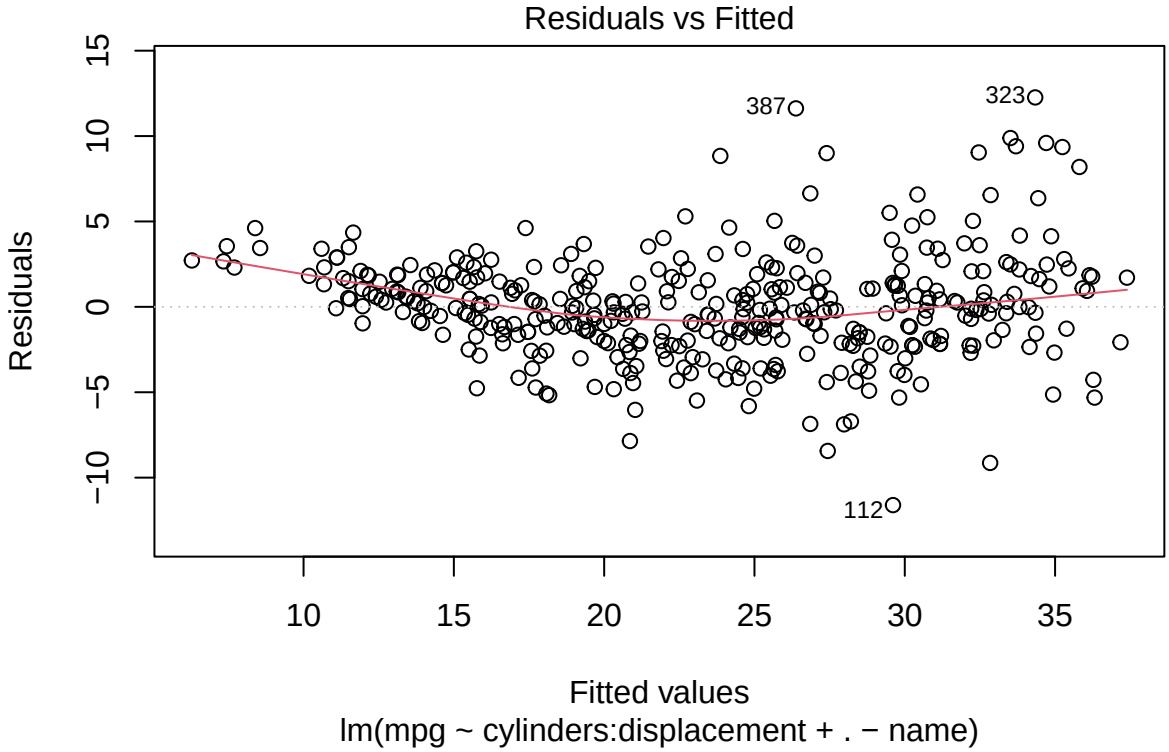
(e) Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

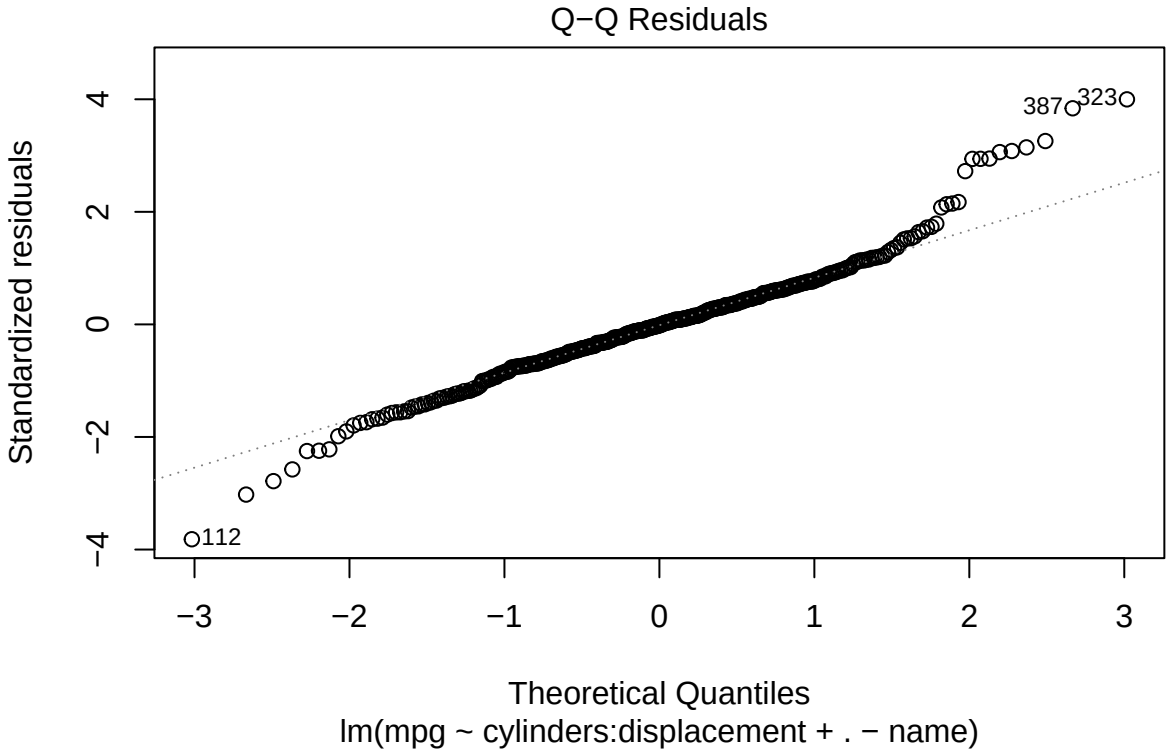
```
model2 = lm(mpg ~ cylinders:displacement + .   - name , data = Auto)
summary(model2)
```

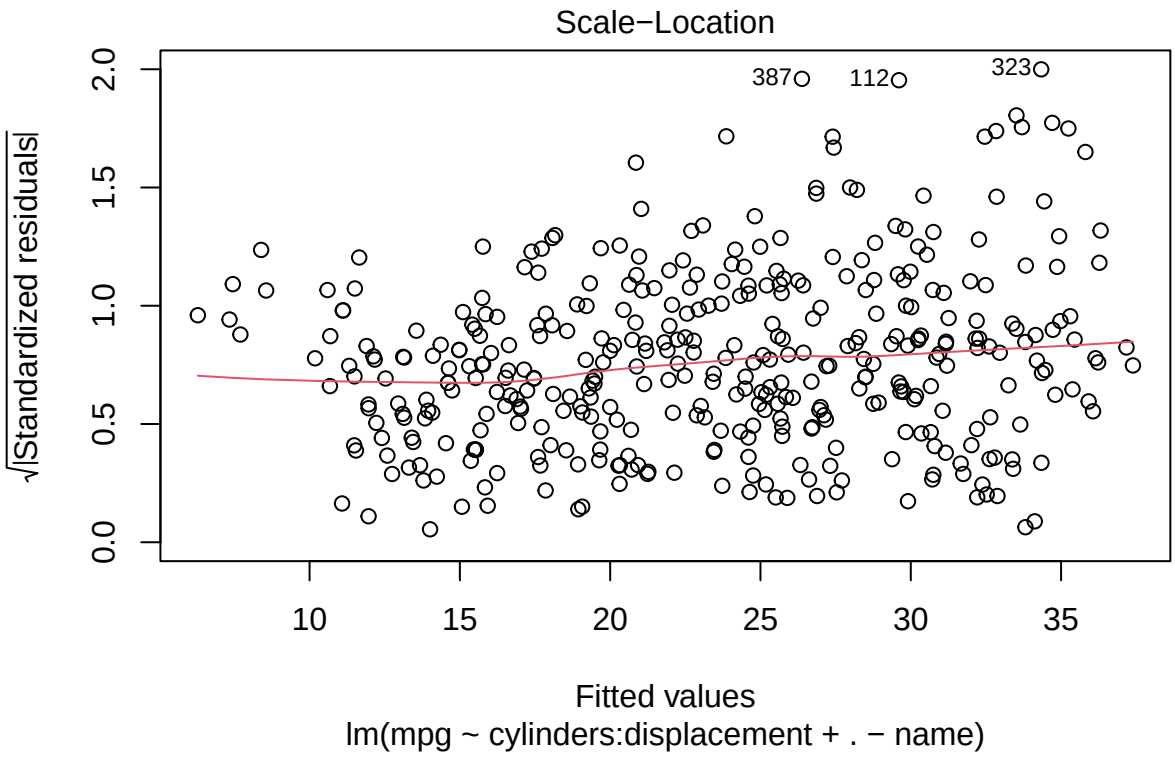
```
##
## Call:
## lm(formula = mpg ~ cylinders:displacement + . - name, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6081  -1.7833  -0.0465   1.6821  12.2617
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -2.7096590  4.6858582  -0.578 0.563426
## cylinders              -2.6962123  0.4094916  -6.584 1.51e-10 ***
## displacement           -0.0774797  0.0141535  -5.474 7.96e-08 ***
## horsepower             -0.0476026  0.0133736  -3.559 0.000418 ***
## weight                 -0.0052339  0.0006253  -8.370 1.10e-15 ***
## acceleration            0.0597997  0.0918038   0.651 0.515188
## year                    0.7594500  0.0473354  16.044  < 2e-16 ***
## origin                  0.7087399  0.2736917   2.590 0.009976 **
## cylinders:displacement  0.0136081  0.0017209   7.907 2.84e-14 ***
```

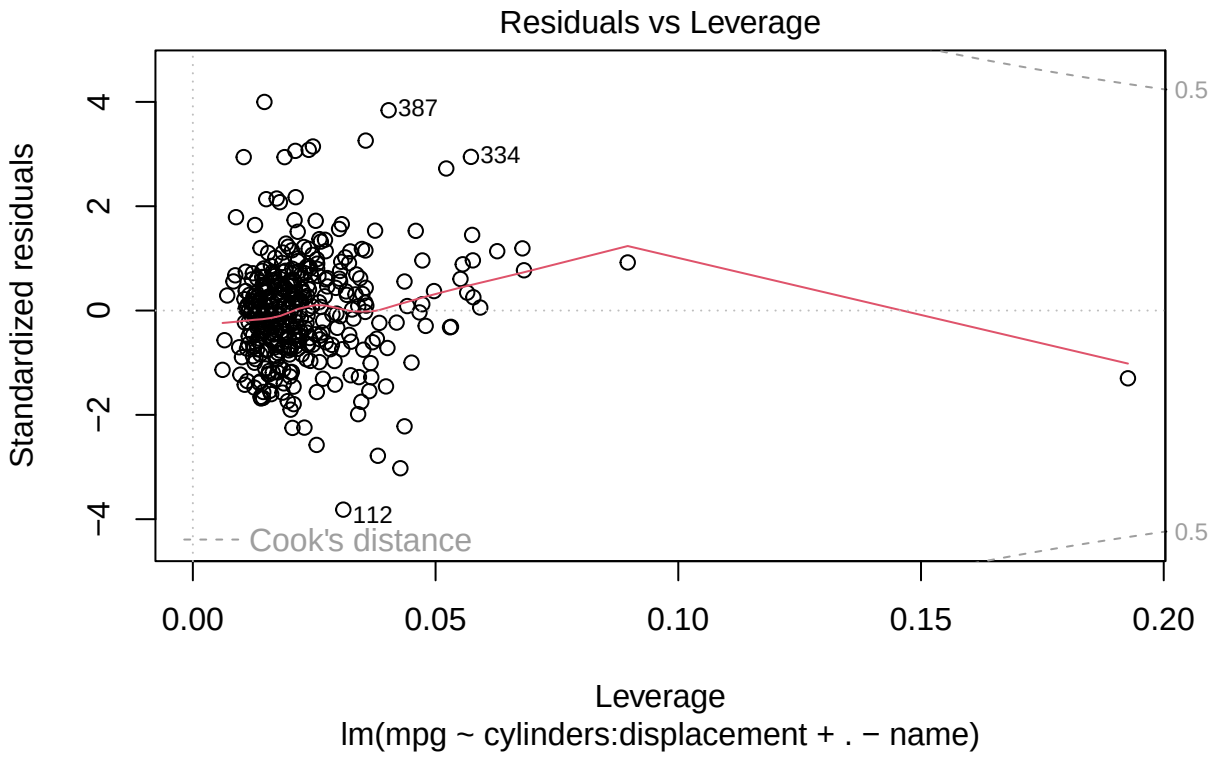
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.089 on 383 degrees of freedom
## Multiple R-squared:  0.8465, Adjusted R-squared:  0.8433
## F-statistic: 264.1 on 8 and 383 DF,  p-value: < 2.2e-16
```

