chapter3_Exercises_labs

2024-08-19

Exercises 1. Describe the null hypotheses to which the p-values given in Table 3.4 correspond. Explain what conclusions you can draw based on these p-values. Your explanation should be phrased in terms of sales, TV, radio, and newspaper, rather than in terms of the coefficients of the linear model.

#the p-values associated with TV and radio are significant,(reject H0) and they indicate that TV and radio are related to sales, but that there is no evidence that newspaper is associated with sales, when TV and radio are held fxed.

2. Carefully explain the differences between the KNN classifier and KNN regression methods.

#KNN regression first identifies the K training observations that are closest to x0 (represented by N0), and then estimates f(x0) using the average of all the training responses in N0.

#KNN classifier classifies the test observation x0 to the class with the largest probability from

3. Suppose we have a data set with five predictors, X1 = GPA, X2 = IQ, X3 = Level (1 for College and 0 for High School), X4 = Interaction between GPA and IQ, and X5 = Interaction between GPA and Level. The response is starting salary after graduation (in thousands of dollars). Suppose we use least squares to ft the model, and get ˆ0 = 50, ˆ1 = 20, ˆ2 = 0.07, ˆ3 = 35, ˆ4 = 0.01, ˆ5 = −10.

#start_salary = 50+20*GPA+0.07*IQ+35*Level+0.01*GPA*IQ+ −10*GPA*Level

(a) Which answer is correct, and why?

i. For a fixed value of IQ and GPA, high school graduates earn more, on average, than college graduates. F;35>0
ii. For a fixed value of IQ and GPA, college graduates earn more, on average, than high school graduates.

iii. For a fixed value of IQ and GPA, high school graduates earn more, on average, than college graduates provided that the GPA is high enough.
iv. For a fixed value of IQ and GPA, college graduates earn more, on average, than high school graduates provided that the GPA is high enough.

#iii $(35 - 10\ GPA)$Level ; if the GPA is high enough, the coefficient can be negative

(b) Predict the salary of a college graduate with IQ of 110 and a GPA of 4.0.

#137.1k

```
50 + 20*4.0 + 0.07*110 + 35*1 + 0.01*4.0*110 - 10*4.0*1
```

```
[1] 137.1
```

(c) True or false: Since the coefficient for the GPA/IQ interaction term is very small, there is very little evidence of an interaction effect. Justify your answer.

#the p-value is needed to indicate whether there's interaction effect

4. I collect a set of data (n = 100 observations) containing a single predictor and a quantitative response. I then fit a linear regression model to the data, as well as a separate cubic regression, i.e. Y = 0 + 1X + 2X2 + 3X3 + .

(a) Suppose that the true relationship between X and Y is linear, i.e. Y = 0 + 1X + . Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

#when the true relationship is linear, the resulting fit of cubic regression seems unnecessarily wiggy

#we expect the training RSS for the cubic regression to be lower than the other, casue it's more flexible and may lead to overfit

(b) Answer (a) using test rather than training RSS.

#the RSS for linear regression would be lower, casue the true relationship between X and Y is linear

(c) Suppose that the true relationship between X and Y is not linear, but we don't know how far it is from linear. Consider the training RSS for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

#we expect the training RSS for the cubic regression to be lower than the other

(d) Answer (c) using test rather than training RSS.

#there's not enough information to tell; because we are not sure if the true f is highly non-linear

5. Consider the fitted values that result from performing linear regression without an intercept. In this setting, the ith fitted value takes the form y^i = xi , where

```r
formula1 <- readPNG("formula1.png")
p1<-ggplot()+background_image(formula1)+theme_void()
p1
```

where

$$\hat{\beta} = \left(\sum_{i=1}^{n} x_i y_i\right) \Big/ \left(\sum_{i'=1}^{n} x_{i'}^2\right). \tag{3.38}$$

Show that we can write

$$\hat{y}_i = \sum_{i'=1}^{n} a_{i'} y_{i'}.$$

What is $a_{i'}$?

Note: We interpret this result by saying that the fitted values from linear regression are linear combinations of the response values.

$$\alpha_{i'} = \frac{x_i \times x_{i'}}{\sum_{i''=1}^{n} x_{i''}^2}$$

6. Using (3.4), argue that in the case of simple linear regression, the least squares line always passes through the point (¯x, y¯).

```r
formula2 <- readPNG("6.png")
p2<-ggplot()+background_image(formula2)+theme_void()
p2
```

$$x = \bar{x}$$

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

$$\hat{\beta}\_0 = \bar{y} - \hat{\beta}\_1\bar{x}$$

$$\hat{y} = \left(\bar{y} - \hat{\beta}\_1\bar{x}\right) + \hat{\beta}\_1\bar{x}$$

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

7. It is claimed in the text that in the case of simple linear regression of Y onto X, the R2 statistic (3.17) is equal to the square of the correlation between X and Y (3.18). Prove that this is the case. For simplicity, you may assume that ¯x = ¯y = 0.

```r
formula3 <- readPNG("7.png")
p3<-ggplot()+background_image(formula3)+theme_void()
p3
```

$$\text{TSS} = \sum_{i=1}^{n} y_i^2$$

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \hat{\beta}_1 x_i\right)^2$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2} \quad (\bar{x} = \bar{y} = 0)$$

$$\hat{\beta}_0 = 0$$

$$\hat{y}_i = \hat{\beta}_1 x_i$$

$$\text{RSS} = \sum_{i=1}^{n} y_i^2 - \frac{\left(\sum_{i=1}^{n} x_i y_i\right)^2}{\sum_{i=1}^{n} x_i^2}$$

Thus:

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

The correlation (r) between (X) and (Y) is:

$$r = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$$

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

Thus, we have:

$$R^2 = r^2$$

8. This question involves the use of simple linear regression on the Auto data set. (a) Use the lm() function to perform a simple linear regression with mpg as the response and horsepower as the predictor. Use the summary() function to print the results. Comment on the output. For example:

```r
Auto <- read_csv("Auto.csv")
```

```
Rows: 397 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): horsepower, name
dbl (7): mpg, cylinders, displacement, weight, acceleration, year, origin

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
Auto$horsepower <- as.numeric(Auto$horsepower)
```

```
Warning: NAs introduced by coercion
```

```r
lm.fit.1 <- lm(mpg ~ horsepower, data = Auto)
summary(lm.fit.1)
```

```
Call:
lm(formula = mpg ~ horsepower, data = Auto)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.5710  -3.2592  -0.3435   2.7630  16.9240 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 39.935861   0.717499   55.66   <2e-16 ***
horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.906 on 390 degrees of freedom
  (5 observations deleted due to missingness)
Multiple R-squared:  0.6059,    Adjusted R-squared:  0.6049 
F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

i. Is there a relationship between the predictor and the response? #the low p-value of horsepower indicates there's a relationship
ii. How strong is the relationship between the predictor and the response?

#the RSE is 4.9057569 units while the mean value for the response is 23.515869 units, indicating a percentage error of roughly 0.2086147

#the R^2 statistic records the percentage of variability in the response that is explained by the predictors.The predictors explain around 60 % of the variance in mpg.

iii. Is the relationship between the predictor and the response positive or negative? #negative
iv. What is the predicted mpg associated with a horsepower of 98? What are the associated 95 % confidence and prediction intervals?

#the predicted mpg associated with a horsepower of 98 and the associated 95 % confidence intervals: 24.4670772, 23.973079, 24.9610753

```r
predict(lm.fit.1, data.frame(horsepower = 98), interval = "confidence")
```

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

#the 95% prediction intervals:24.4670772, 14.8093961, 34.1247582

```r
predict(lm.fit.1, data.frame(horsepower = 98), interval = "prediction")
```

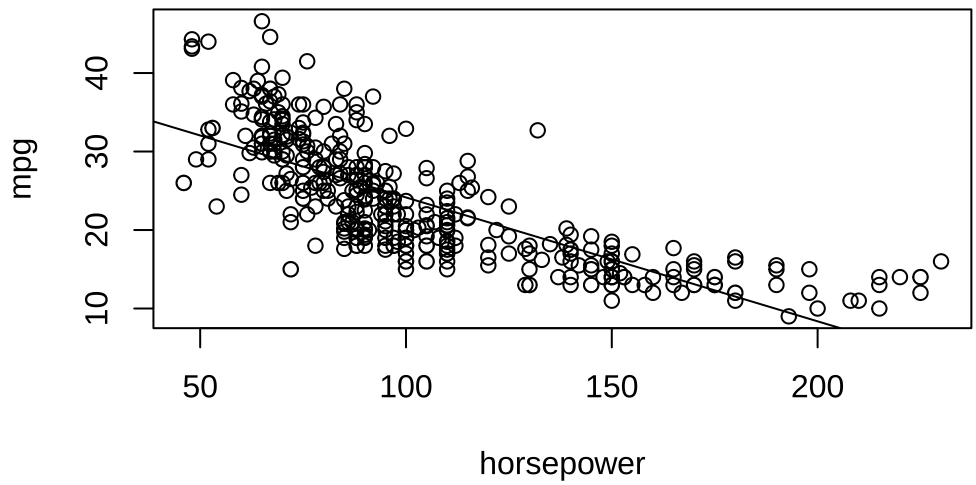
```
       fit     lwr      upr
1 24.46708 14.8094 34.12476
```

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

```r
attach(Auto)
```

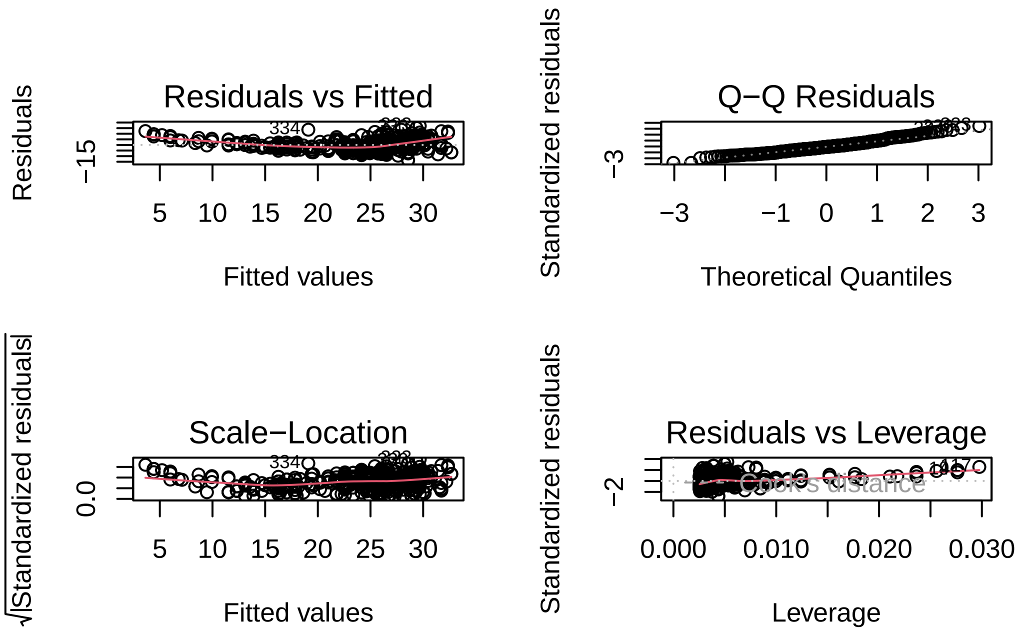
```
The following object is masked from package:lubridate:

    origin

The following object is masked from package:ggplot2:

    mpg
```

```r
plot(horsepower, mpg)
abline(lm.fit.1)
```



(c) Use the plot() function to produce diagnostic plots of the least squares regression ft. Comment on any problems you see with the fit. #there is some evidence of non-linearity

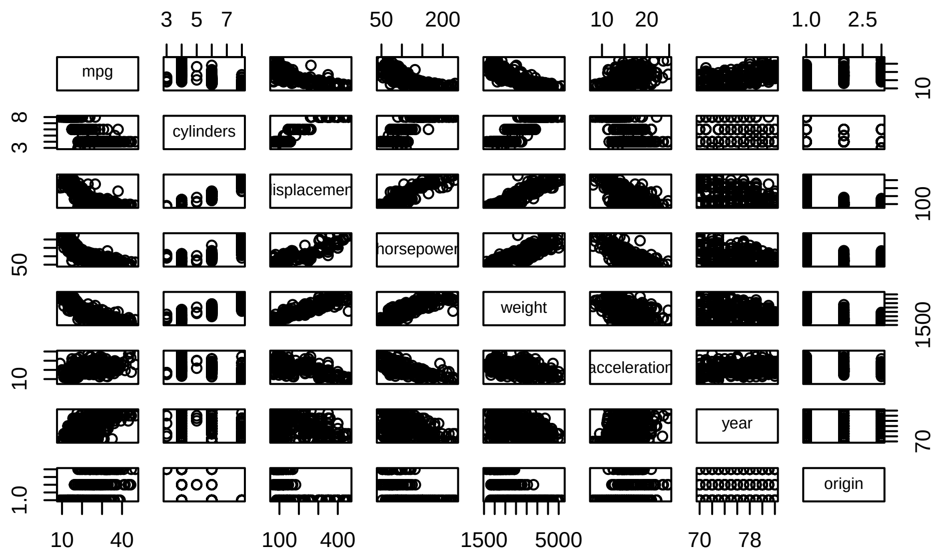
```
par(mfrow = c(2, 2))
plot(lm.fit.1)
```



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.

```
#name is not numeric
pairs(Auto[, 1:8])
```



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

```
cor(Auto[, 1:8],use = "complete.obs") #ignore nas
```

```
                    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:

```r
lm.fit.2 <- lm(mpg ~ .-name, data = Auto)
summary(lm.fit.2)
```

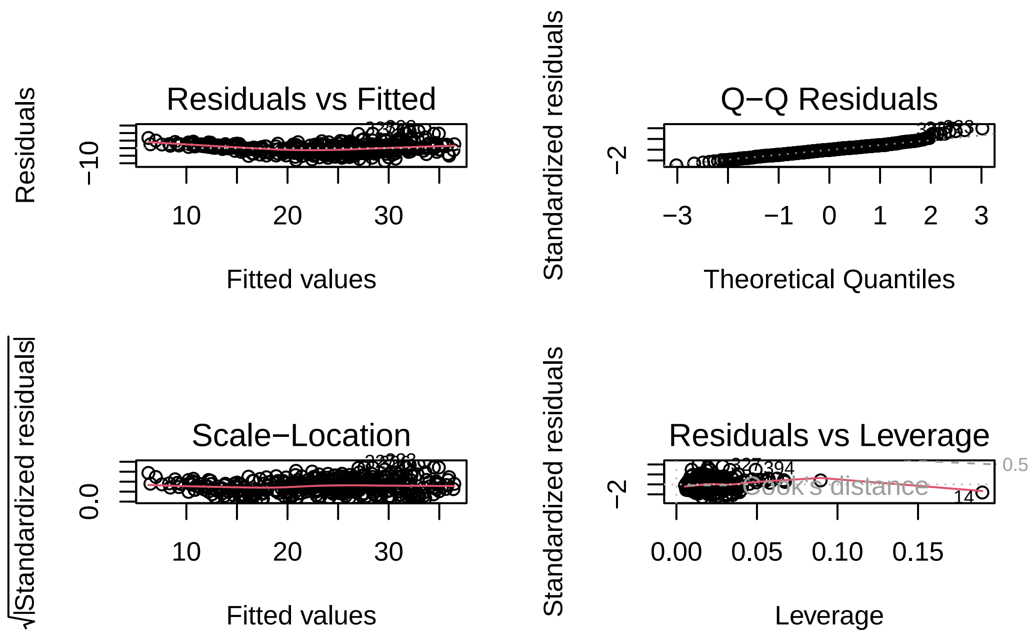
```
Call:
lm(formula = mpg ~ . - name, data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.5903 -2.1565 -0.1169  1.8690 13.0604 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
cylinders     -0.493376   0.323282  -1.526  0.12780    
displacement   0.019896   0.007515   2.647  0.00844 ** 
horsepower    -0.016951   0.013787  -1.230  0.21963    
weight        -0.006474   0.000652  -9.929  < 2e-16 ***
acceleration   0.080576   0.098845   0.815  0.41548    
year           0.750773   0.050973  14.729  < 2e-16 ***
origin         1.426141   0.278136   5.127 4.67e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.328 on 384 degrees of freedom
  (5 observations deleted due to missingness)
Multiple R-squared:  0.8215,    Adjusted R-squared:  0.8182 
F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

i. Is there a relationship between the predictors and the response? #yes, We reject the null hypothesis according to the F-statistic and p-value
ii. Which predictors appear to have a statistically significant relationship to the response? #displacement;weight;year;origin
iii. What does the coefficient for the year variable suggest? #the coefficient suggest that a 1-year increase is associated with an average increase in mpg of about 0.75 units.

(d) Use the plot() function to produce diagnostic plots of the linear regression ft. 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?

#there is some evidence of non-linearity #residual plots don't suggest any unusually large outliers #observation 14 in the residuals and leverage has high leverage

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



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

```
lm.fit.3 <- lm(mpg ~ . * ., data = Auto[, -9])
summary(lm.fit.3)
```

```
Call:
lm(formula = mpg ~ . * ., data = Auto[, -9])

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-7.6303 -1.4481  0.0596  1.2739 11.1386

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)   
(Intercept)               3.548e+01  5.314e+01   0.668  0.50475   
cylinders                 6.989e+00  8.248e+00   0.847  0.39738   
displacement             -4.785e-01  1.894e-01  -2.527  0.01192 * 
horsepower                5.034e-01  3.470e-01   1.451  0.14769   
weight                    4.133e-03  1.759e-02   0.235  0.81442   
acceleration             -5.859e+00  2.174e+00  -2.696  0.00735 **
year                      6.974e-01  6.097e-01   1.144  0.25340   
origin                   -2.090e+01  7.097e+00  -2.944  0.00345 **
cylinders:displacement   -3.383e-03  6.455e-03  -0.524  0.60051   
cylinders:horsepower      1.161e-02  2.420e-02   0.480  0.63157   
cylinders:weight          3.575e-04  8.955e-04   0.399  0.69000   
cylinders:acceleration    2.779e-01  1.664e-01   1.670  0.09584 . 
cylinders:year           -1.741e-01  9.714e-02  -1.793  0.07389 . 
cylinders:origin          4.022e-01  4.926e-01   0.816  0.41482   
displacement:horsepower  -8.491e-05  2.885e-04  -0.294  0.76867   
displacement:weight       2.472e-05  1.470e-05   1.682  0.09342 . 
displacement:acceleration -3.479e-03  3.342e-03  -1.041  0.29853   
displacement:year         5.934e-03  2.391e-03   2.482  0.01352 * 
displacement:origin       2.398e-02  1.947e-02   1.232  0.21875   
horsepower:weight        -1.968e-05  2.924e-05  -0.673  0.50124   
horsepower:acceleration  -7.213e-03  3.719e-03  -1.939  0.05325 . 
horsepower:year          -5.838e-03  3.938e-03  -1.482  0.13916   
horsepower:origin         2.233e-03  2.930e-02   0.076  0.93931   
weight:acceleration       2.346e-04  2.289e-04   1.025  0.30596   
weight:year              -2.245e-04  2.127e-04  -1.056  0.29182   
weight:origin            -5.789e-04  1.591e-03  -0.364  0.71623   
acceleration:year         5.562e-02  2.558e-02   2.174  0.03033 * 
acceleration:origin       4.583e-01  1.567e-01   2.926  0.00365 **
year:origin               1.393e-01  7.399e-02   1.882  0.06062 . 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.695 on 363 degrees of freedom
  (5 observations deleted due to missingness)
Multiple R-squared:  0.8893,    Adjusted R-squared:  0.8808 
F-statistic: 104.2 on 28 and 363 DF,  p-value: < 2.2e-16
```

#displacement:year #acceleration:year

#acceleration:origin (f) Try a few different transformations of the variables, such as log(X), $\sqrt{X}$, X2. Comment on your findings. #the log(X) model provide a better fit considering that it increases the R^2 and lowers the RSE

```
summary(lm(mpg ~ log(horsepower), data = Auto))
```

```
Call:
lm(formula = mpg ~ log(horsepower), data = Auto)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.2299  -2.7818  -0.2322   2.6661  15.4695 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)     108.6997     3.0496   35.64   <2e-16 ***
log(horsepower) -18.5822     0.6629  -28.03   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.501 on 390 degrees of freedom
  (5 observations deleted due to missingness)
Multiple R-squared:  0.6683,	Adjusted R-squared:  0.6675 
F-statistic: 785.9 on 1 and 390 DF,  p-value: < 2.2e-16
```

10. This question should be answered using the Carseats data set.

(a) Fit a multiple regression model to predict Sales using Price, Urban, and US.

```
lm.fit.3 <- lm(Sales ~ Price + Urban + US, data = Carseats)
summary(lm.fit.3)
```

```
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) Provide an interpretation of each coefficient in the model. Be careful—some of the variables in the model are qualitative! #the coefficient of price indicates that a 1 unit price decrease is associated with an average increase in sales of about 5.4% unit #the baseline is UrbanNOT. The coefficient of UrbanYes indicates that sales in Urban will be 0.021916 units lower(high p-value; not significant) #the coefficient of USYES shows that the sales in US will be 1.200573 higher compared with those NON-US

(c) Write out the model in equation form, being careful to handle the qualitative variables properly. #sales = 13.043469 - 0.054459*Price - 0.021916*Urban + 1.200573*US #(if_else(Urban = TRUE,1,0)) #(if_else(US = TRUE,1,0))

(d) For which of the predictors can you reject the null hypothesis H0 : j = 0? #T-test; we can reject the null hypothesis for Price and USYes; but there's no enough evidence that we could reject N0 for UrbanYes

(e) On the basis of your response to the previous question, fit a smaller model that only uses the predictors for which there is evidence of association with the outcome.

```r
lm.fit.4 <- lm(Sales ~ Price + US, data = Carseats)
summary(lm.fit.4)
```

```
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) How well do the models in (a) and (e) fit the data? # models in (a) fits as well as (e), and the simplified (e) is preferred.