```
plot(model2)
```



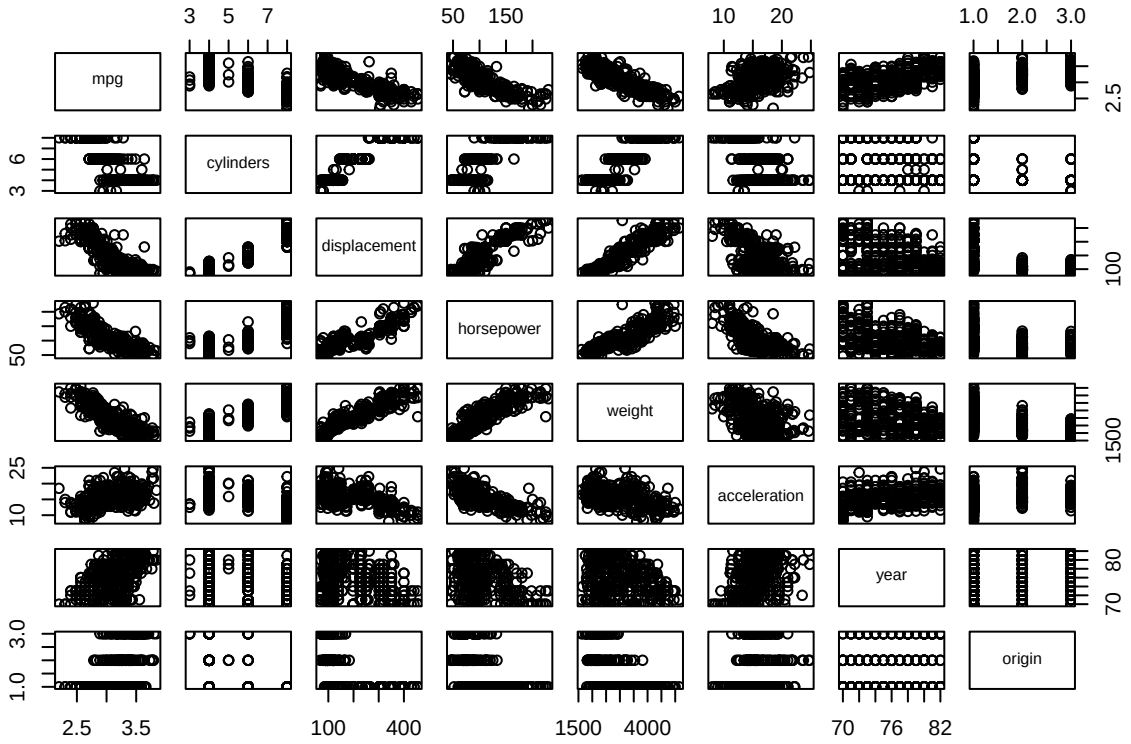

Q−Q Residuals
Standardized residuals
4
2
0
−2
−4
387
323
112
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(mpg ~ cylinders:displacement + . − name)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
387
112
323
10
15
20
25
30
35
Fitted values
lm(mpg ~ cylinders:displacement + . − name)



(f) Try a few different transformations of the variables, such as $log(X), \sqrt{X}, X^2$. Comment on your findings.

```
new_auto = Auto[,-9]
new_auto$mpg = log(new_auto$mpg)
pairs(new_auto)
```

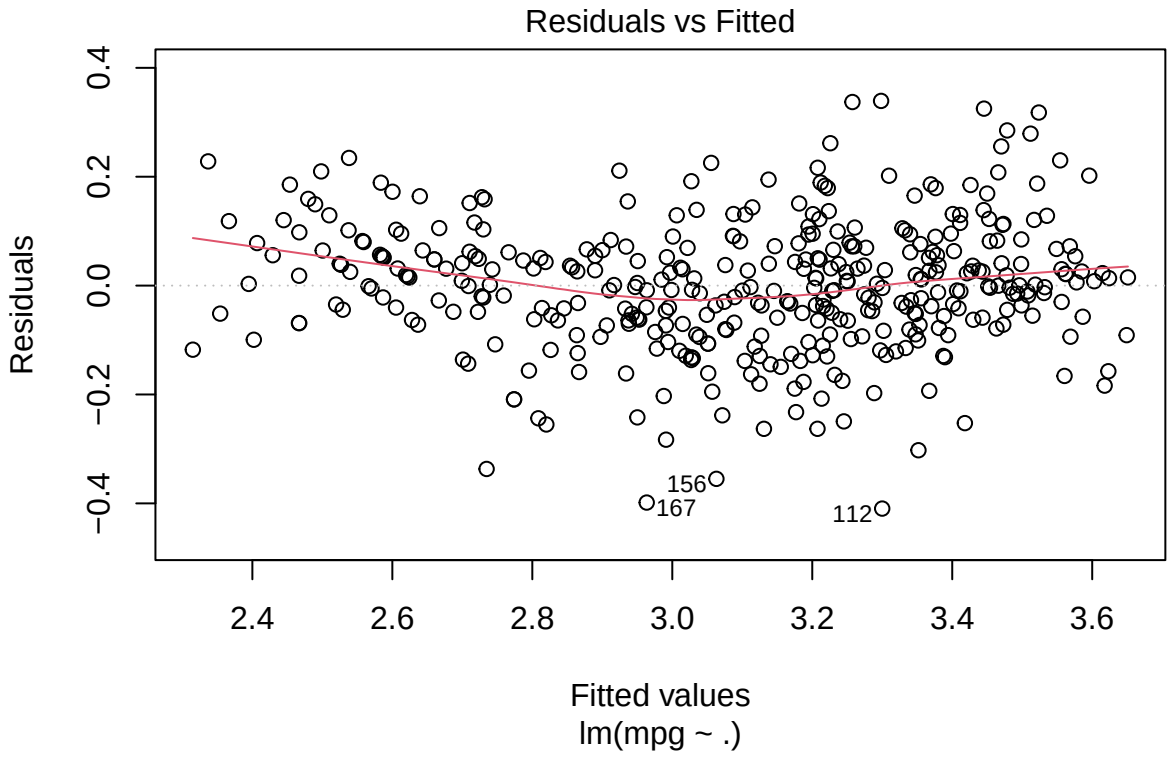


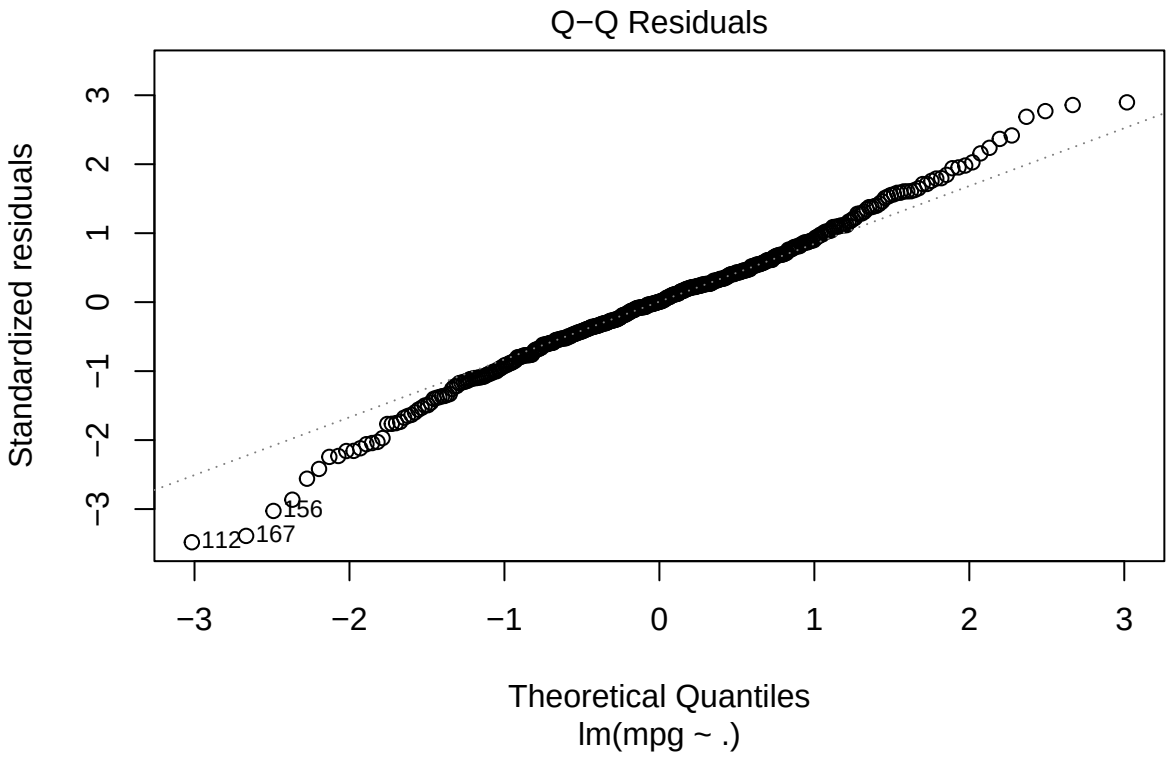

```
model3 = lm(mpg ~ ., data = new_auto)
summary(model3)
```

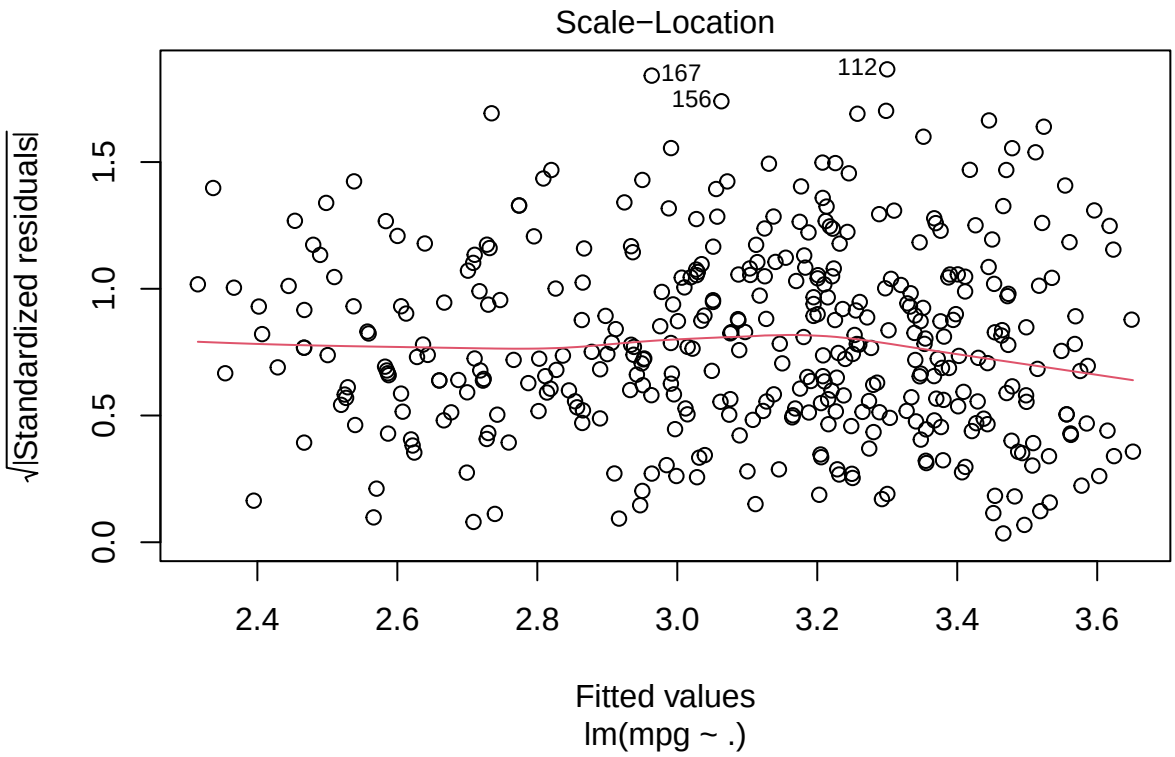
```
##
## Call:
## lm(formula = mpg ~ ., data = new_auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40955 -0.06533  0.00079  0.06785  0.33925
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.751e+00  1.662e-01  10.533  < 2e-16 ***
## cylinders    -2.795e-02  1.157e-02  -2.415  0.01619 *
## displacement  6.362e-04  2.690e-04   2.365  0.01852 *
## horsepower   -1.475e-03  4.935e-04  -2.989  0.00298 **
## weight       -2.551e-04  2.334e-05 -10.931  < 2e-16 ***
## acceleration -1.348e-03  3.538e-03  -0.381  0.70339
## year          2.958e-02  1.824e-03  16.211  < 2e-16 ***
## origin        4.071e-02  9.955e-03   4.089 5.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1191 on 384 degrees of freedom
```

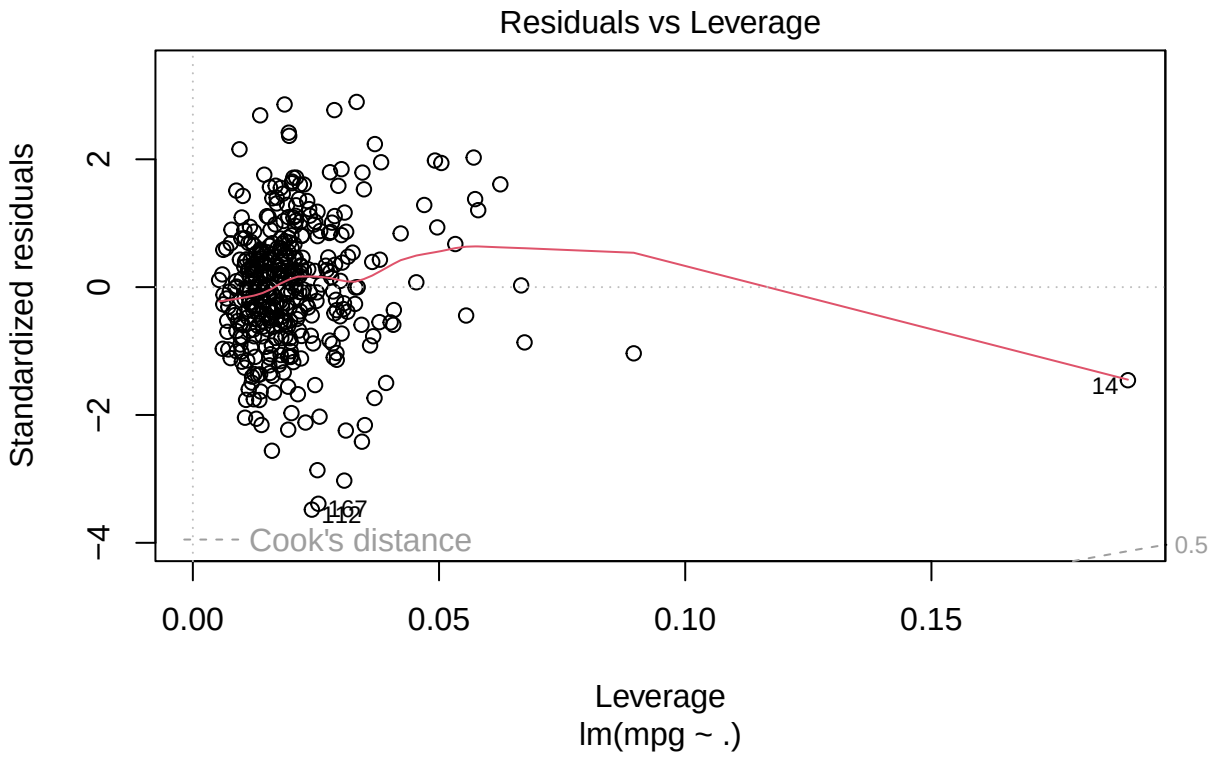
```
## Multiple R-squared:  0.8795, Adjusted R-squared:  0.8773
## F-statistic: 400.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
plot(model3)
```




Q−Q Residuals
Standardized residuals
−3
−2
−1
0
1
2
3
112
167
156
Theoretical Quantiles
lm(mpg ~ .)


Scale−Location
167
156
112
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.4
2.6
2.8
3.0
3.2
3.4
3.6
Fitted values
lm(mpg ~ .)



Question 12

(a) Recall that the coefficient estimate $\hat{\beta}$ for the linear regression of `Y` onto `X` without an intercept is given by (3.38). Under what circumstance is the coefficient estimate for the regression of `X` onto `Y` the same as the coefficient estimate for the regression of `Y` onto `X`?

The coefficient estimate for the regression of X onto Y is the same as the coefficient estimate for the regression of Y onto X when the two variables (X and Y) are inversely related and the slope of the regression line is the same for both cases.

(b) Generate an example in R with n = 100 observations in which the coefficient estimate for the regression of `X` onto `Y` is different from the coefficient estimate for the regression of `Y` onto `X`.