```r
anova(lm.fit.3,lm.fit.4)
```

```
Analysis of Variance Table

Model 1: Sales ~ Price + Urban + US
Model 2: Sales ~ Price + US
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    396 2420.8
2    397 2420.9 -1  -0.03979 0.0065 0.9357
```

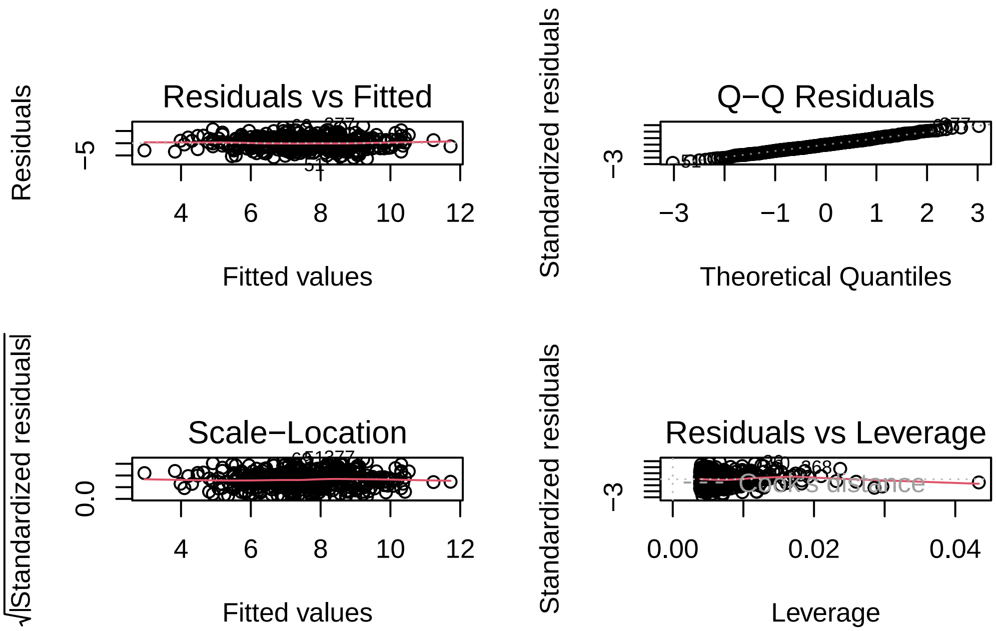
(g) Using the model from (e), obtain 95 % confidence intervals for the coefficient(s).

```r
confint(lm.fit.4, level = 0.95)
```

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

(h) Is there evidence of outliers or high leverage observations in the model from (e)? #there's no evidence for outliers but there are high leverage observations in the model

```r
par(mfrow = c(2, 2))
plot(lm.fit.4)
```



11. In this problem we will investigate the t-statistic for the null hypothesis H0 : = 0 in simple linear regression without an intercept. To begin, we generate a predictor x and a response y as follows.

(a) Perform a simple linear regression of y onto x, without an intercept. Report the coefficient estimate ˆ, the standard error of this coefficient estimate, and the t-statistic and p-value associated with the null hypothesis H0 : = 0. Comment on these results. (You can perform regression without an intercept using the command lm(y x+0).) #coefficient estimate: 1.9939 #t-statistic: 18.73 #p-value: <2e-16 #these indicate that there's a relationship between response and the predictor

```
set.seed(1)
x <- rnorm(100)
y <- 2 * x + rnorm(100)
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) Now perform a simple linear regression of x onto y without an intercept, and report the coefficient estimate, its standard error, and the corresponding t-statistic and p-values associated with the null hypothesis H0 :   = 0. Comment on these results. #coefficient estimate: 0.39111 #t-statistic: 18.73

#p-value: <2e-16 #these indicate that there's a relationship between response and the predictor

```r
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) What is the relationship between the results obtained in (a) and (b)?

#the T value and R^2 in these two models are equal; while the Coefficients and RSE are different.

(d) For the regression of Y onto X without an intercept, the t-statistic for H0 : = 0 takes the form ˆ/SE(ˆ), where ˆ is given by (3.38), and where (These formulas are slightly different from those given in Sections 3.1.1 and 3.1.2, since here we are performing regression without an intercept.) Show algebraically, and confirm numerically in R, that the t-statistic can be written as (e) Using the results from (d), argue that the t-statistic for the regression of y onto x is the same as the t-statistic for the regression of x onto y.

```
formula4 <- readPNG("8.png")
p4<-ggplot()+background_image(formula4)+theme_void()
p4
```

$$Y_i = \hat{\beta} X_i + \epsilon\_i$$

$$SE(\hat{\beta}) = \sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i\hat{\beta})^2}{(n-1)\sum_{i'=1}^{n} x_{i'}^2}}$$

$$t = \frac{\hat{\beta}}{SE(\hat{\beta})} = \frac{\hat{\beta}}{\sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i\hat{\beta})^2}{(n-1)\sum_{i'=1}^{n} x_{i'}^2}}}$$

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

$$t = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\left(\sum_{i=1}^{n} x_i^2\right)(n-1)^{-1}\sum_{i=1}^{n}\left(y_i - x_i\hat{\beta}\right)^2}}$$

$$t = \frac{\left(\sqrt{n-1}\right)\sum_{i=1}^{n} x_i y_i}{\sqrt{\left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i^2\right) - \left(\sum_{i'=1}^{n} x_{i'} y_{i'}\right)^2}}$$

(f) In R, show that when regression is performed with an intercept, the t-statistic for H0 : 1 = 0 is the same for the regression of y onto x as it is for the regression of x onto y. #the t-statistic are equal 18.556

```
lm.fit.5 <- lm(y~x)
lm.fit.6 <- lm(x~y)
summary(lm.fit.5)
```

```
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            1.99894    0.10773  18.556   <2e-16 ***
---
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:  0.7784,    Adjusted R-squared:  0.7762 
F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
summary(lm.fit.6)
```

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

Residuals:
     Min       1Q   Median       3Q      Max 
-0.90848 -0.28101  0.06274  0.24570  0.85736 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.03880    0.04266    0.91    0.365    
y            0.38942    0.02099   18.56   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4249 on 98 degrees of freedom
Multiple R-squared:  0.7784,    Adjusted R-squared:  0.7762 
F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

12. This problem involves simple linear regression without an intercept.

(a) Recall that the coefficient estimate ˆ 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?

```
formula5 <- readPNG("12.png")
p5<-ggplot()+background_image(formula5)+theme_void()
p5
```

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

$$\hat{\beta} * XY = * \frac{\sum_{i=1}^{n} y_i x_i}{\sum_{i=1}^{n} y_i^2}$$

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

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

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

(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(6)

n <- 100

x1 <- rnorm(n, mean = 50, sd = 10)

y1 <- 2 * x + rnorm(n, mean = 0, sd = 5)

(sum(x1^2)-sum(y1^2)) != 0
```

```
[1] TRUE
```

```r
lm.fit.7<- lm(x1 ~ y1 + 0)
lm.fit.8<- lm(y1 ~ x1 + 0)

coef(summary(lm.fit.7))
```

```
      Estimate Std. Error     t value  Pr(>|t|)
y1 -0.07188739   1.088089 -0.06606755 0.9474573
```

```r
coef(summary(lm.fit.8))
```

```
        Estimate  Std. Error     t value  Pr(>|t|)
x1 -0.0006132949 0.009282846 -0.06606755 0.9474573
```

(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.

```r
x2 <- rnorm(n, mean = 50, sd = 10)
y2 <- -x2
cat("Sum of squares of X: ", sum(x2^2), "\n")
```

```
Sum of squares of X:  251564.1
```

```r
cat("Sum of squares of Y: ", sum(y2^2), "\n")
```

```
Sum of squares of Y:  251564.1
```

```r
lm.fit.9<- lm(x2 ~ y2 + 0)
lm.fit.10<- lm(y2 ~ x2 + 0)

summary(lm.fit.9)
```

```
Warning in summary.lm(lm.fit.9): essentially perfect fit: summary may be
unreliable
```

```
Call:
lm(formula = x2 ~ y2 + 0)

Residuals:
       Min         1Q     Median         3Q        Max 
-1.379e-13 -3.510e-16  1.462e-15  3.590e-15  7.871e-15 

Coefficients:
     Estimate Std. Error    t value Pr(>|t|)    
y2 -1.000e+00  2.823e-17 -3.543e+16   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.416e-14 on 99 degrees of freedom
Multiple R-squared:      1,	Adjusted R-squared:      1 
F-statistic: 1.255e+33 on 1 and 99 DF,  p-value: < 2.2e-16
```

```r
summary(lm.fit.10)
```

```
Warning in summary.lm(lm.fit.10): essentially perfect fit: summary may be
unreliable


Call:
lm(formula = y2 ~ x2 + 0)

Residuals:
       Min         1Q     Median         3Q        Max 
-7.871e-15 -3.590e-15 -1.462e-15  3.510e-16  1.379e-13 

Coefficients:
     Estimate Std. Error    t value Pr(>|t|)    
x2 -1.000e+00  2.823e-17 -3.543e+16   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.416e-14 on 99 degrees of freedom
Multiple R-squared:      1,	Adjusted R-squared:      1 
F-statistic: 1.255e+33 on 1 and 99 DF,  p-value: < 2.2e-16
```

13. In this exercise you will create some simulated data and will ft simple linear regression models to it. Make sure to use set.seed(1) prior to starting part (a) to ensure consistent results.

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

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

(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.

```
EPS <- rnorm(100, 0, sqrt(0.25))
```

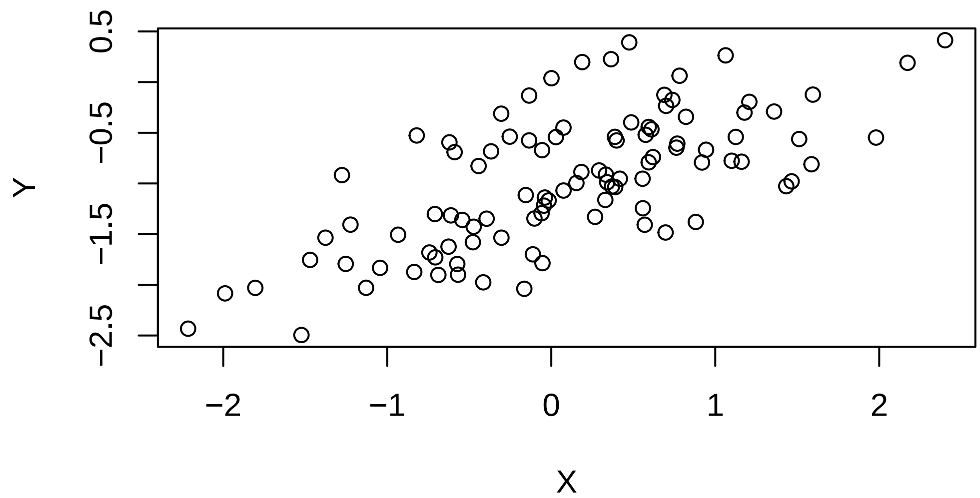
(c) Using x and eps, generate a vector y according to the model $Y = -1+0.5X +$. (3.39) What is the length of the vector y? What are the values of 0 and 1 in this linear model? #100 # 0 and 1:-1, 0.5

```
Y <- (- 1+0.5*X + EPS)
length(Y)
```

```
[1] 100
```

(d) Create a scatterplot displaying the relationship between x and y. Comment on what you observe. #there's a positive linear relationship between x and y

```
plot(Y~X)
```



(e) Fit a least squares linear model to predict y using x. Comment on the model obtained. How do ˆ0 and ˆ1 compare to 0 and 1? # ˆ1 is lower while ˆ0 is a bit higher

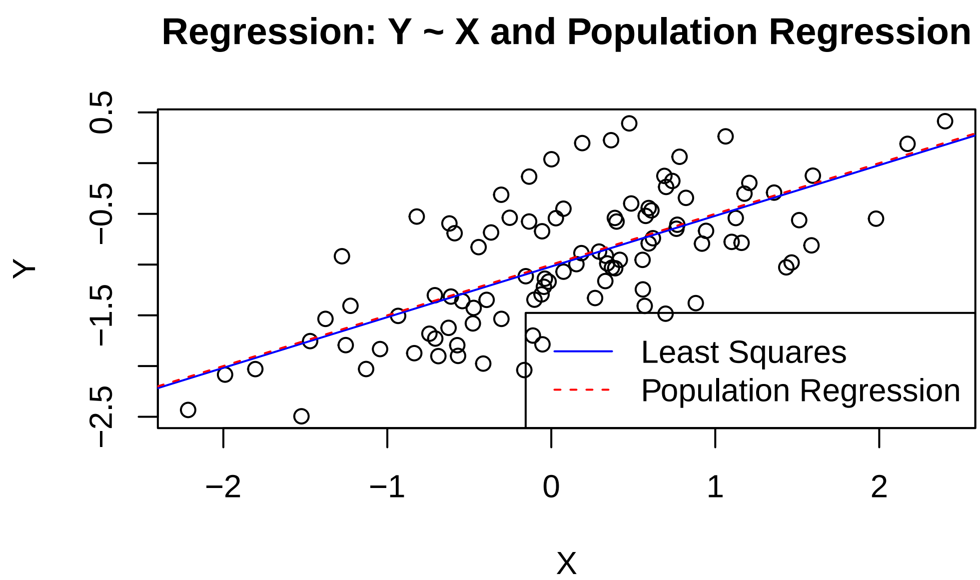
```
summary(lm(Y~X))
```

```
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) 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.

```r
plot(X, Y, main = "Regression: Y ~ X and Population Regression")

abline(lm(Y ~ X), col = "blue", lty = 1)  # Least squares line

abline(a = -1, b = 0.5, col = "red", lty = 2)  # Population regression line

# Add a legend to the plot
legend("bottomright", legend = c("Least Squares", "Population Regression"),
       col = c("blue", "red"), lty = 1:2)
```



(g) Now fit a polynomial regression model that predicts y using x and x2. Is there evidence that the quadratic term improves the model fit? Explain your answer. #not improved, because the true relationship is linear and a more flexible model can't fit better(p-value high)

```r
summary(lm(Y ~ poly(X,2)))
```

```
Call:
lm(formula = Y ~ poly(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)  -0.9645     0.0479 -20.134  < 2e-16 ***
poly(X, 2)1   4.4638     0.4790   9.319 3.97e-15 ***
poly(X, 2)2  -0.6720     0.4790  -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
```

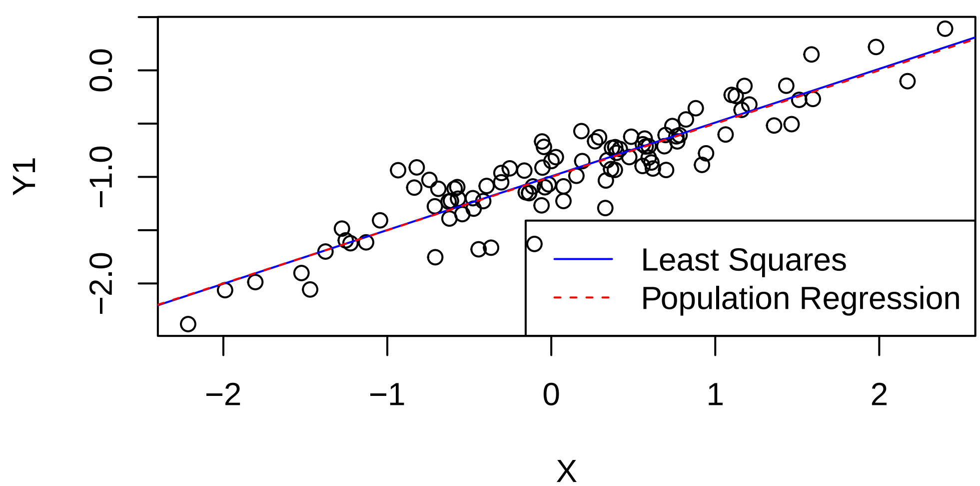
(h) Repeat (a)–(f) after modifying the data generation process in such a way that there is less noise in the data. The model (3.39) should remain the same. You can do this by decreasing the variance of the normal distribution used to generate the error term in (b). Describe your results.

```r
EPS1 <- rnorm(100, sd = 0.2)
Y1 <- (- 1+0.5*X + EPS1)
summary(lm(Y1 ~ X))
```

```
Call:
lm(formula = Y1 ~ X)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.58282 -0.09646 -0.00907  0.12985  0.52831 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.99453    0.02094  -47.49   <2e-16 ***
X            0.50423    0.02326   21.68   <2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2079 on 98 degrees of freedom
Multiple R-squared:  0.8275,	Adjusted R-squared:  0.8257 
F-statistic:   470 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
plot(X, Y1)

abline(lm(Y1 ~ X), col = "blue", lty = 1)  # Least squares line

abline(a = -1, b = 0.5, col = "red", lty = 2)  # Population regression line

# Add a legend to the plot
legend("bottomright", legend = c("Least Squares", "Population Regression"),
       col = c("blue", "red"), lty = 1:2)
```




```r
summary(lm(Y1 ~ poly(X,2)))
```

```
Call:
```

```
lm(formula = Y1 ~ poly(X, 2))

Residuals:
     Min       1Q   Median       3Q      Max 
-0.58548 -0.09905 -0.00717  0.13191  0.52908 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.93962    0.02089 -44.979   <2e-16 ***
poly(X, 2)1  4.50632    0.20890  21.571   <2e-16 ***
poly(X, 2)2 -0.03919    0.20890  -0.188    0.852    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2089 on 97 degrees of freedom
Multiple R-squared:  0.8275,    Adjusted R-squared:  0.824 
F-statistic: 232.7 on 2 and 97 DF,  p-value: < 2.2e-16
```

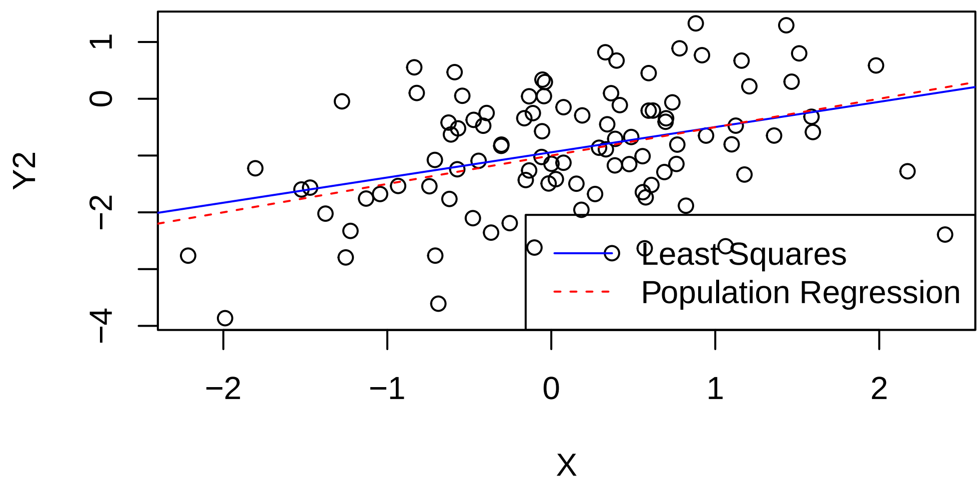
(i) Repeat (a)–(f) after modifying the data generation process in such a way that there is more noise in the data. The model (3.39) should remain the same. You can do this by increasing the variance of the normal distribution used to generate the error term in (b). Describe your results.

```r
EPS2 <- rnorm(100, sd = 1)
Y2 <- (- 1+0.5*X + EPS2)
summary(lm(Y2 ~ X))
```

```
Call:
lm(formula = Y2 ~ X)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.51626 -0.54525 -0.03776  0.67289  1.87887 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -0.9423     0.1003  -9.397 2.47e-15 ***
X             0.4443     0.1114   3.989 0.000128 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.9955 on 98 degrees of freedom
Multiple R-squared:  0.1397,    Adjusted R-squared:  0.1309
F-statistic: 15.91 on 1 and 98 DF,  p-value: 0.000128
```

```r
plot(X, Y2)

abline(lm(Y2 ~ X), col = "blue", lty = 1)  # Least squares line

abline(a = -1, b = 0.5, col = "red", lty = 2)  # Population regression line

# Add a legend to the plot
legend("bottomright", legend = c("Least Squares", "Population Regression"),
       col = c("blue", "red"), lty = 1:2)
```




```r
summary(lm(Y2 ~ poly(X,2)))
```

```
Call:
lm(formula = Y2 ~ poly(X, 2))
```

```
Residuals:
     Min       1Q   Median       3Q      Max 
-2.40143 -0.61013 -0.06042  0.69430  1.87328 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.89395    0.09767  -9.153 9.05e-15 ***
poly(X, 2)1  3.97082    0.97668   4.066 9.74e-05 ***
poly(X, 2)2 -2.14069    0.97668  -2.192   0.0308 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9767 on 97 degrees of freedom
Multiple R-squared:  0.1803,	Adjusted R-squared:  0.1634 
F-statistic: 10.67 on 2 and 97 DF,  p-value: 6.497e-05
```

(j) What are the confidence intervals for 0 and 1 based on the original data set, the noisier data set, and the less noisy data set? Comment on your results.

```r
confint(lm(Y ~ X))
```

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

```r
confint(lm(Y1 ~ X))
```

```
                 2.5 %     97.5 %
(Intercept) -1.0360826 -0.9529699
X            0.4580754  0.5503914
```

```r
confint(lm(Y2 ~ X))
```

```
                 2.5 %     97.5 %
(Intercept) -1.1413399 -0.7433293
X            0.2232721  0.6653558
```

14. This problem focuses on the collinearity problem.

(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? #coefficients: 0 = 2; 1 = 2; 2 = 0.3

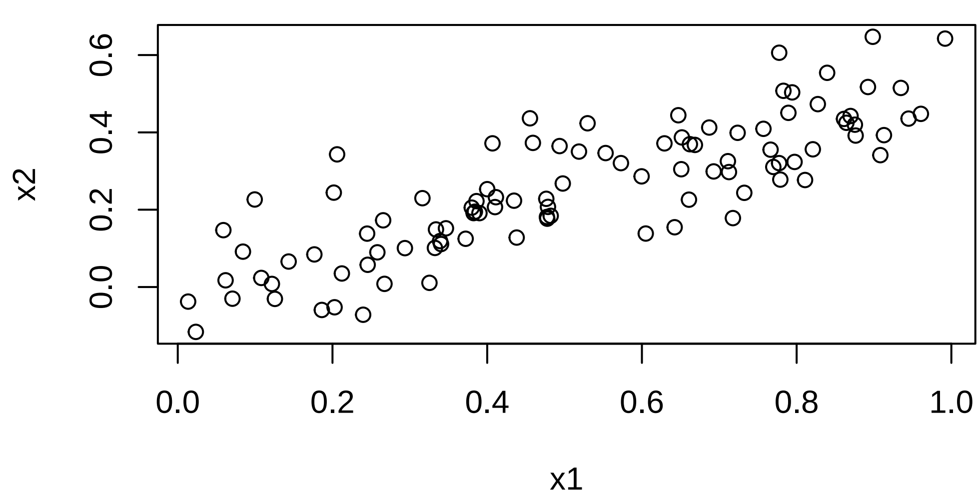
```r
set.seed(1)
x1 <- runif(100)
x2 <- 0.5 * x1 + rnorm(100) / 10
y <- 2 + 2 * x1 + 0.3 * x2 + rnorm(100)
```

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

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