```
set.seed(1)

# Generate random data
n <- 100
X <- rnorm(n)
Y <- -X + rnorm(n)

# Fit regression models
model_X_on_Y <- lm(X ~ Y)
model_Y_on_X <- lm(Y ~ X)

summary(model_X_on_Y)
```

```
##
## Call:
## lm(formula = X ~ Y)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.53746 -0.45396  0.05084  0.44973  1.43654
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.04025    0.06623   0.608    0.545
## Y           -0.46791    0.05035  -9.293 4.15e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6582 on 98 degrees of freedom
## Multiple R-squared:  0.4684, Adjusted R-squared:  0.463
## F-statistic: 86.35 on 1 and 98 DF,  p-value: 4.154e-15
```

```
summary(model_Y_on_X)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8768 -0.6138 -0.1395  0.5394  2.3462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03769    0.09699  -0.389    0.698
## X           -1.00106    0.10773  -9.293 4.15e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9628 on 98 degrees of freedom
## Multiple R-squared:  0.4684, Adjusted R-squared:  0.463
## F-statistic: 86.35 on 1 and 98 DF,  p-value: 4.154e-15
```

(c) Generate an example in R with n = 100 observations in which the coefficient estimate for the regression of `X` onto `Y` is the same as the coefficient estimate for the regression of `Y` onto `X`.

```
# Generate random data
n <- 100
X <- rnorm(n)
Y <- -X + 0.03*rnorm(n)

# Fit regression models
model_X_on_Y <- lm(X ~ Y)
model_Y_on_X <- lm(Y ~ X)

# Display coefficient estimates
```

```r
coeff_X_on_Y <- coef(model_X_on_Y)[2]
coeff_Y_on_X <- coef(model_Y_on_X)[2]

cat("Coefficient estimate for X onto Y:", coeff_X_on_Y, "\n")
```

```
## Coefficient estimate for X onto Y: -1.002372
```

```r
cat("Coefficient estimate for Y onto X:", coeff_Y_on_X, "\n")
```

```
## Coefficient estimate for Y onto X: -0.9968134
```

Question 13

In this exercise you will create some simulated data and will 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`.

```r
x <- rnorm(100, mean = 0, sd = 1)
```

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

```r
eps <- rnorm(100, mean = 0, sd = sqrt(0.25))
```

(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?

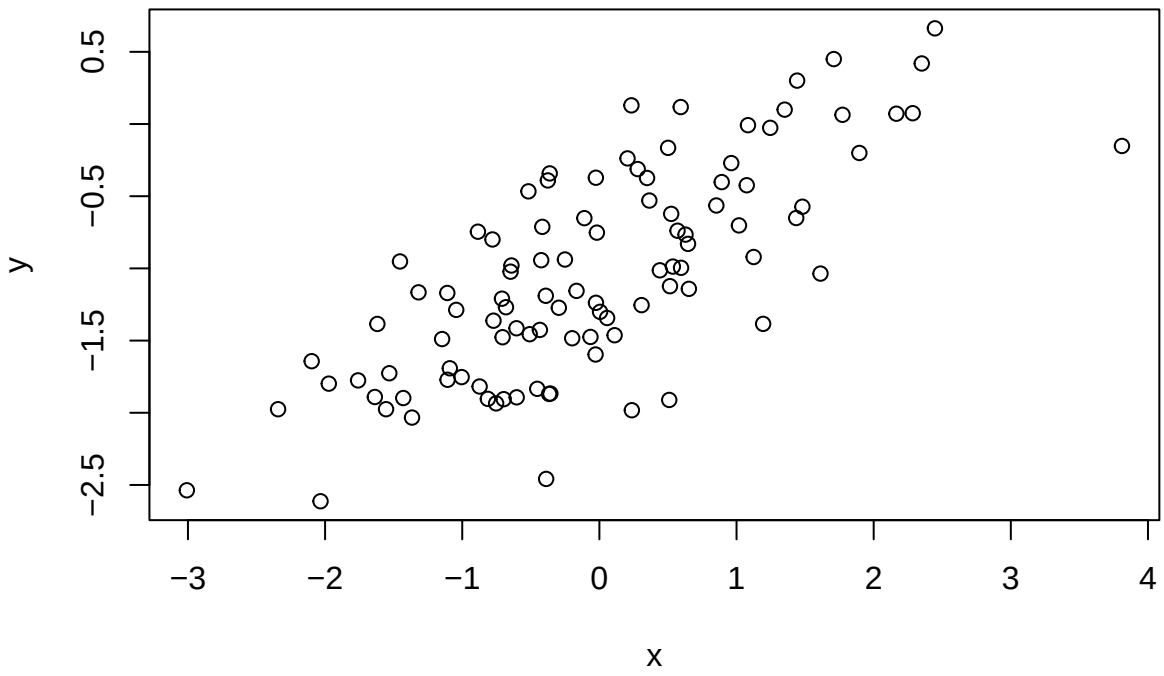
```r
y = -1 + (0.5 * x) + eps
length(y)
```

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

$\beta_0 = -1$ and $\beta_1 = 0.5$.

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

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



(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?

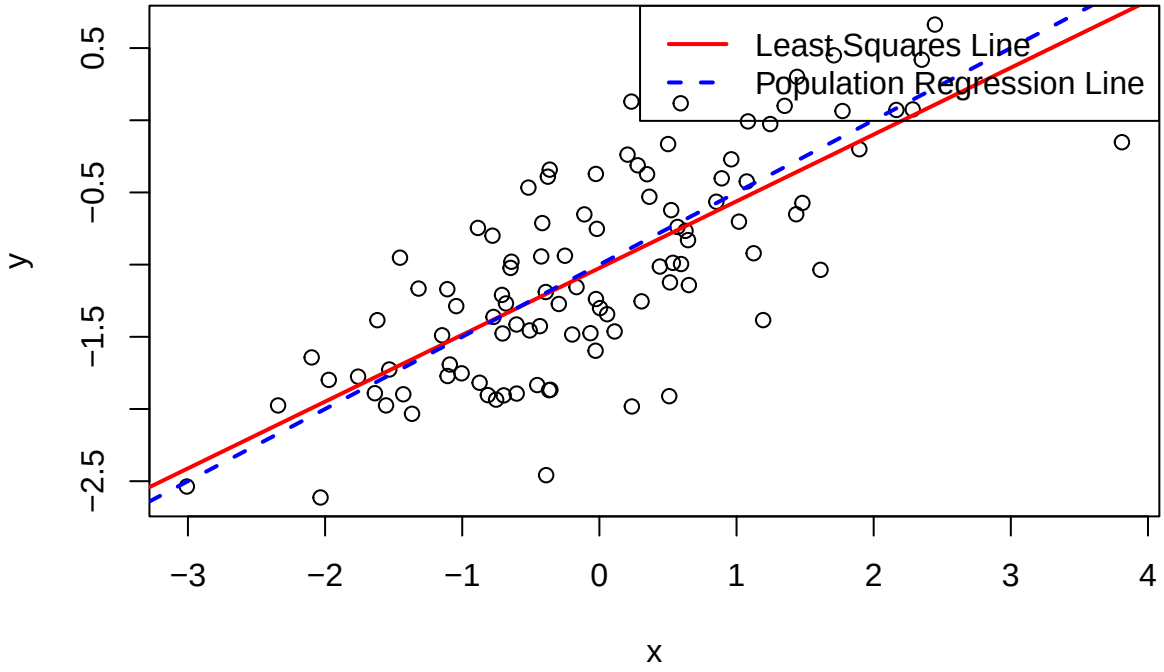
```
model = lm(y ~ x)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.25507 -0.30275  0.01032  0.35241  1.04490
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.02373    0.04838  -21.16   <2e-16 ***
## x            0.46253    0.04155   11.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4835 on 98 degrees of freedom
## Multiple R-squared:  0.5584, Adjusted R-squared:  0.5539
## F-statistic: 123.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

They are approximate to each other.

(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()` command to create an appropriate legend.

```
plot(x, y)
abline(model, col = "red", lwd = 2)
abline(a = -1, b = 0.5, col = "blue", lty = 2, lwd = 2)
legend("topright", legend = c("Least Squares Line",
                              "Population Regression Line"),
       col = c("red", "blue"), lty = c(1, 2), 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.

```
model2 = lm(y ~ x + I(x^2))
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.25507 -0.30275  0.01032  0.35241  1.04490
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.02373    0.04838  -21.16   <2e-16 ***
## x            0.46253    0.04155   11.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4835 on 98 degrees of freedom
## Multiple R-squared:  0.5584, Adjusted R-squared:  0.5539
## F-statistic: 123.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26806 -0.31002  0.00414  0.37569  1.02831
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00741    0.05834 -17.267   <2e-16 ***
## x            0.46647    0.04244  10.992   <2e-16 ***
## I(x^2)      -0.01192    0.02362  -0.505    0.615
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4854 on 97 degrees of freedom
## Multiple R-squared:  0.5595, Adjusted R-squared:  0.5505
## F-statistic: 61.61 on 2 and 97 DF,  p-value: < 2.2e-16
```

Not really. If we use R^2 as a measure for model fitness, it only slightly increases. The quadratic term itself is not significant in predicting `y`.

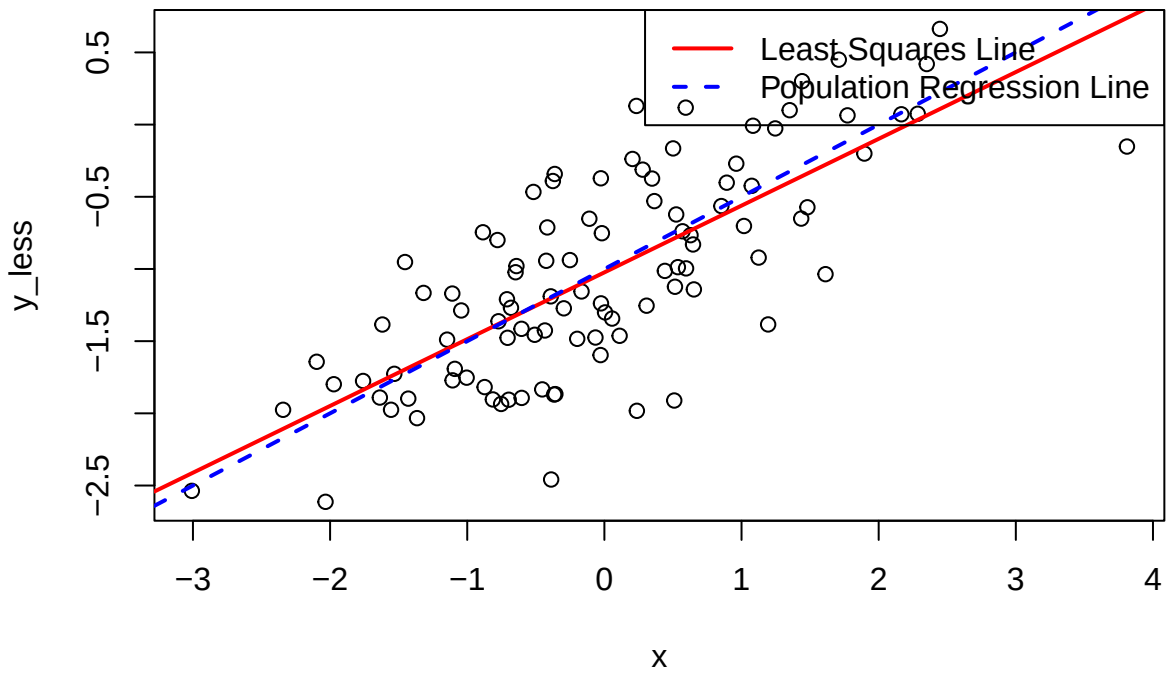
(h) Repeat (a)–(f) after modifying the data generation process in such a way that there is less noise in the data. The model (3.39) should remain the same. You can do this by decreasing the variance of the normal distribution used to generate the error term ϵ in (b). Describe your results.