```
[1] 0.8351212
```

```r
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 ˆ0, ˆ1, and ˆ2? How do these relate to the true 0, 1, and 2? Can you reject the null hypothesis H0 : 1 = 0? How about the null hypothesis H0 : 2 = 0? #coefficients: 0 = 2.1305; 1 = 1.4396; 2 = 1.0097 # there's not enough evidence to reject null hypothesis H0 : 2 = 0 #while null hypothesis H0 : 1 = 0 can be rejected

```r
summary(lm(y ~ x1 + x2))
```

```
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) Now fit a least squares regression to predict y using only x1. Comment on your results. Can you reject the null hypothesis H0 : 1 = 0? #Yes

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

```
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) Now fit a least squares regression to predict y using only x2. Comment on your results. Can you reject the null hypothesis H0 : 1 = 0? #Yes

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

```

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) Do the results obtained in (c)–(e) contradict each other? Explain your answer. #results don't contradict, because the x1 and x2 may be highly correlated, which affects the coefficient estimates.

(g) Now suppose we obtain one additional observation, which was unfortunately mismeasured. # in the model lm(y ~ x1 + x2), the x2 is insignificant before, while after obtaining one additional observation, the x1 is insignificant

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

```r
summary(lm(y ~ x1 + x2))
```

```

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
```

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

```

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
```

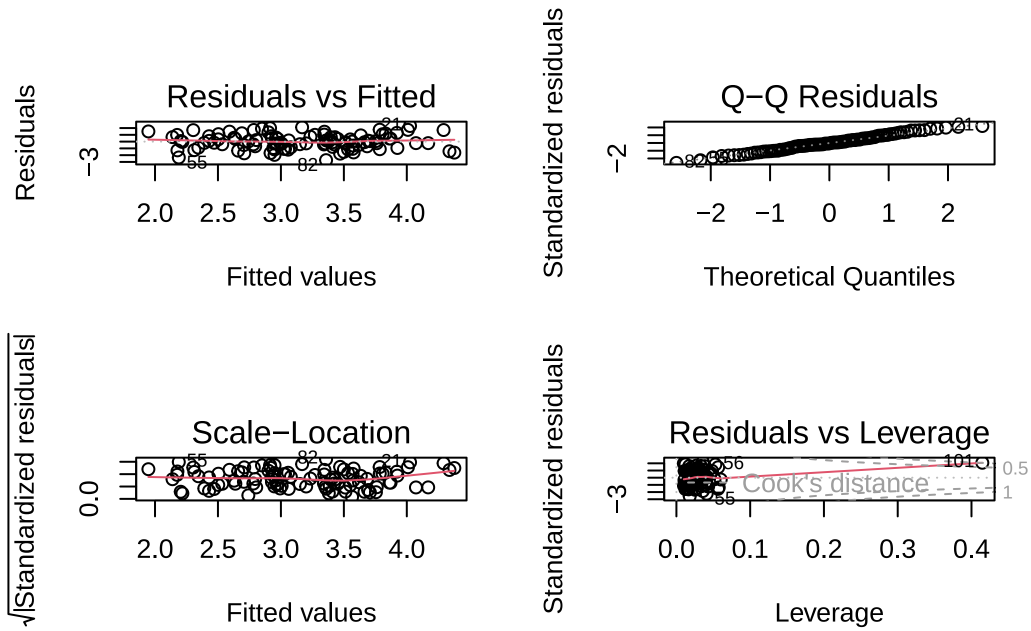
```
Multiple R-squared:  0.1562,    Adjusted R-squared:  0.1477
F-statistic: 18.33 on 1 and 99 DF,  p-value: 4.295e-05
```

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

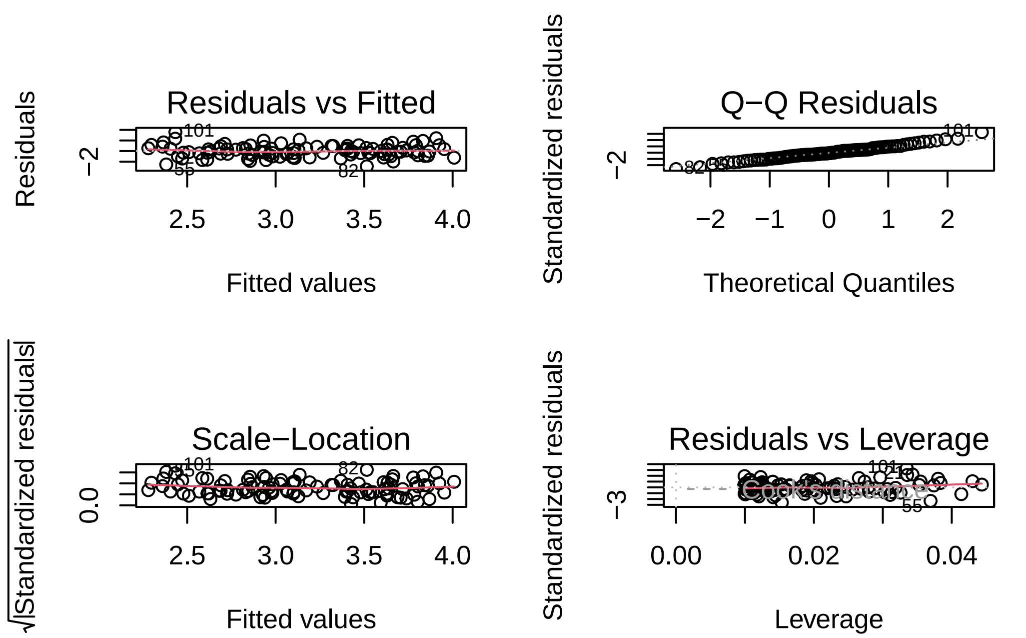
```
Call:
lm(formula = y ~ x2)

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
```

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. #on each of the models, there are both outlier and high-leverage point

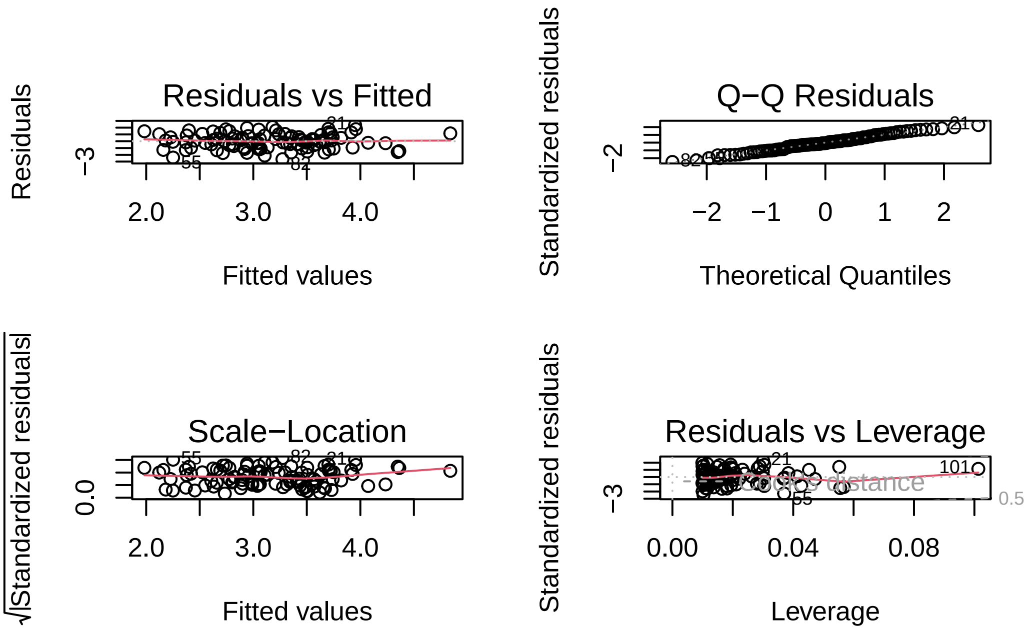
```r
par(mfrow = c(2, 2))
plot(lm(y ~ x1 + x2))
```


Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

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


Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

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



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. #only the chas is not statistically significant

```r
response <- "crim"
predictors <- setdiff(names(Boston), response)

results <-
  lapply(predictors, function(predictor) {
    model <- lm(as.formula(paste(response, "~", predictor)), data = Boston)
    summary(model)
})

names(results) <- predictors
```

```r
for (predictor in predictors) {
  res <- results[[predictor]]

  p_value <- res$coefficients[2, 4]
  cat("Predictor:", predictor, "- p-value:", p_value,
      ifelse(p_value < 0.05, "- Significant\n", "- Not Significant\n"))
}
```

```
Predictor: zn - p-value: 5.506472e-06 - Significant
Predictor: indus - p-value: 1.450349e-21 - Significant
Predictor: chas - p-value: 0.2094345 - Not Significant
Predictor: nox - p-value: 3.751739e-23 - Significant
Predictor: rm - p-value: 6.346703e-07 - Significant
Predictor: age - p-value: 2.854869e-16 - Significant
Predictor: dis - p-value: 8.519949e-19 - Significant
Predictor: rad - p-value: 2.693844e-56 - Significant
Predictor: tax - p-value: 2.357127e-47 - Significant
Predictor: ptratio - p-value: 2.942922e-11 - Significant
Predictor: lstat - p-value: 2.654277e-27 - Significant
Predictor: medv - p-value: 1.173987e-19 - Significant
```

(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 H0 : j = 0? #zn;dis;rad;medv

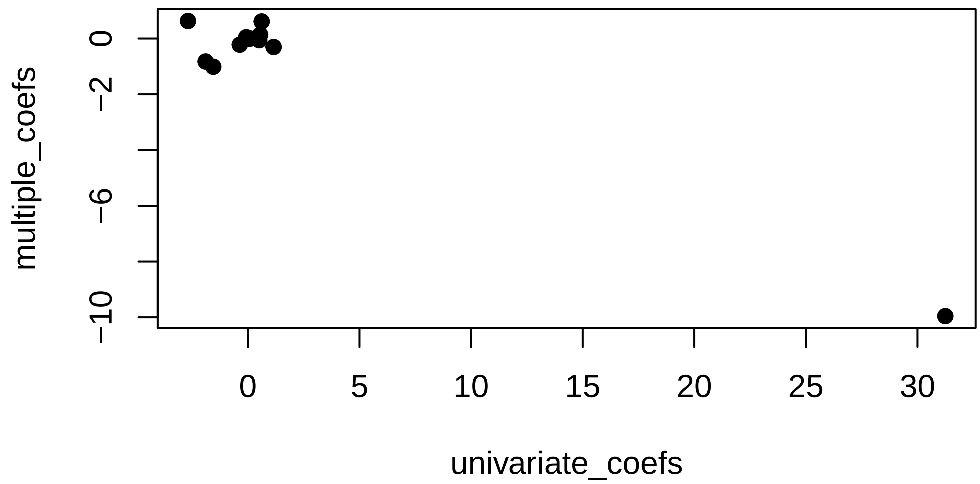
```r
lm.fit.11 <- lm(crim ~ .,data = Boston)
summary(lm.fit.11)
```

```
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
```

(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
univariate_coefs <- sapply(predictors, function(predictor) {
  res <- results[[predictor]]
  coef(res)[2]  # Extract the coefficient for the predictor
})

#coefficients(lm.fit.11) in y-axis
multiple_coefs <- coef(lm.fit.11)[-1]

plot(univariate_coefs, multiple_coefs,pch = 19)
```



(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 = 0 + 1X + 2X2 + 3X3 + . #X^3:indus;nox;age;dis;ptratio;medv

```
response <- "crim"
predictors <- setdiff(names(Boston), response)

poly_results <-
  lapply(predictors, function(predictor) {
    formula <- as.formula(paste(response, "~ poly(", predictor, ", 3, raw = TRUE)"))
    model <- lm(formula, data = Boston)
    summary(model)
})

names(poly_results) <- predictors

poly_results
```

```
$zn

Call:
lm(formula = formula, data = Boston)
```

```
Residuals:
   Min     1Q Median     3Q    Max 
-4.821 -4.614 -1.294  0.473 84.130 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)               4.846e+00  4.330e-01  11.192  < 2e-16 ***
poly(zn, 3, raw = TRUE)1 -3.322e-01  1.098e-01  -3.025  0.00261 ** 
poly(zn, 3, raw = TRUE)2  6.483e-03  3.861e-03   1.679  0.09375 .  
poly(zn, 3, raw = TRUE)3 -3.776e-05  3.139e-05  -1.203  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


$indus

Call:
lm(formula = formula, data = Boston)

Residuals:
   Min     1Q Median     3Q    Max 
-8.278 -2.514  0.054  0.764 79.713 

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  3.6625683  1.5739833   2.327   0.0204 *  
poly(indus, 3, raw = TRUE)1 -1.9652129  0.4819901  -4.077 5.30e-05 ***
poly(indus, 3, raw = TRUE)2  0.2519373  0.0393221   6.407 3.42e-10 ***
poly(indus, 3, raw = TRUE)3 -0.0069760  0.0009567  -7.292 1.20e-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


$chas
```

```
Call:
lm(formula = formula, data = Boston)

Residuals:
   Min     1Q Median     3Q    Max 
-3.738 -3.661 -3.435  0.018 85.232 

Coefficients: (2 not defined because of singularities)
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  3.7444     0.3961   9.453   <2e-16 ***
poly(chas, 3, raw = TRUE)1  -1.8928     1.5061  -1.257    0.209    
poly(chas, 3, raw = TRUE)2       NA         NA      NA       NA    
poly(chas, 3, raw = TRUE)3       NA         NA      NA       NA    
---
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


$nox

Call:
lm(formula = formula, data = Boston)

Residuals:
   Min     1Q Median     3Q    Max 
-9.110 -2.068 -0.255  0.739 78.302 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)                 233.09      33.64   6.928 1.31e-11 ***
poly(nox, 3, raw = TRUE)1 -1279.37     170.40  -7.508 2.76e-13 ***
poly(nox, 3, raw = TRUE)2  2248.54     279.90   8.033 6.81e-15 ***
poly(nox, 3, raw = TRUE)3 -1245.70     149.28  -8.345 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
```

```
$rm

Call:
lm(formula = formula, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max 
-18.485  -3.468  -2.221  -0.015  87.219 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)  
(Intercept)              112.6246    64.5172   1.746   0.0815 .
poly(rm, 3, raw = TRUE)1 -39.1501    31.3115  -1.250   0.2118  
poly(rm, 3, raw = TRUE)2   4.5509     5.0099   0.908   0.3641  
poly(rm, 3, raw = TRUE)3  -0.1745     0.2637  -0.662   0.5086  
---
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


$age

Call:
lm(formula = formula, data = Boston)

Residuals:
   Min     1Q Median     3Q    Max 
-9.762 -2.673 -0.516  0.019 82.842 

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)   
(Intercept)               -2.549e+00  2.769e+00  -0.920  0.35780   
poly(age, 3, raw = TRUE)1  2.737e-01  1.864e-01   1.468  0.14266   
poly(age, 3, raw = TRUE)2 -7.230e-03  3.637e-03  -1.988  0.04738 * 
poly(age, 3, raw = TRUE)3  5.745e-05  2.109e-05   2.724  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


$dis

Call:
lm(formula = formula, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-10.757  -2.588   0.031   1.267  76.378

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                30.0476     2.4459  12.285  < 2e-16 ***
poly(dis, 3, raw = TRUE)1 -15.5543     1.7360  -8.960  < 2e-16 ***
poly(dis, 3, raw = TRUE)2   2.4521     0.3464   7.078 4.94e-12 ***
poly(dis, 3, raw = TRUE)3  -0.1186     0.0204  -5.814 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


$rad

Call:
lm(formula = formula, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-10.381  -0.412  -0.269   0.179  76.217

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)               -0.605545   2.050108  -0.295    0.768
poly(rad, 3, raw = TRUE)1  0.512736   1.043597   0.491    0.623
poly(rad, 3, raw = TRUE)2 -0.075177   0.148543  -0.506    0.613
poly(rad, 3, raw = TRUE)3  0.003209   0.004564   0.703    0.482
```

```
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


$tax

Call:
lm(formula = formula, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-13.273  -1.389   0.046   0.536  76.950

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                1.918e+01  1.180e+01   1.626    0.105
poly(tax, 3, raw = TRUE)1 -1.533e-01  9.568e-02  -1.602    0.110
poly(tax, 3, raw = TRUE)2  3.608e-04  2.425e-04   1.488    0.137
poly(tax, 3, raw = TRUE)3 -2.204e-07  1.889e-07  -1.167    0.244

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


$ptratio

Call:
lm(formula = formula, data = Boston)

Residuals:
   Min     1Q Median     3Q    Max
-6.833 -4.146 -1.655  1.408 82.697

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                   477.18405  156.79498   3.043  0.00246 **
poly(ptratio, 3, raw = TRUE)1 -82.36054   27.64394  -2.979  0.00303 **
poly(ptratio, 3, raw = TRUE)2   4.63535    1.60832   2.882  0.00412 **
poly(ptratio, 3, raw = TRUE)3  -0.08476    0.03090  -2.743  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


$lstat

Call:
lm(formula = formula, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-15.234  -2.151  -0.486   0.066  83.353

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  1.2009656  2.0286452   0.592   0.5541
poly(lstat, 3, raw = TRUE)1 -0.4490656  0.4648911  -0.966   0.3345
poly(lstat, 3, raw = TRUE)2  0.0557794  0.0301156   1.852   0.0646 .
poly(lstat, 3, raw = TRUE)3 -0.0008574  0.0005652  -1.517   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


$medv

Call:
lm(formula = formula, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-24.427  -1.976  -0.437   0.439  73.655

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                53.1655381  3.3563105  15.840  < 2e-16 ***
poly(medv, 3, raw = TRUE)1 -5.0948305  0.4338321 -11.744  < 2e-16 ***
```

```
poly(medv, 3, raw = TRUE)2  0.1554965  0.0171904   9.046  < 2e-16 ***
poly(medv, 3, raw = TRUE)3 -0.0014901  0.0002038  -7.312 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
```

```r
for (predictor in predictors) {
  res <- poly_results[[predictor]]

  p_value_quad <- tryCatch(res$coefficients[3, 4], error = function(e) NA)
  p_value_cubic <- tryCatch(res$coefficients[4, 4], error = function(e) NA)

  cat("Predictor:", predictor, "\n")

  cat("  p-value for X^2:", p_value_quad)
  if (!is.na(p_value_quad) & p_value_quad < 0.05) {
    cat(" - Significant non-linear term (X^2)\n")
  } else {
    cat(" - Not significant (X^2)\n")
  }

  # Print p-value for X^3 and indicate if it's significant
  cat("  p-value for X^3:", p_value_cubic)
  if (!is.na(p_value_cubic) & p_value_cubic < 0.05) {
    cat(" - Significant non-linear term (X^3)\n")
  } else {
    cat(" - Not significant (X^3)\n")
  }
}
```

```
Predictor: zn
  p-value for X^2: 0.0937505 - Not significant (X^2)
  p-value for X^3: 0.2295386 - Not significant (X^3)
Predictor: indus
  p-value for X^2: 3.420187e-10 - Significant non-linear term (X^2)
  p-value for X^3: 1.196405e-12 - Significant non-linear term (X^3)
Predictor: chas
  p-value for X^2: NA - Not significant (X^2)
```

```
  p-value for X^3: NA - Not significant (X^3)
Predictor: nox
  p-value for X^2: 6.8113e-15 - Significant non-linear term (X^2)
  p-value for X^3: 6.96111e-16 - Significant non-linear term (X^3)
Predictor: rm
  p-value for X^2: 0.3641094 - Not significant (X^2)
  p-value for X^3: 0.5085751 - Not significant (X^3)
Predictor: age
  p-value for X^2: 0.04737733 - Significant non-linear term (X^2)
  p-value for X^3: 0.006679915 - Significant non-linear term (X^3)
Predictor: dis
  p-value for X^2: 4.941214e-12 - Significant non-linear term (X^2)
  p-value for X^3: 1.088832e-08 - Significant non-linear term (X^3)
Predictor: rad
  p-value for X^2: 0.6130099 - Not significant (X^2)
  p-value for X^3: 0.4823138 - Not significant (X^3)
Predictor: tax
  p-value for X^2: 0.1374682 - Not significant (X^2)
  p-value for X^3: 0.2438507 - Not significant (X^3)
Predictor: ptratio
  p-value for X^2: 0.004119552 - Significant non-linear term (X^2)
  p-value for X^3: 0.006300514 - Significant non-linear term (X^3)
Predictor: lstat
  p-value for X^2: 0.06458736 - Not significant (X^2)
  p-value for X^3: 0.1298906 - Not significant (X^3)
Predictor: medv
  p-value for X^2: 3.260523e-18 - Significant non-linear term (X^2)
  p-value for X^3: 1.04651e-12 - Significant non-linear term (X^3)
```

labs

#We will start by using the lm() function to ft a simple linear regression model, with medv as the response and lstat as the predictor.

```r
lm.fit <- lm(medv ~ lstat, data = Boston)
summary(lm.fit)
```

```
Call:
lm(formula = medv ~ lstat, data = Boston)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-15.168  -3.990  -1.318   2.034  24.500 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 34.55384    0.56263   61.41   <2e-16 ***
lstat       -0.95005    0.03873  -24.53   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.216 on 504 degrees of freedom
Multiple R-squared:  0.5441,    Adjusted R-squared:  0.5432 
F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
lm.fit$coefficients
```

```
(Intercept)       lstat 
 34.5538409  -0.9500494 
```

```
confint(lm.fit)
```

```
                2.5 %     97.5 %
(Intercept) 33.448457 35.6592247
lstat       -1.026148 -0.8739505
```

The predict() function can be used to produce confdence intervals and prediction intervals

#a predicted value of 29.80359 for medv when lstat equals 5; #95 % confdence intervals: (29.00741,30.59978) #95 % prediction intervals: (17.565675,42.04151)

```
predict(lm.fit, data.frame(lstat= c(5, 10, 15)), interval = "confidence")
```