```
eps_less <- rnorm(100, mean = 0, sd = 0.1)
y_less = -1 + (0.5 * x) + eps

model = lm(y_less ~ x)
summary(model)
```

```
##
## Call:
## lm(formula = y_less ~ x)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.25507 -0.30275  0.01032  0.35241  1.04490
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.02373    0.04838  -21.16   <2e-16 ***
## x            0.46253    0.04155   11.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4835 on 98 degrees of freedom
## Multiple R-squared:  0.5584, Adjusted R-squared:  0.5539
## F-statistic: 123.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
plot(x, y_less)
abline(model, col = "red", lwd = 2)
abline(a = -1, b = 0.5, col = "blue", lty = 2, lwd = 2)
legend("topright", legend = c("Least Squares Line",
                              "Population Regression Line"),
       col = c("red", "blue"), lty = c(1, 2), lwd = 2)
```



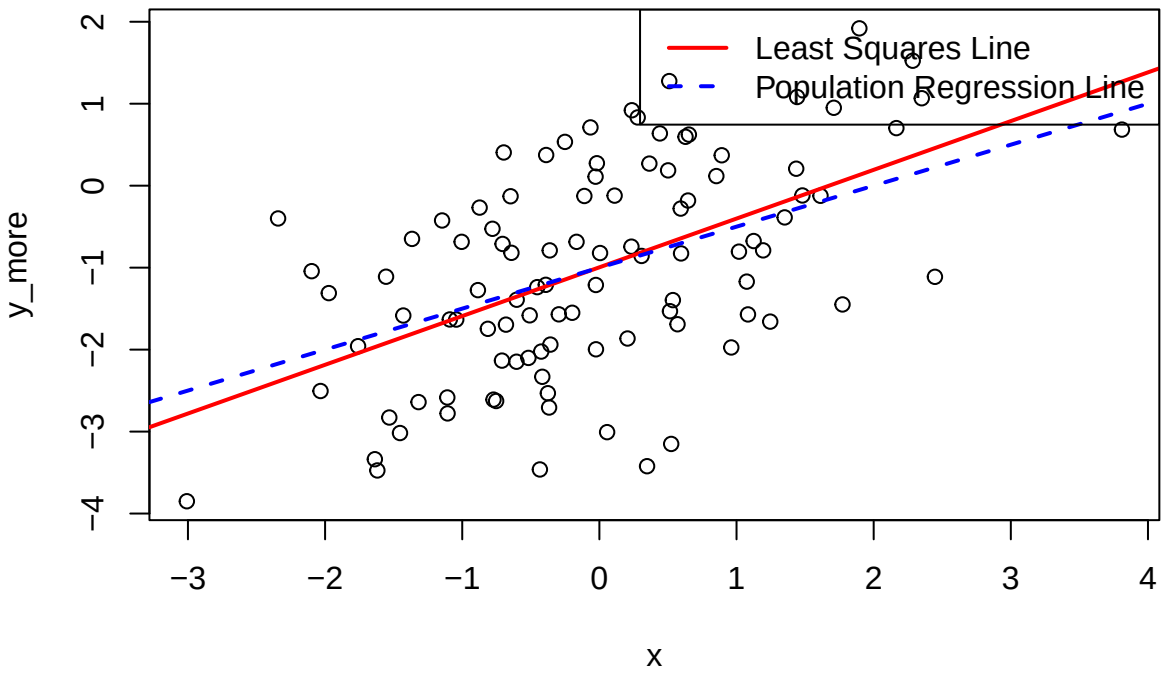
The estimated population regression line is almost indistinguishable from the actual population line.

(i) Repeat (a)–(f) after modifying the data generation process in such a way that there is more noise in the data. The model (3.39) should remain the same. You can do this by increasing the variance of the normal distribution used to generate the error term ϵ in (b). Describe your results.

```r
eps <- rnorm(100, mean = 0, sd = 1)
y_more = -1 + (0.5 * x) + eps

model = lm(y_more ~ x)
summary(model)
```

```
##
## Call:
## lm(formula = y_more ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.63295 -0.84596 -0.01012  0.91369  1.98919
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.99604    0.10979  -9.072 1.25e-14 ***
## x            0.59459    0.09429   6.306 8.25e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.097 on 98 degrees of freedom
## Multiple R-squared:  0.2887, Adjusted R-squared:  0.2814
## F-statistic: 39.77 on 1 and 98 DF,  p-value: 8.254e-09
```

```r
plot(x, y_more)
abline(model, col = "red", lwd = 2)
abline(a = -1, b = 0.5, col = "blue", lty = 2, lwd = 2)
legend("topright", legend = c("Least Squares Line",
                              "Population Regression Line"),
       col = c("red", "blue"), lty = c(1, 2), lwd = 2)
```



Towards higher (absolute) values of X, the least squares line estimates increasingly differently from the population regression line.

(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.

```
# Fit linear models for the original, noisier, and less noisy data sets
x <- rnorm(100, mean = 0, sd = 1)
y = -1 + (0.5 * x) + eps
model_original <- lm(y ~ x)
model_more <- lm(y_more ~ x)
model_less_noisy <- lm(y_less ~ x)

# Calculate confidence intervals
confint_original <- confint(model_original)
confint_noisier <- confint(model_more)
confint_less_noisy <- confint(model_less_noisy)

print(confint_original)
```

```
##                   2.5 %     97.5 %
## (Intercept) -1.2181417 -0.7806519
## x            0.2634561  0.6674349
```

```
print(confint_noisier)
```

```
##                   2.5 %     97.5 %
## (Intercept) -1.2775876 -0.7612282
## x           -0.2289065  0.2479005
```

```
print(confint_less_noisy)
```

```
##                    2.5 %     97.5 %
## (Intercept) -1.18529971 -0.9006095
## x           -0.01952347  0.2433598
```

The conf. intervals for the noisier data are wider than the original, while the intervals are relatively shorter for the less noisy data set.

Question 14

(a) Perform the following commands in R. 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?

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

$$\hat{y} = 2 + 2X_1 + 0.3X_2 + \epsilon$$

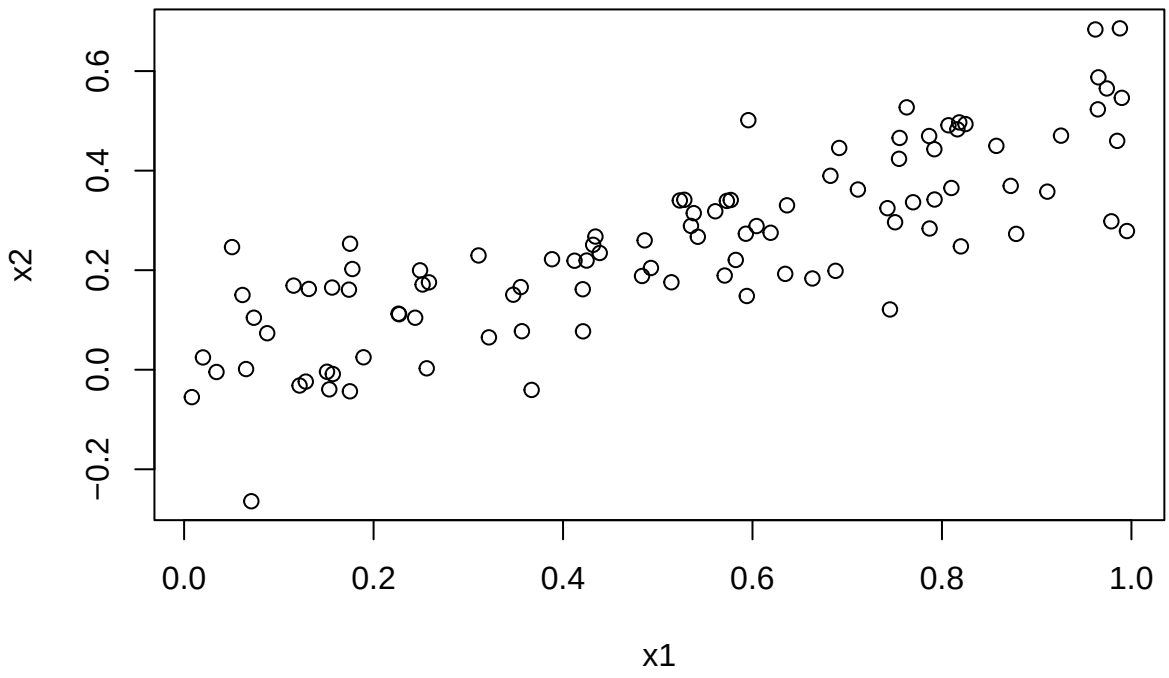
$\beta_1 = 2, \beta_2 = 0.3$

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

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

```
## [1] 0.8216854
```

```
plot(x1, 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 $\hat{\beta}_2$? 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 $H_0 : \beta_2 = 0$?

```
model_a = lm(y ~ x1 + x2)
summary(model_a)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.36481 -0.68580 -0.01947  0.75561  2.50537
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9327     0.2062   9.372 3.04e-15 ***
## x1            2.1194     0.6117   3.465 0.000791 ***
## x2           -0.0487     1.0068  -0.048 0.961520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.017 on 97 degrees of freedom
## Multiple R-squared:  0.2713, Adjusted R-squared:  0.2563
```

```
## F-statistic: 18.06 on 2 and 97 DF,  p-value: 2.15e-07
```

$\hat{\beta}_0 = 2.038, \hat{\beta}_1 = 2.187, \hat{\beta}_2 = -0.619$

The coefficient for $\hat{\beta}_2$ is twice as big in magnitude and has a different sign.

We can reject the first null hypothesis but not the second.

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

```
model_b = lm(y ~ x1)
summary(model_b)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.36882 -0.68825 -0.02031  0.75616  2.50397
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9331     0.2050   9.427 2.12e-15 ***
## x1            2.0951     0.3468   6.041 2.77e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.012 on 98 degrees of freedom
## Multiple R-squared:  0.2713, Adjusted R-squared:  0.2639
## F-statistic: 36.49 on 1 and 98 DF,  p-value: 2.765e-08
```

The estimates are about the same. We can still reject this null hypothesis.

(e) Now fit a least squares regression to predict `y` using only `x2`. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

```
model_c = lm(y ~ x2)
summary(model_c)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.78314 -0.69768  0.03858  0.74328  2.43719
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3066     0.1854  12.444  < 2e-16 ***
```

```
## x2            2.8177     0.6052   4.656 1.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.073 on 98 degrees of freedom
## Multiple R-squared:  0.1811, Adjusted R-squared:  0.1728
## F-statistic: 21.68 on 1 and 98 DF,  p-value: 1.013e-05
```

The estimate for $\hat{\beta}_2$ is still different than originally defined but now we do reject that null hypothesis.

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

They do contradict each other. Individually, `x1` and `x2` each are significant for predicting `y` because alone, they have all the power to predict `y`. Together, the interpretation changes because the two predictors may interact with each other in order to predict `y` together.

(g) Now suppose we obtain one additional observation, which was unfortunately mis-measured. Re-fit the linear models from (c) to (e) using this new data. What effect does this new observation have on the each of the models? In each model, is this observation an outlier? A high-leverage point? Both? Explain your answers.

```
x11 = c(x1[1:99], 0,1)
x22 = c(x2, 0.8)
y = c(y, 6)

model_a = lm(y ~ x11 + x22)
summary(model_a)
```

```
##
## Call:
## lm(formula = y ~ x11 + x22)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.42095 -0.69851 -0.01612  0.73285  2.46054
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9241     0.2040   9.433 2.06e-15 ***
## x11           1.8887     0.5878   3.213  0.00178 **
## x22           0.5634     0.9509   0.592  0.55490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.029 on 98 degrees of freedom
## Multiple R-squared:  0.2924, Adjusted R-squared:  0.2779
## F-statistic: 20.24 on 2 and 98 DF,  p-value: 4.372e-08
```

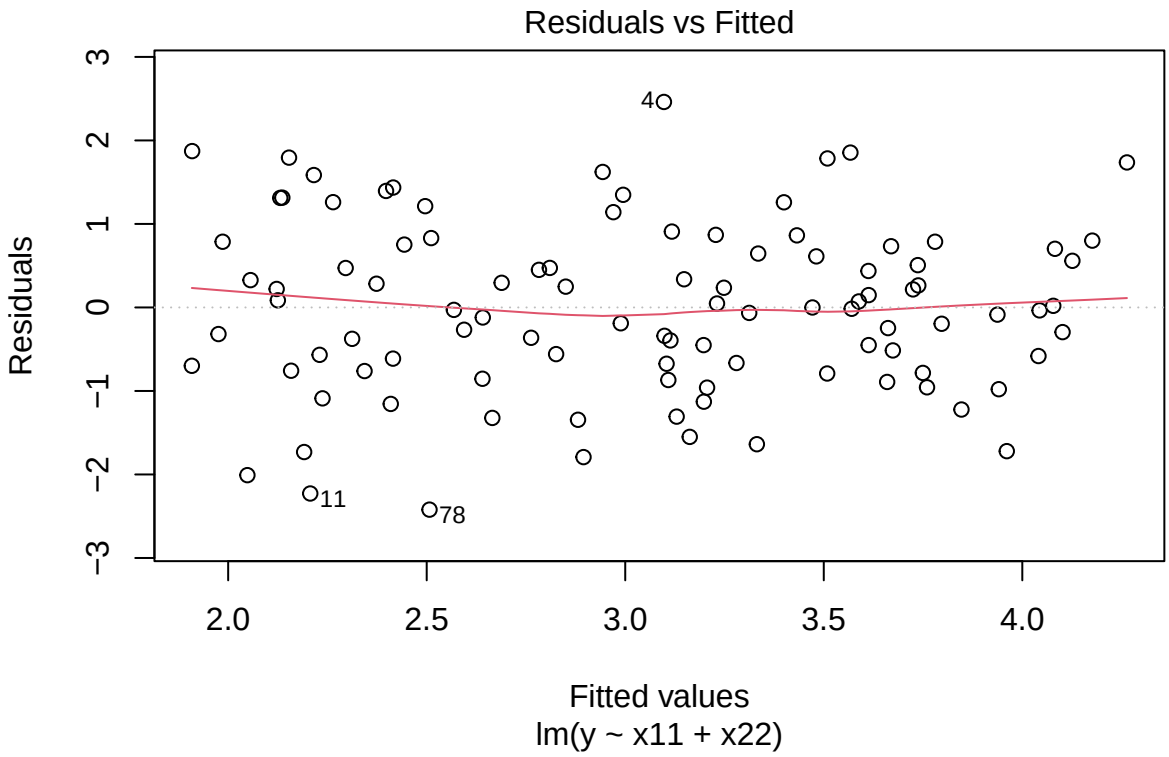
```
model_b = lm(y ~ x11)
summary(model_b)
```