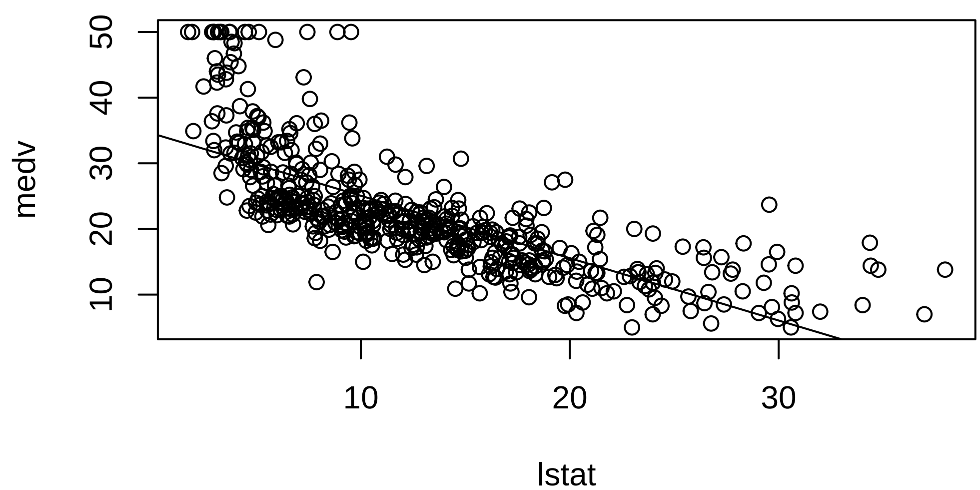
```
       fit      lwr      upr
1 29.80359 29.00741 30.59978
2 25.05335 24.47413 25.63256
3 20.30310 19.73159 20.87461
```

```r
predict(lm.fit, data.frame(lstat= c(5, 10, 15)), interval = "prediction")
```

```
       fit       lwr      upr
1 29.80359 17.565675 42.04151
2 25.05335 12.827626 37.27907
3 20.30310  8.077742 32.52846
```

least squares regression line

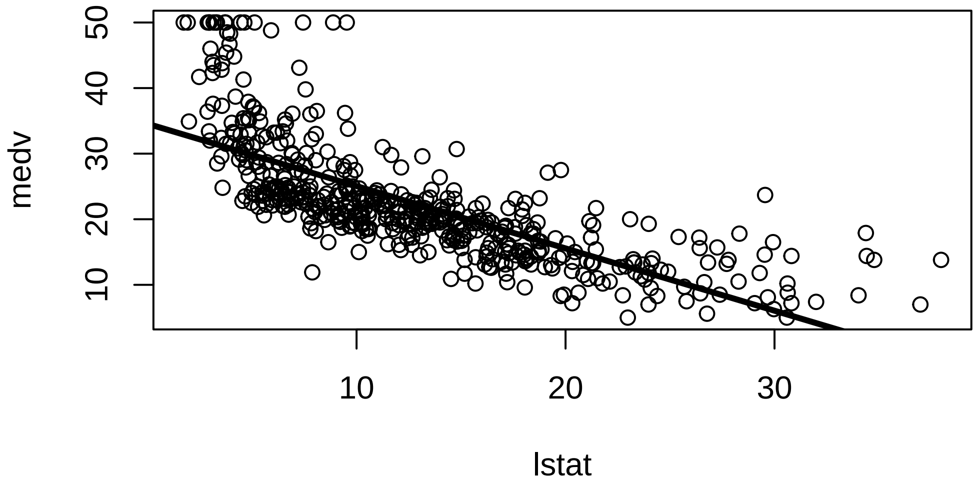
```r
attach(Boston)
plot(lstat, medv)
abline(lm.fit)
```



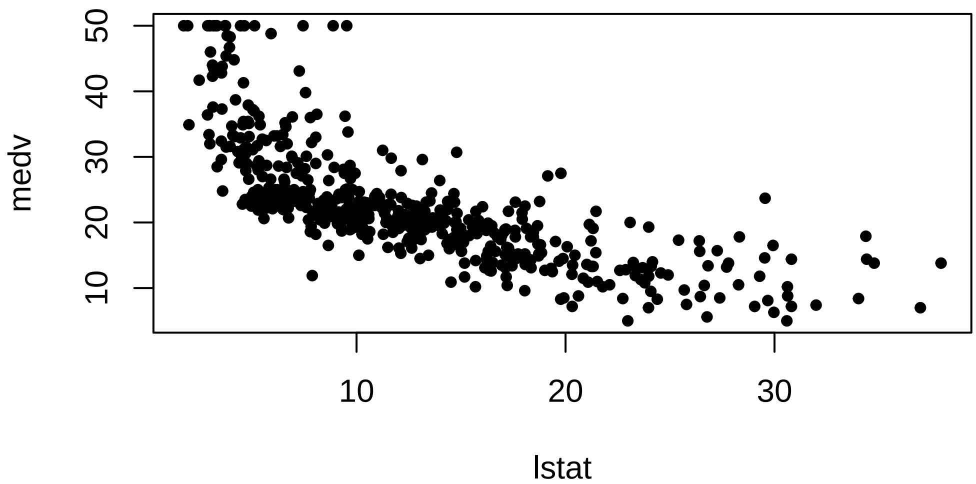
To draw a line with intercept a and slope b, we type abline(a, b).

#The lwd = 3 command causes the width of the regression line to be increased by a factor of 3

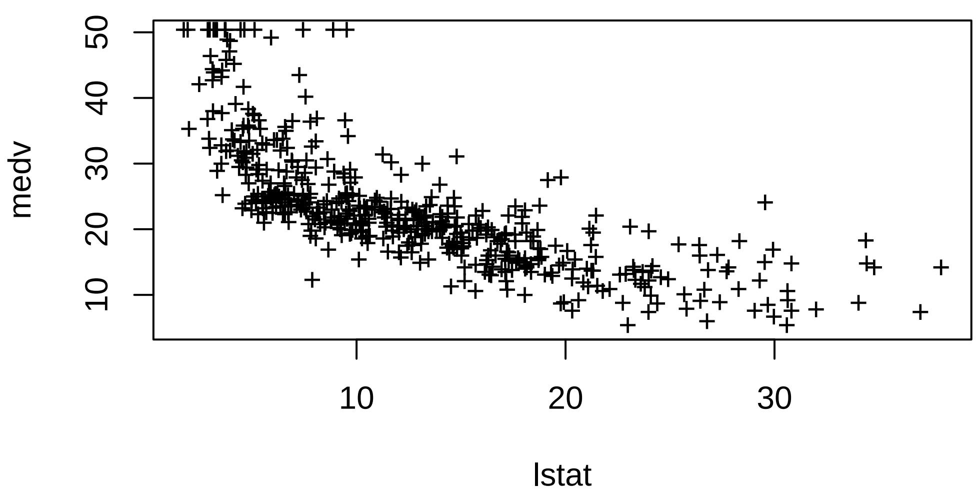
```r
plot(lstat, medv)
abline(lm.fit, lwd = 3)
```



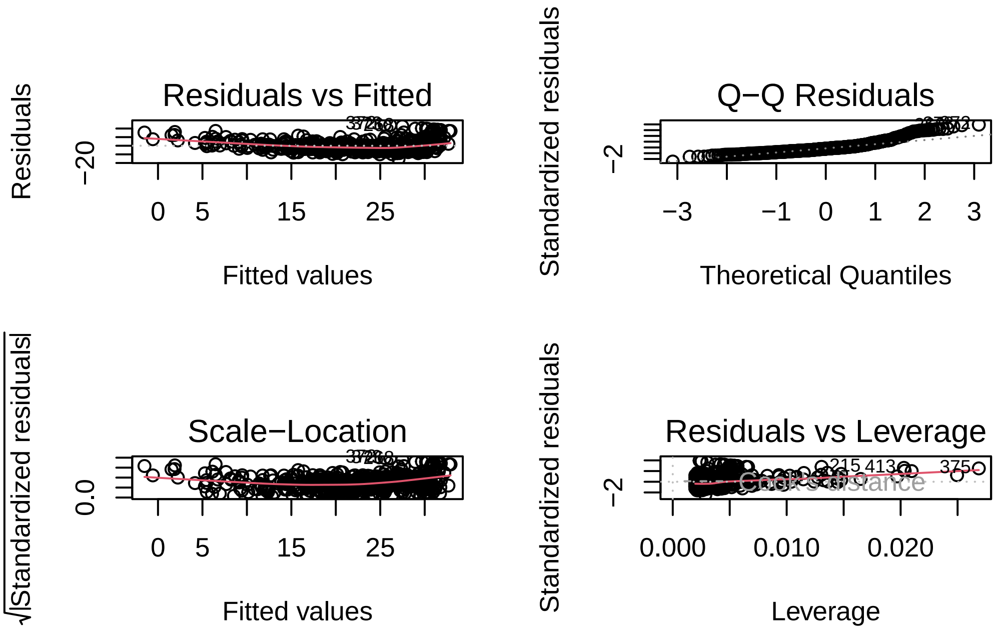

```r
plot(lstat, medv, pch = 20)
```




```
plot(lstat, medv, pch = "+")
```

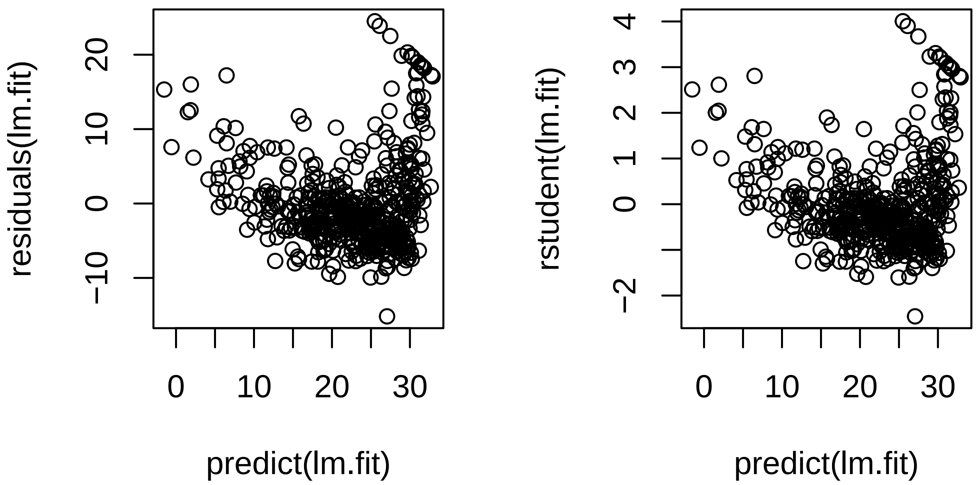


```
par(mfrow = c(2, 2))
plot(lm.fit)
```



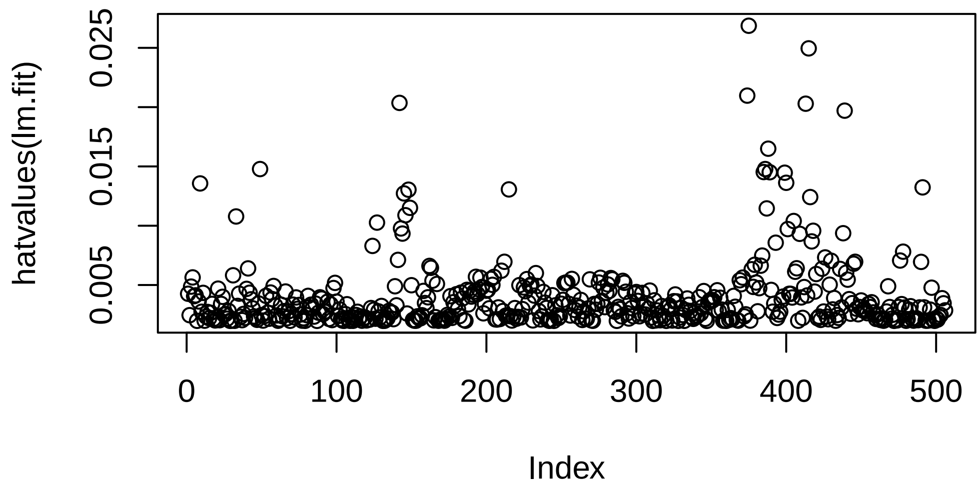
#The function rstudent() will return the studentized residuals;an outlier as well as a high leverage observation. # plot the residuals against the ftted values.

```
par(mfrow = c(1, 2))
plot(predict(lm.fit),residuals(lm.fit))
plot(predict(lm.fit),rstudent(lm.fit))
```



#On the basis of the residual plots, there is some evidence of non-linearity. #Leverage statistics can be computed for any number of predictors using the hatvalues() function. #which.max() which observation has the largest leverage statistic

```
plot(hatvalues(lm.fit))
```




```
which.max(hatvalues(lm.fit))
```

```
375
375
```

#fit a multiple linear regression model

```
lm.fit2 <- lm(medv ~ lstat + age, data = Boston)
summary(lm.fit2)
```

```
Call:
lm(formula = medv ~ lstat + age, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.981  -3.978  -1.283   1.968  23.158 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
age          0.03454    0.01223   2.826  0.00491 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.173 on 503 degrees of freedom
Multiple R-squared:  0.5513,    Adjusted R-squared:  0.5495
F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

#12 variables in Boston; the df may changed

```r
lm.fit3 <- lm(medv ~ ., data = Boston)
summary(lm.fit3)
```

```
Call:
lm(formula = medv ~ ., data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-15.1304  -2.7673  -0.5814   1.9414  26.2526

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  41.617270   4.936039   8.431 3.79e-16 ***
crim         -0.121389   0.033000  -3.678 0.000261 ***
zn            0.046963   0.013879   3.384 0.000772 ***
indus         0.013468   0.062145   0.217 0.828520
chas          2.839993   0.870007   3.264 0.001173 **
nox         -18.758022   3.851355  -4.870 1.50e-06 ***
rm            3.658119   0.420246   8.705  < 2e-16 ***
age           0.003611   0.013329   0.271 0.786595
dis          -1.490754   0.201623  -7.394 6.17e-13 ***
rad           0.289405   0.066908   4.325 1.84e-05 ***
tax          -0.012682   0.003801  -3.337 0.000912 ***
ptratio      -0.937533   0.132206  -7.091 4.63e-12 ***
lstat        -0.552019   0.050659 -10.897  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.798 on 493 degrees of freedom
```

```
Multiple R-squared:  0.7343,    Adjusted R-squared:  0.7278
F-statistic: 113.5 on 12 and 493 DF,  p-value: < 2.2e-16
```

```
summary(lm.fit3)$r.sq#R^2
```

```
[1] 0.734307
```

```
summary(lm.fit3)$sigma#RSE
```

```
[1] 4.798034
```

#vif() compute variance infation factors in car packages #quantifies the extent of correlation and collinearity among independent variables in a regression model. #diagnose collinearity problems

```
vif(lm.fit3)
```

```
    crim       zn    indus     chas      nox       rm      age      dis
1.767486 2.298459 3.987181 1.071168 4.369093 1.912532 3.088232 3.954037
     rad      tax  ptratio    lstat
7.445301 9.002158 1.797060 2.870777
```

#age has a high p value 0.958229 #Backward selection. For instance, we may stop when all remaining variables have a p-value below some threshold. #next,indus?

```
lm.fit4 <- lm(medv ~ . - age, data = Boston) #alternative:lm.fit4 <- update(lm.fit,   . - a
summary(lm.fit4)
```

```
Call:
lm(formula = medv ~ . - age, data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-15.1851  -2.7330  -0.6116   1.8555  26.3838

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  41.525128   4.919684   8.441 3.52e-16 ***
crim         -0.121426   0.032969  -3.683 0.000256 ***
zn            0.046512   0.013766   3.379 0.000785 ***
indus         0.013451   0.062086   0.217 0.828577
chas          2.852773   0.867912   3.287 0.001085 **
nox         -18.485070   3.713714  -4.978 8.91e-07 ***
rm            3.681070   0.411230   8.951  < 2e-16 ***
dis          -1.506777   0.192570  -7.825 3.12e-14 ***
rad           0.287940   0.066627   4.322 1.87e-05 ***
tax          -0.012653   0.003796  -3.333 0.000923 ***
ptratio      -0.934649   0.131653  -7.099 4.39e-12 ***
lstat        -0.547409   0.047669 -11.483  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.794 on 494 degrees of freedom
Multiple R-squared:  0.7343,    Adjusted R-squared:  0.7284
F-statistic: 124.1 on 11 and 494 DF,  p-value: < 2.2e-16
```

interaction

#the interaction term does not have a very small p-value,

```
summary(lm(medv ~ lstat * age, data = Boston))#shorthand for lstat + age + lstat:age
```

```
Call:
lm(formula = medv ~ lstat * age, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-15.806  -4.045  -1.333   2.085  27.552

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
age         -0.0007209  0.0198792  -0.036   0.9711
lstat:age    0.0041560  0.0018518   2.244   0.0252 *
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.149 on 502 degrees of freedom
Multiple R-squared:  0.5557,    Adjusted R-squared:  0.5531
F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

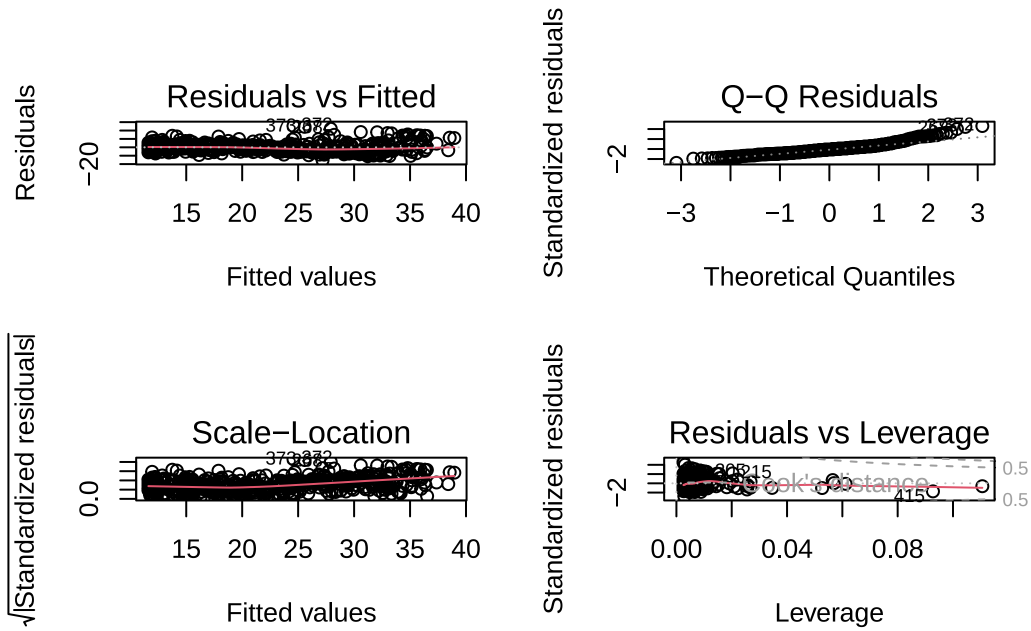
#3.6.5 Non-linear Transformations

```
#The function I() is needed since the ^ has a special meaning I() in a formula object
lm.fit5 <- lm(medv ~ lstat + I(lstat^2))
#anova() function to further quantify the extent to which the quadratic fit is superior to
anova(lm.fit, lm.fit5)
```

```
Analysis of Variance Table

Model 1: medv ~ lstat
Model 2: medv ~ lstat + I(lstat^2)
  Res.Df   RSS Df Sum of Sq     F    Pr(>F)
1    504 19472
2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#null hypothesis is that the two models fit the data equally well
#F = 135;  associated p-value near 0 ->Model 2 is much better
```

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



#high order polynomials; poly() # up to fifth order, leads to an improvement in the model fit

```
lm.fit6 <- lm(medv ~ poly(lstat, 5))
summary(lm.fit6)
```

```
Call:
lm(formula = medv ~ poly(lstat, 5))

Residuals:
     Min       1Q   Median       3Q      Max 
-13.5433  -3.1039  -0.7052   2.0844  27.1153 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       22.5328     0.2318  97.197  < 2e-16 ***
poly(lstat, 5)1 -152.4595     5.2148 -29.236  < 2e-16 ***
poly(lstat, 5)2   64.2272     5.2148  12.316  < 2e-16 ***
poly(lstat, 5)3  -27.0511     5.2148  -5.187 3.10e-07 ***
poly(lstat, 5)4   25.4517     5.2148   4.881 1.42e-06 ***
poly(lstat, 5)5  -19.2524     5.2148  -3.692 0.000247 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.215 on 500 degrees of freedom
Multiple R-squared:  0.6817,    Adjusted R-squared:  0.6785
F-statistic: 214.2 on 5 and 500 DF,  p-value: < 2.2e-16
```

#log transformation

```
summary(lm(medv ~ log(rm), data = Boston))
```

```
Call:
lm(formula = medv ~ log(rm), data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-19.487  -2.875  -0.104   2.837  39.816

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -76.488      5.028  -15.21   <2e-16 ***
log(rm)       54.055      2.739   19.73   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.915 on 504 degrees of freedom
Multiple R-squared:  0.4358,    Adjusted R-squared:  0.4347
F-statistic: 389.3 on 1 and 504 DF,  p-value: < 2.2e-16
```

#3.6.6 Qualitative Predictors: ShelveLoc # predict Sales

```
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
```

```r
lm.fit7 <- lm(Sales ~ . + Income:Advertising + Price:Age, data = Carseats)
summary(lm.fit7)
```

```
Call:
lm(formula = Sales ~ . + Income:Advertising + Price:Age, data = Carseats)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.9208 -0.7503  0.0177  0.6754  3.3413 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)         6.5755654  1.0087470   6.519 2.22e-10 ***
CompPrice           0.0929371  0.0041183  22.567  < 2e-16 ***
Income              0.0108940  0.0026044   4.183 3.57e-05 ***
Advertising         0.0702462  0.0226091   3.107 0.002030 ** 
Population          0.0001592  0.0003679   0.433 0.665330    
Price              -0.1008064  0.0074399 -13.549  < 2e-16 ***
ShelveLocGood       4.8486762  0.1528378  31.724  < 2e-16 ***
ShelveLocMedium     1.9532620  0.1257682  15.531  < 2e-16 ***
Age                -0.0579466  0.0159506  -3.633 0.000318 ***
Education          -0.0208525  0.0196131  -1.063 0.288361    
UrbanYes            0.1401597  0.1124019   1.247 0.213171    
USYes              -0.1575571  0.1489234  -1.058 0.290729    
Income:Advertising  0.0007510  0.0002784   2.698 0.007290 ** 
Price:Age           0.0001068  0.0001333   0.801 0.423812    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.011 on 386 degrees of freedom
Multiple R-squared:  0.8761,    Adjusted R-squared:  0.8719 
F-statistic:   210 on 13 and 386 DF,  p-value: < 2.2e-16
```

#contrasts() function returns the coding that R uses for the dummy variables. #The fact that the coeffcient for ShelveLocGood in the regression output is positive indicates that a good shelving location is associated with high sales (relative to a bad location).

```
contrasts(Carseats$ShelveLoc)
```

```
       Good Medium
Bad       0      0
Good      1      0
Medium    0      1
```