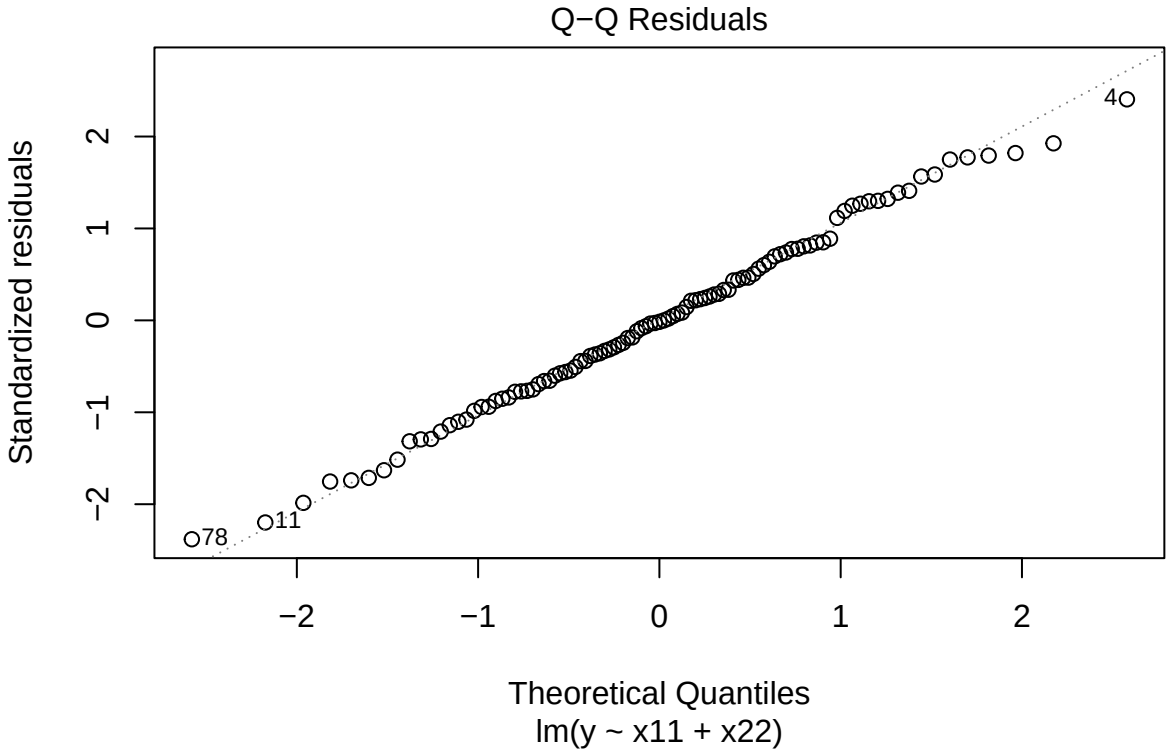
```
##
## Call:
## lm(formula = y ~ x11)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.37711 -0.71836 -0.05293  0.74684  2.47378
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9223     0.2033   9.456 1.68e-15 ***
## x11           2.1716     0.3416   6.356 6.37e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.025 on 99 degrees of freedom
## Multiple R-squared:  0.2898, Adjusted R-squared:  0.2827
## F-statistic:  40.4 on 1 and 99 DF,  p-value: 6.374e-09
```

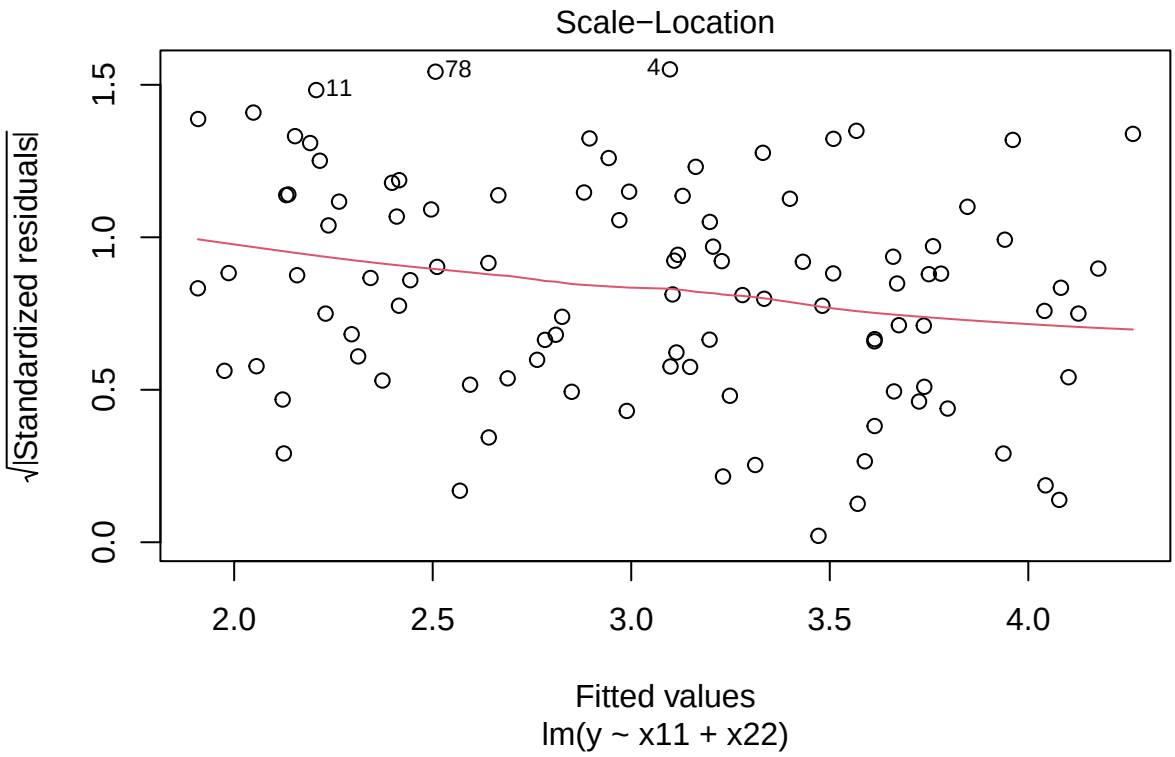
```
model_c = lm(y ~ x22)
summary(model_c)
```

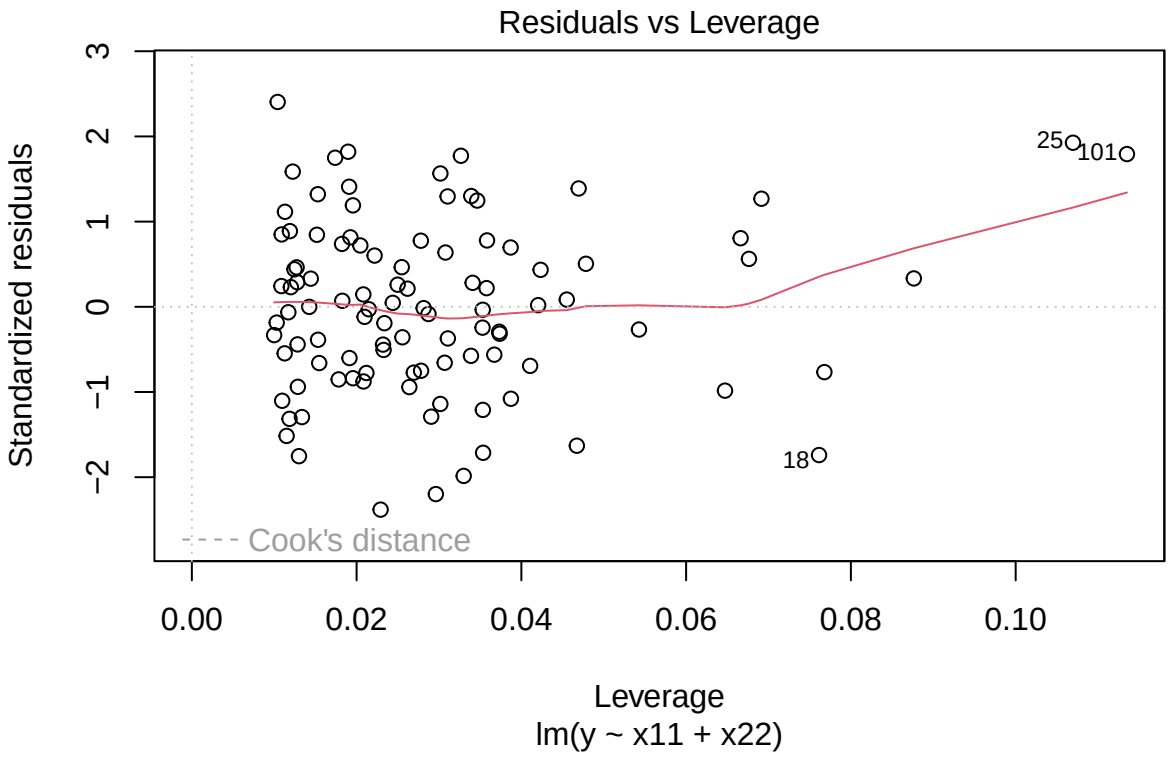
```
##
## Call:
## lm(formula = y ~ x22)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.78473 -0.68116  0.04555  0.75883  2.41526
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2627     0.1827  12.384  < 2e-16 ***
## x22           3.0455     0.5800   5.251 8.68e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.076 on 99 degrees of freedom
## Multiple R-squared:  0.2178, Adjusted R-squared:  0.2099
## F-statistic: 27.57 on 1 and 99 DF,  p-value: 8.677e-07
```

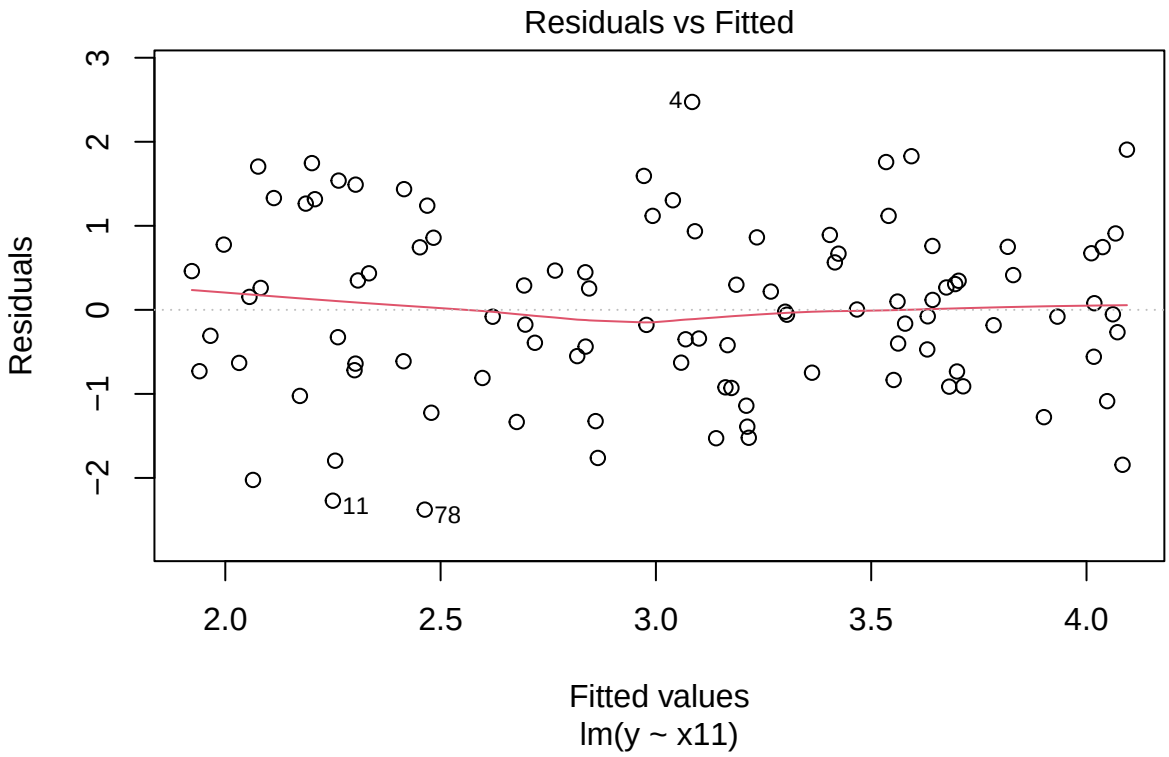
```
plot(model_a); plot(model_b); plot(model_c)
```

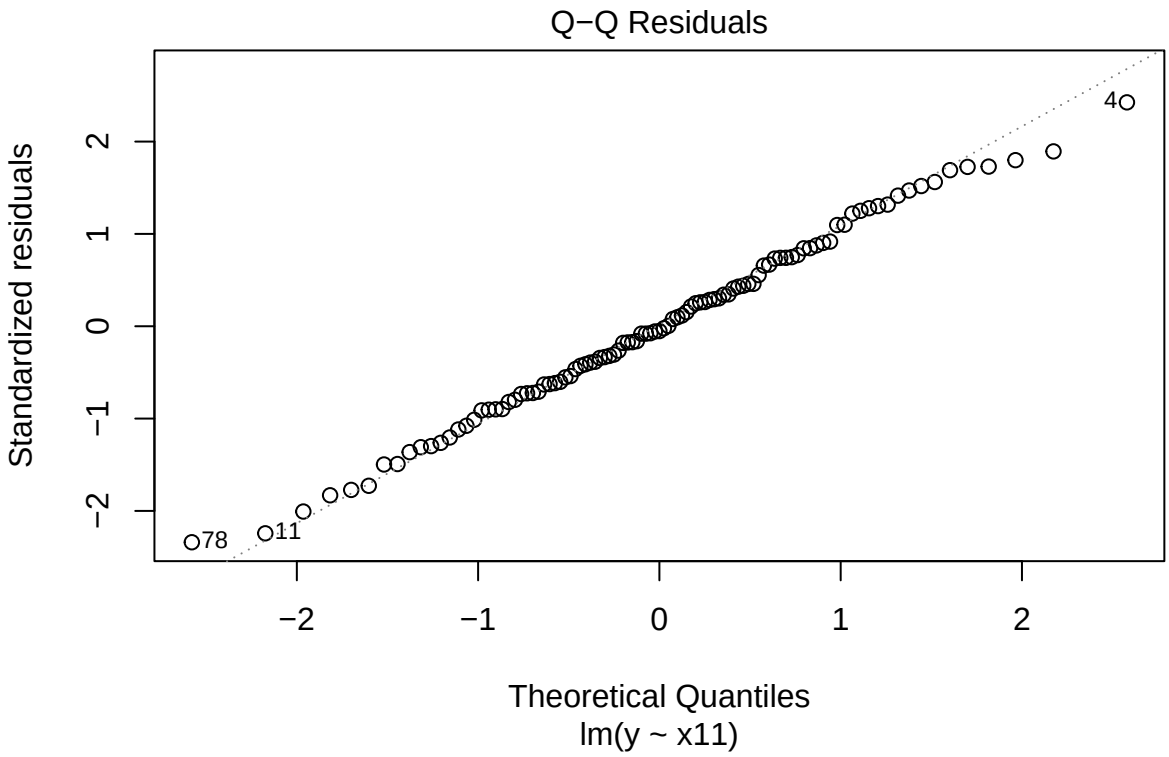

Residuals vs Fitted
Residuals
Fitted values
lm(y ~ x11 + x22)
4
11
78
3
2
1
0
−1
−2
−3
2.0
2.5
3.0
3.5
4.0

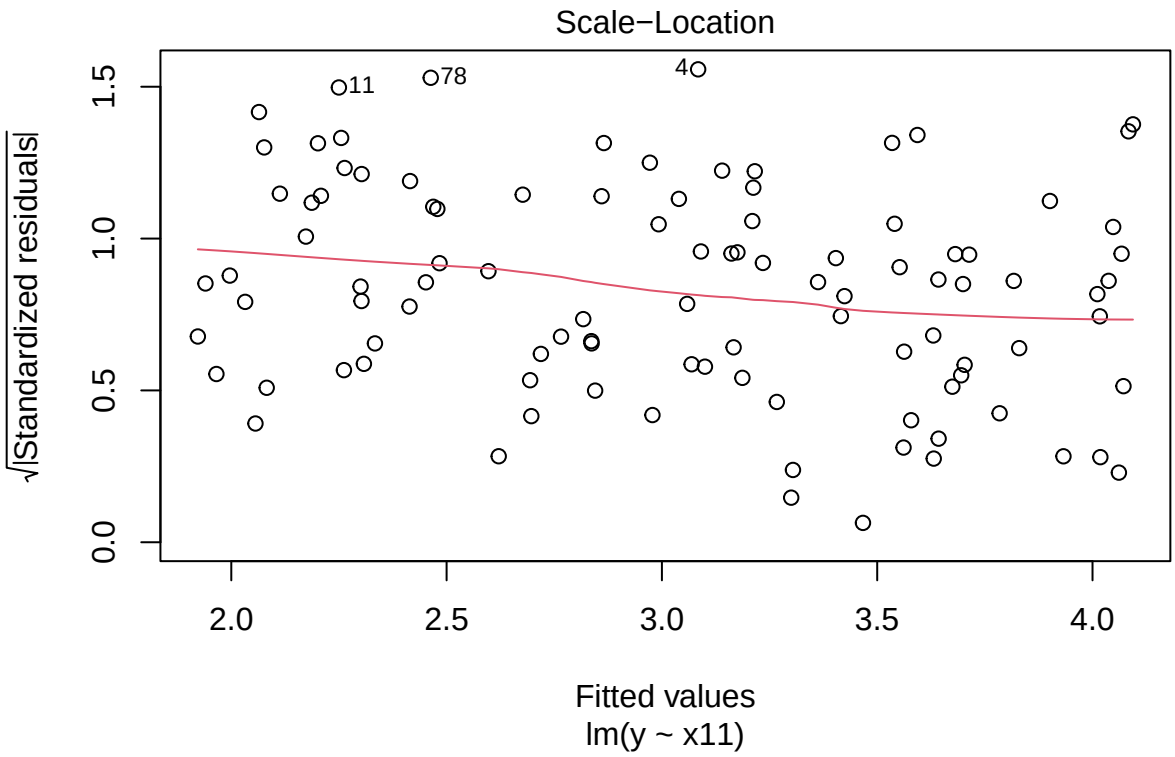

Q−Q Residuals
Standardized residuals
−2
−1
0
1
2
78
11
4
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x11 + x22)

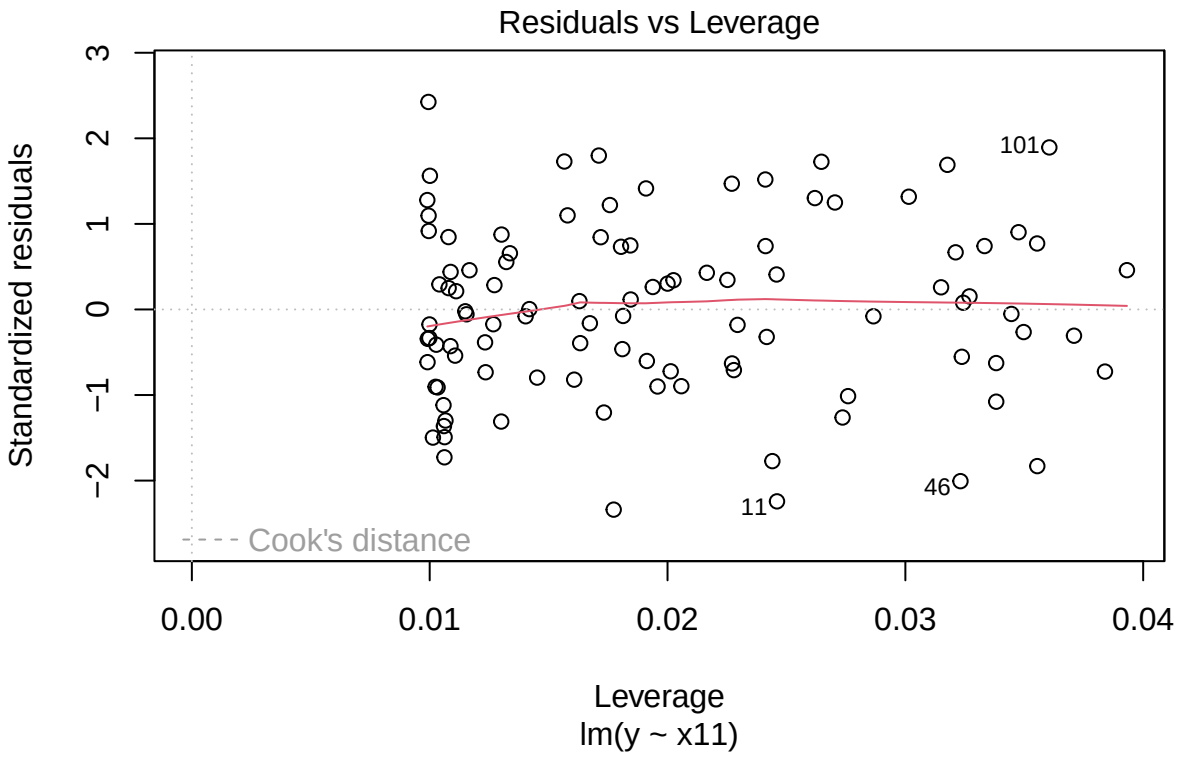

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
11
78
4
2.0
2.5
3.0
3.5
4.0
Fitted values
lm(y ~ x11 + x22)

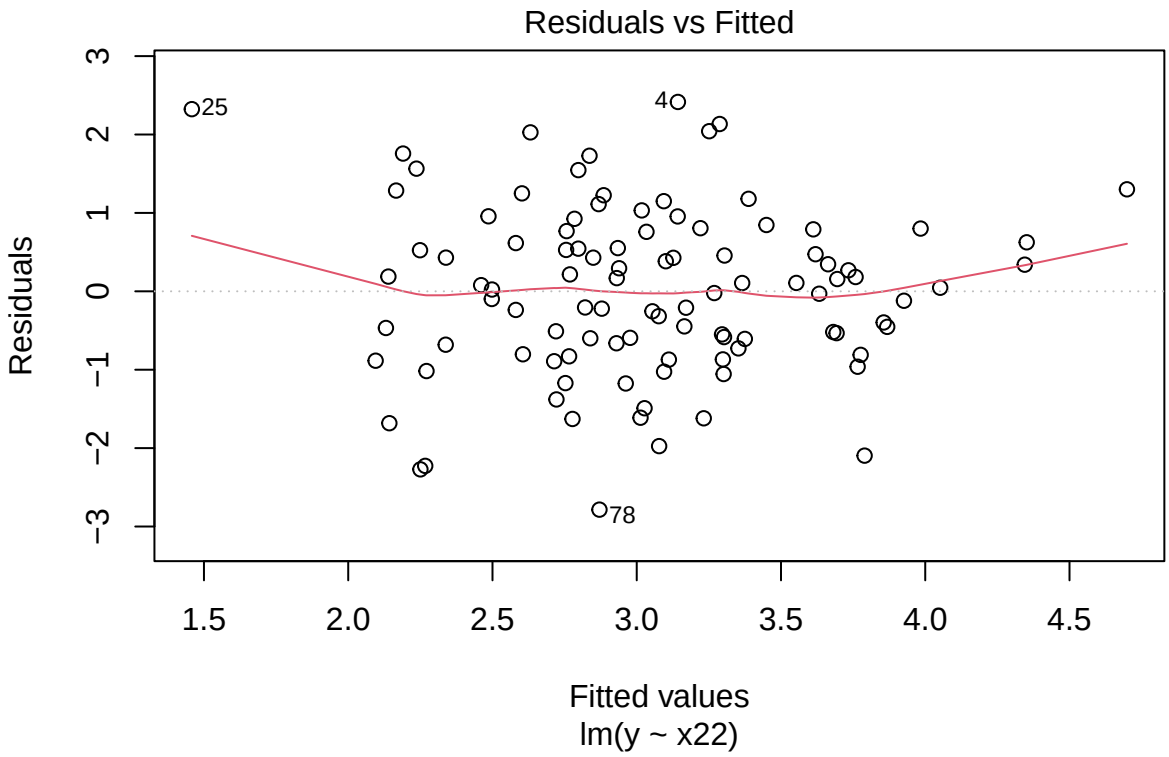

Residuals vs Leverage
Standardized residuals
3
2
1
0
−1
−2
25
101
18
Cook's distance
0.00
0.02
0.04
0.06
0.08
0.10
Leverage
lm(y ~ x11 + x22)

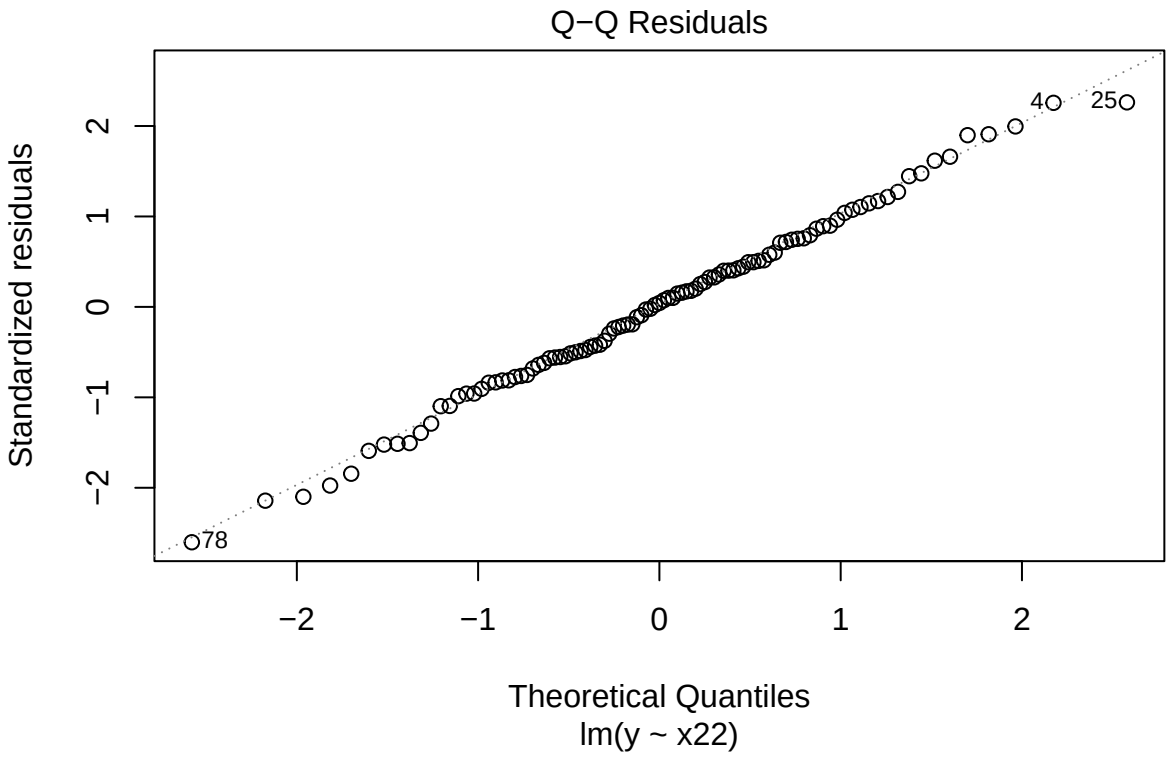

Residuals vs Fitted
Residuals
3
2
1
0
−1
−2
4
11
78
2.0
2.5
3.0
3.5
4.0
Fitted values
lm(y ~ x11)

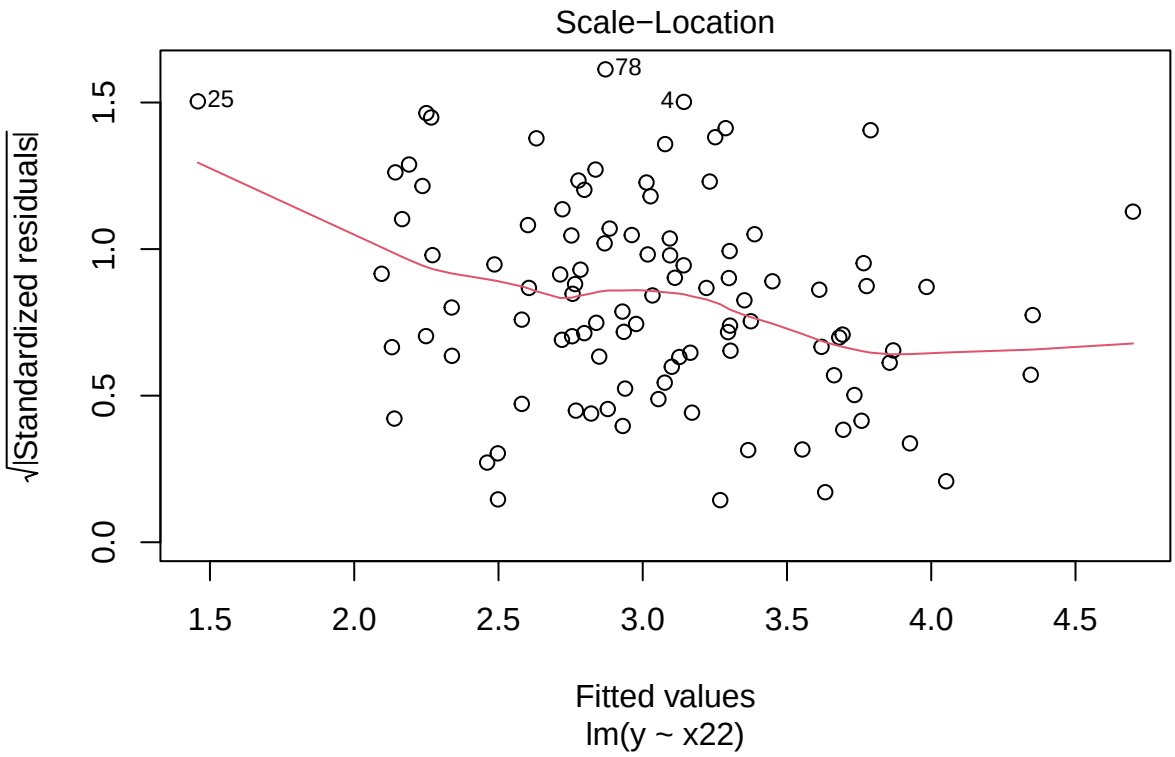

Q−Q Residuals
Standardized residuals
−2
−1
0
1
2
78
11
4
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x11)

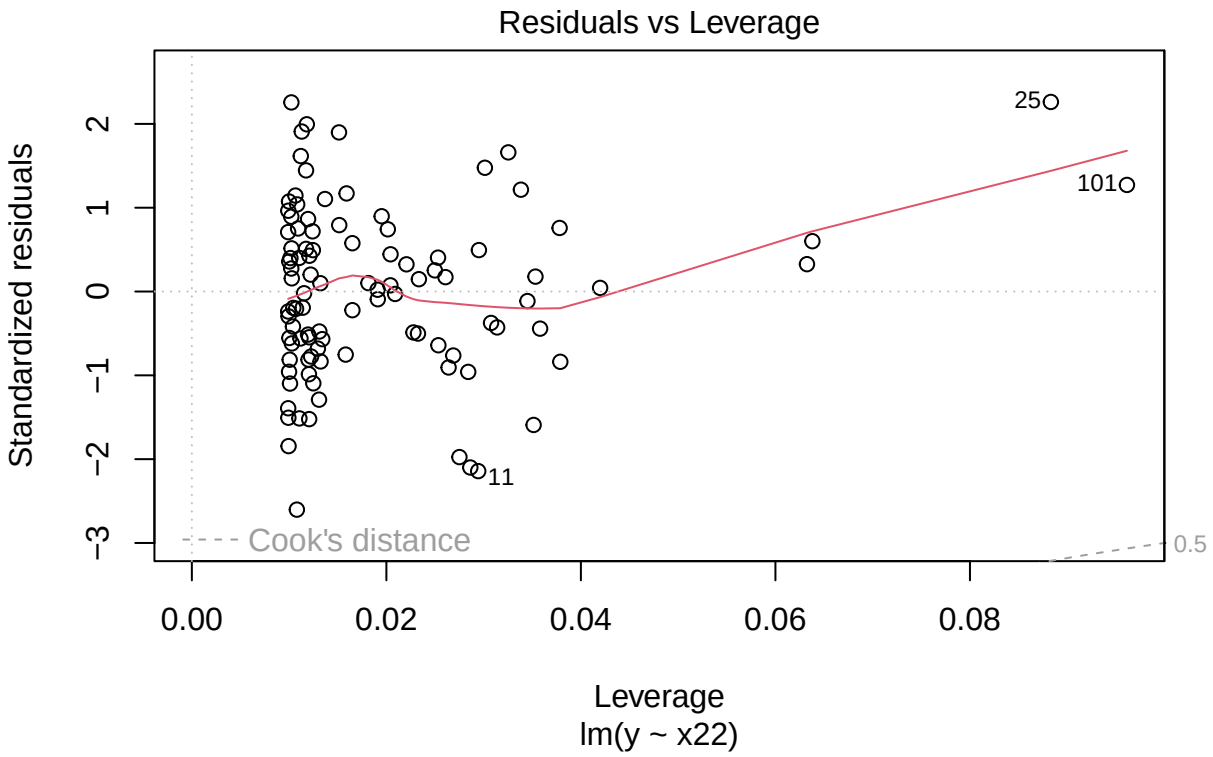

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
11
78
4
2.0
2.5
3.0
3.5
4.0
Fitted values
lm(y ~ x11)


Residuals vs Leverage
Standardized residuals
3
2
1
0
−1
−2
101
46
11
Cook's distance
0.00
0.01
0.02
0.03
0.04
Leverage
lm(y ~ x11)


Residuals vs Fitted
Residuals
Fitted values
lm(y ~ x22)
25
4
78
3
2
1
0
−1
−2
−3
1.5
2.0
2.5
3.0
3.5
4.0
4.5


Q−Q Residuals
Standardized residuals
−2
−1
0
1
2
78
4
25
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x22)


Scale−Location
78
25
4
√|Standardized residuals|
0.0
0.5
1.0
1.5
1.5
2.0
2.5
3.0
3.5
4.0
4.5
Fitted values
lm(y ~ x22)



The new data point is a high leverage point for the combined model and the model only containing `x2`. For the combined model, the new point is also a high residual point, being over 2 standard deviations away.

The interpretation for the combined model changes, because at the 0.05 significance level, now `x1` is insignificant in predicting `y` while `x2` now is. It is possible that this is due to the influence of the new data point having a high residual and high leverage.

The interpretations remain the same for the two individual-predictor models due to each individual predictor having all of the predictive power when used alone.

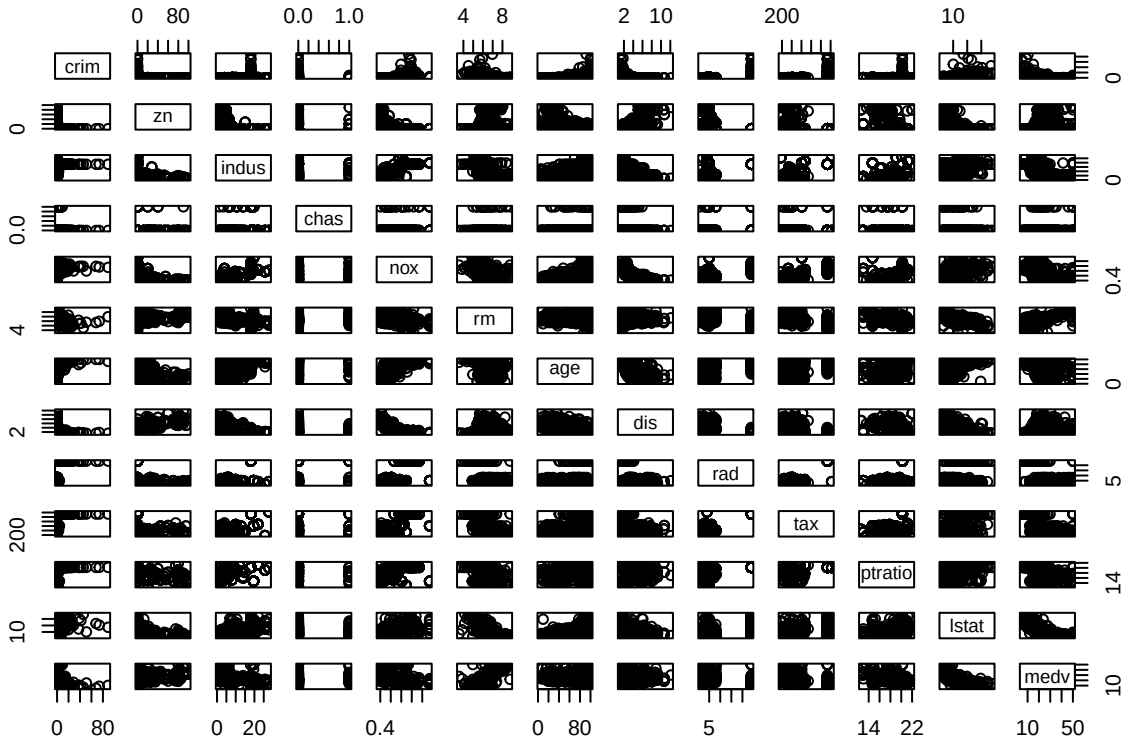
Question 15

This problem involves the `Boston` data set, which we saw in the lab for this chapter. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.

(a) For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions.

```
attach(Boston)

mod1 = lm(crim ~ zn)
mod2 = lm(crim ~ indus)
mod3 = lm(crim ~ chas)
mod4 = lm(crim ~ nox)
```

```
mod5 = lm(crim ~ rm)
mod6 = lm(crim ~ age)
mod7 = lm(crim ~ dis)
mod8 = lm(crim ~ rad)
mod9 = lm(crim ~ tax)
mod10 = lm(crim ~ ptratio)
mod11 = lm(crim ~ lstat)
mod12 = lm(crim ~ medv)

pairs(Boston)
```




```
summary(mod1)
```

```
##
## Call:
## lm(formula = crim ~ zn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = crim ~ indus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
summary(mod3)
```

```
##
## Call:
## lm(formula = crim ~ chas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

```
summary(mod4)
```

```
##
```

```
## Call:
## lm(formula = crim ~ nox)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
summary(mod5)
```

```
##
## Call:
## lm(formula = crim ~ rm)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

```
summary(mod6)
```

```
##
## Call:
## lm(formula = crim ~ age)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

```
summary(mod7)
```

```
##
## Call:
## lm(formula = crim ~ dis)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
summary(mod8)
```

```
##
## Call:
## lm(formula = crim ~ rad)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
summary(mod9)
```

```
##
```

```
## Call:
## lm(formula = crim ~ tax)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
summary(mod10)
```

```
##
## Call:
## lm(formula = crim ~ ptratio)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

```
summary(mod11)
```

```
##
## Call:
## lm(formula = crim ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
summary(mod12)
```

```
##
## Call:
## lm(formula = crim ~ medv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

All are individually significant except for `chas`.

(b) Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis $H_0 : \beta_j = 0$?

```
model_full = lm(crim  ~ . , data = Boston)
summary(model_full)
```

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.534 -2.248 -0.348  1.087 73.923
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.7783938  7.0818258   1.946 0.052271 .
## zn           0.0457100  0.0187903   2.433 0.015344 *
## indus       -0.0583501  0.0836351  -0.698 0.485709
## chas        -0.8253776  1.1833963  -0.697 0.485841
## nox         -9.9575865  5.2898242  -1.882 0.060370 .
## rm           0.6289107  0.6070924   1.036 0.300738
```

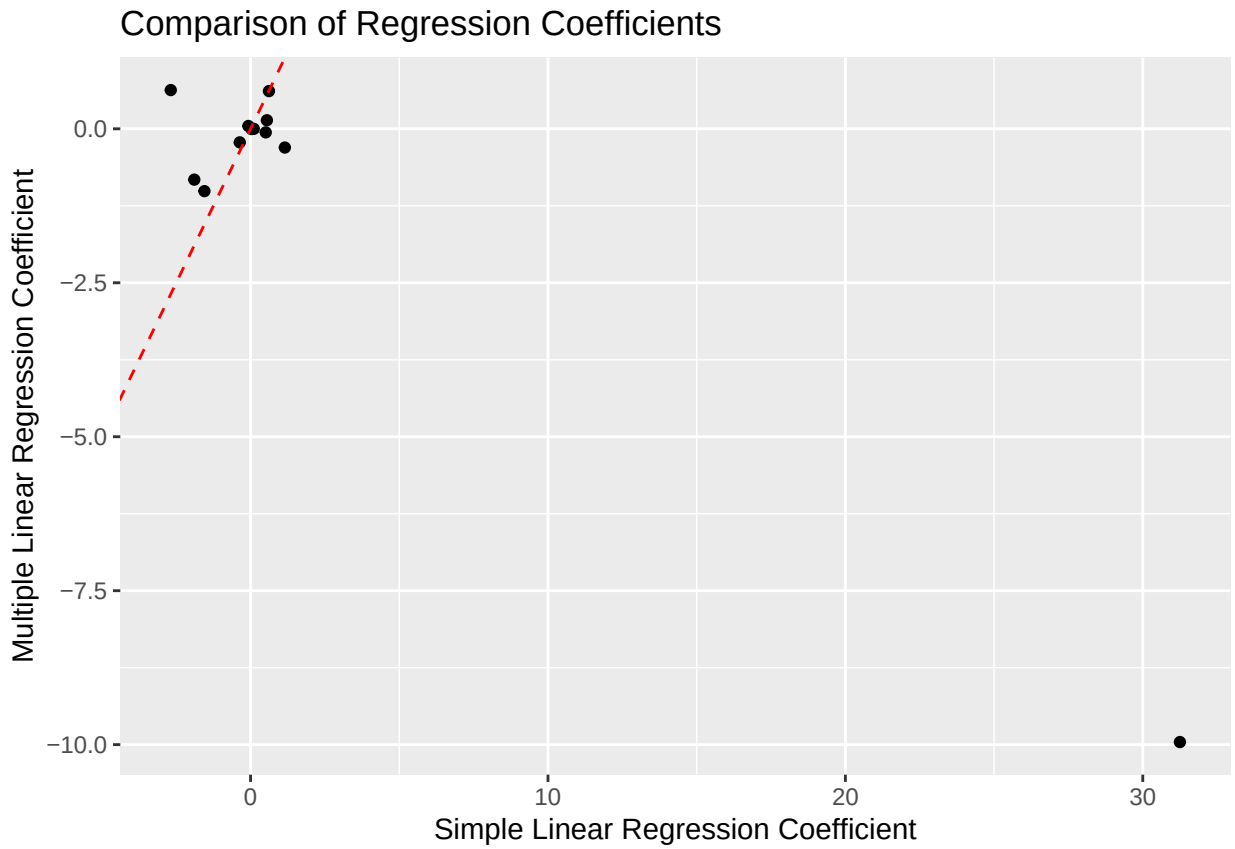
```
## age         -0.0008483  0.0179482  -0.047 0.962323
## dis         -1.0122467  0.2824676  -3.584 0.000373 ***
## rad          0.6124653  0.0875358   6.997 8.59e-12 ***
## tax         -0.0037756  0.0051723  -0.730 0.465757
## ptratio     -0.3040728  0.1863598  -1.632 0.103393
## lstat        0.1388006  0.0757213   1.833 0.067398 .
## medv        -0.2200564  0.0598240  -3.678 0.000261 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.46 on 493 degrees of freedom
## Multiple R-squared:  0.4493, Adjusted R-squared:  0.4359
## F-statistic: 33.52 on 12 and 493 DF,  p-value: < 2.2e-16
```

We can reject the null for all predictors except for `indus`, `chas`, `rm`, `age`, `tax` and `ptratio`.

(c) How do your results from (a) compare to your results from (b)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis.

```r
# Load the required library
library(ggplot2)

# Fit simple linear regression models for each predictor
simple_regression_models <- lapply(names(Boston)[-1], function(predictor) {
  lm_formula <- as.formula(paste("crim ~", predictor))
  lm(lm_formula, data = Boston)
})

# Extract coefficients from simple linear regression models
simple_regression_coefficients <- sapply(simple_regression_models, function(model) {
  coef(model)[2]  # Extract the coefficient for the predictor
})

# Fit the multiple regression model
multiple_regression_model <- lm(crim ~ ., data = Boston)

# Extract coefficients from the multiple regression model
multiple_regression_coefficients <- coef(multiple_regression_model)[-1]  # Exclude intercept

# Combine the results into a data frame
comparison_data <- data.frame(
  Predictor = names(Boston)[-1],
  SimpleRegressionCoefficient = simple_regression_coefficients,
  MultipleRegressionCoefficient = multiple_regression_coefficients
)

# Create the plot
ggplot(comparison_data, aes(x = SimpleRegressionCoefficient, y = MultipleRegressionCoefficient)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
```

```
  labs(title = "Comparison of Regression Coefficients",
       x = "Simple Linear Regression Coefficient",
       y = "Multiple Linear Regression Coefficient")
```



(d) Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor `X`, fit a model of the form: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X^2 + \beta_3 X^3 + \epsilon$

```
# Function to fit a polynomial regression model and display summary
fit_poly_model <- function(predictor, max_degree) {
  unique_values <- length(unique(Boston[[predictor]]))

  # Choose the maximum degree based on the number of unique values
  degree <- min(3, unique_values - 1)

  poly_model <- lm(crim ~ poly(Boston[[predictor]], degree = degree),
                   data = Boston)

  # Display summary
  cat("Summary for polynomial regression model with predictor:",
      predictor, "\n")
  print(summary(poly_model))

  # Return the model
  return(poly_model)
}
```

```r
# Fit polynomial regression models for each predictor
poly_models <- lapply(names(Boston)[-1],
                      function(predictor) fit_poly_model(predictor, 3))
```

```
## Summary for polynomial regression model with predictor: zn
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                                3.6135     0.3722   9.709
## poly(Boston[[predictor]], degree = degree)1 -38.7498     8.3722  -4.628
## poly(Boston[[predictor]], degree = degree)2  23.9398     8.3722   2.859
## poly(Boston[[predictor]], degree = degree)3 -10.0719     8.3722  -1.203
##                                          Pr(>|t|)
## (Intercept)                               < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)1  4.7e-06 ***
## poly(Boston[[predictor]], degree = degree)2  0.00442 **
## poly(Boston[[predictor]], degree = degree)3  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
##
## Summary for polynomial regression model with predictor: indus
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                                3.614      0.330  10.950
## poly(Boston[[predictor]], degree = degree)1   78.591      7.423  10.587
## poly(Boston[[predictor]], degree = degree)2  -24.395      7.423  -3.286
## poly(Boston[[predictor]], degree = degree)3  -54.130      7.423  -7.292
##                                          Pr(>|t|)
## (Intercept)                               < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)1  < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)2  0.00109 **
## poly(Boston[[predictor]], degree = degree)3  1.2e-12 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
##
## Summary for polynomial regression model with predictor: chas
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                3.6135     0.3822   9.455   <2e-16
## poly(Boston[[predictor]], degree = degree) -10.8036     8.5966  -1.257    0.209
##
## (Intercept)                                ***
## poly(Boston[[predictor]], degree = degree)
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
##
## Summary for polynomial regression model with predictor: nox
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##                                           Estimate Std. Error t value
## (Intercept)                                 3.6135     0.3216  11.237
## poly(Boston[[predictor]], degree = degree)1  81.3720     7.2336  11.249
## poly(Boston[[predictor]], degree = degree)2 -28.8286     7.2336  -3.985
## poly(Boston[[predictor]], degree = degree)3 -60.3619     7.2336  -8.345
##                                           Pr(>|t|)
## (Intercept)                                < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)1  < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)2 7.74e-05 ***
## poly(Boston[[predictor]], degree = degree)3 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
##
## Summary for polynomial regression model with predictor: rm
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                   3.6135     0.3703   9.758
## poly(Boston[[predictor]], degree = degree)1 -42.3794     8.3297  -5.088
## poly(Boston[[predictor]], degree = degree)2  26.5768     8.3297   3.191
## poly(Boston[[predictor]], degree = degree)3  -5.5103     8.3297  -0.662
##                                             Pr(>|t|)
## (Intercept)                                  < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)1 5.13e-07 ***
## poly(Boston[[predictor]], degree = degree)2  0.00151 **
## poly(Boston[[predictor]], degree = degree)3  0.50858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
##
## Summary for polynomial regression model with predictor: age
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                   3.6135     0.3485  10.368
## poly(Boston[[predictor]], degree = degree)1  68.1820     7.8397   8.697
## poly(Boston[[predictor]], degree = degree)2  37.4845     7.8397   4.781
## poly(Boston[[predictor]], degree = degree)3  21.3532     7.8397   2.724
##                                             Pr(>|t|)
## (Intercept)                                  < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)1  < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)2 2.29e-06 ***
## poly(Boston[[predictor]], degree = degree)3  0.00668 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
##
## Summary for polynomial regression model with predictor: dis
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                   3.6135     0.3259  11.087
## poly(Boston[[predictor]], degree = degree)1 -73.3886     7.3315 -10.010
## poly(Boston[[predictor]], degree = degree)2  56.3730     7.3315   7.689
## poly(Boston[[predictor]], degree = degree)3 -42.6219     7.3315  -5.814
##                                             Pr(>|t|)
## (Intercept)                                  < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)1  < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)2 7.87e-14 ***
## poly(Boston[[predictor]], degree = degree)3 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
##
## Summary for polynomial regression model with predictor: rad
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                   3.6135     0.2971  12.164
## poly(Boston[[predictor]], degree = degree)1 120.9074     6.6824  18.093
## poly(Boston[[predictor]], degree = degree)2  17.4923     6.6824   2.618
## poly(Boston[[predictor]], degree = degree)3   4.6985     6.6824   0.703
##                                             Pr(>|t|)
## (Intercept)                                  < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)1  < 2e-16 ***
```

```
## poly(Boston[[predictor]], degree = degree)2  0.00912 **
## poly(Boston[[predictor]], degree = degree)3  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
##
## Summary for polynomial regression model with predictor: tax
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##                                           Estimate Std. Error t value
## (Intercept)                                 3.6135     0.3047  11.860
## poly(Boston[[predictor]], degree = degree)1 112.6458     6.8537  16.436
## poly(Boston[[predictor]], degree = degree)2  32.0873     6.8537   4.682
## poly(Boston[[predictor]], degree = degree)3  -7.9968     6.8537  -1.167
##                                           Pr(>|t|)
## (Intercept)                                < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)1  < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)2 3.67e-06 ***
## poly(Boston[[predictor]], degree = degree)3    0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16
##
## Summary for polynomial regression model with predictor: ptratio
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                                           Estimate Std. Error t value
## (Intercept)                                  3.614      0.361  10.008
## poly(Boston[[predictor]], degree = degree)1   56.045      8.122   6.901
## poly(Boston[[predictor]], degree = degree)2   24.775      8.122   3.050
## poly(Boston[[predictor]], degree = degree)3  -22.280      8.122  -2.743
##                                           Pr(>|t|)
```

```
## (Intercept)                                     < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)1 1.57e-11 ***
## poly(Boston[[predictor]], degree = degree)2  0.00241 **
## poly(Boston[[predictor]], degree = degree)3  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
##
## Summary for polynomial regression model with predictor: lstat
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##                                           Estimate Std. Error t value
## (Intercept)                                 3.6135     0.3392  10.654
## poly(Boston[[predictor]], degree = degree)1  88.0697     7.6294  11.543
## poly(Boston[[predictor]], degree = degree)2  15.8882     7.6294   2.082
## poly(Boston[[predictor]], degree = degree)3 -11.5740     7.6294  -1.517
##                                           Pr(>|t|)
## (Intercept)                                 <2e-16 ***
## poly(Boston[[predictor]], degree = degree)1   <2e-16 ***
## poly(Boston[[predictor]], degree = degree)2   0.0378 *
## poly(Boston[[predictor]], degree = degree)3   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
##
## Summary for polynomial regression model with predictor: medv
##
## Call:
## lm(formula = crim ~ poly(Boston[[predictor]], degree = degree),
##     data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##                                           Estimate Std. Error t value
## (Intercept)                                  3.614      0.292  12.374
## poly(Boston[[predictor]], degree = degree)1  -75.058      6.569 -11.426
## poly(Boston[[predictor]], degree = degree)2   88.086      6.569  13.409
```

```
## poly(Boston[[predictor]], degree = degree)3  -48.033      6.569  -7.312
##                                                Pr(>|t|)
## (Intercept)                                     < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)1     < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)2     < 2e-16 ***
## poly(Boston[[predictor]], degree = degree)3 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Yes, several models statistically significantly predict `crim` with all three predictors.