HW2

Aidan Horn

3/30/2022

Section 3.7, page 120, question 3

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 fit the model, and get βˆ0 = 50, βˆ1 = 20, βˆ2 = 0.07, βˆ3 = 35, βˆ4 = 0.01, βˆ5 = −10.

a. Which answer is correct, and why?

- i. For a fixed value of IQ and GPA, high school graduates earnmore, on average, than college graduates.
- 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.

The regression line is given by

$$\hat{y} = 50 + 20GPA + 0.07IQ + 35Level + 0.01GPA \times IQ - 10GPA \times Level$$

fixing IQ and GPA, for a highschool graduate, the regression line is

$$\hat{y}_0 = 50 + 20GPA + 0.07IQ + 0.01GPA \times IQ$$

and for a college graduate, the regression line is

$$\hat{y}_1 = 85 + 10GPA + 0.07IQ + 0.01GPA \times IQ$$

The starting salary for college graduate is higher if $\hat{y}_1 > \hat{y}_0$. Substituting and simplifying we get $85 + 10GPA > 50 + 20GPA$ which is equivalent to $GPA < 3.5$ so college graduates only earn more with IQ and GPA fixed if GPA is less than 3.5, therefore statement iii is true

Section 3.7, page 121, question 5

5. Consider the fitted values that result from performing linear regression without an intercept.In this setting, the i-th fitted value takes the form $\hat{y_i} = x_i\hat{\beta}$, where

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

Show that we can write

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

By substitution

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

since xi does not depend on xi', the sumation with respect to i' can be treated as a constant. Therefore we can factorize yi as the dividend of the i' summation, and write it in terms of i'

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

Finally we subsitute the fraction in the summation for ai' and we get our original formula

$$= \sum_{i=1}^{n} a_{i'} y_{i'}$$

Therefore

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

Section 3.7, page 121, question 6

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

The equation for the regression line is $y = \beta_0 + \beta_1 x$ Using the equation for (3.4)
$\hat{\beta}0 = \bar{y} - \hat{\beta}1\bar{x}, \bar{x} = -(\hat{\beta}0 - \bar{y})/\hat{\beta}1$ **If we substitute $\bar{x}$ for x, we get y = B0 + B1$\bar{x}$, the B1 terms cancel out so we are left with : y = B0 – (B0 - $\bar{y}$) which simplifies to y = $\bar{y}$. Therefore, we know $\bar{x}$, $\bar{y}$ is on the regression line**

Section 3.7, page 121, question 13 (do part (j) using cross-validation)

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.

```
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, sd = sqrt(0.25))
```

c. Using x and eps, generate a vector y according to the model

$$Y = -1 + 0.5X + \varepsilon$$

What is the length of the vector “y” ? What are the values of β0 and β1 in this linear model ?

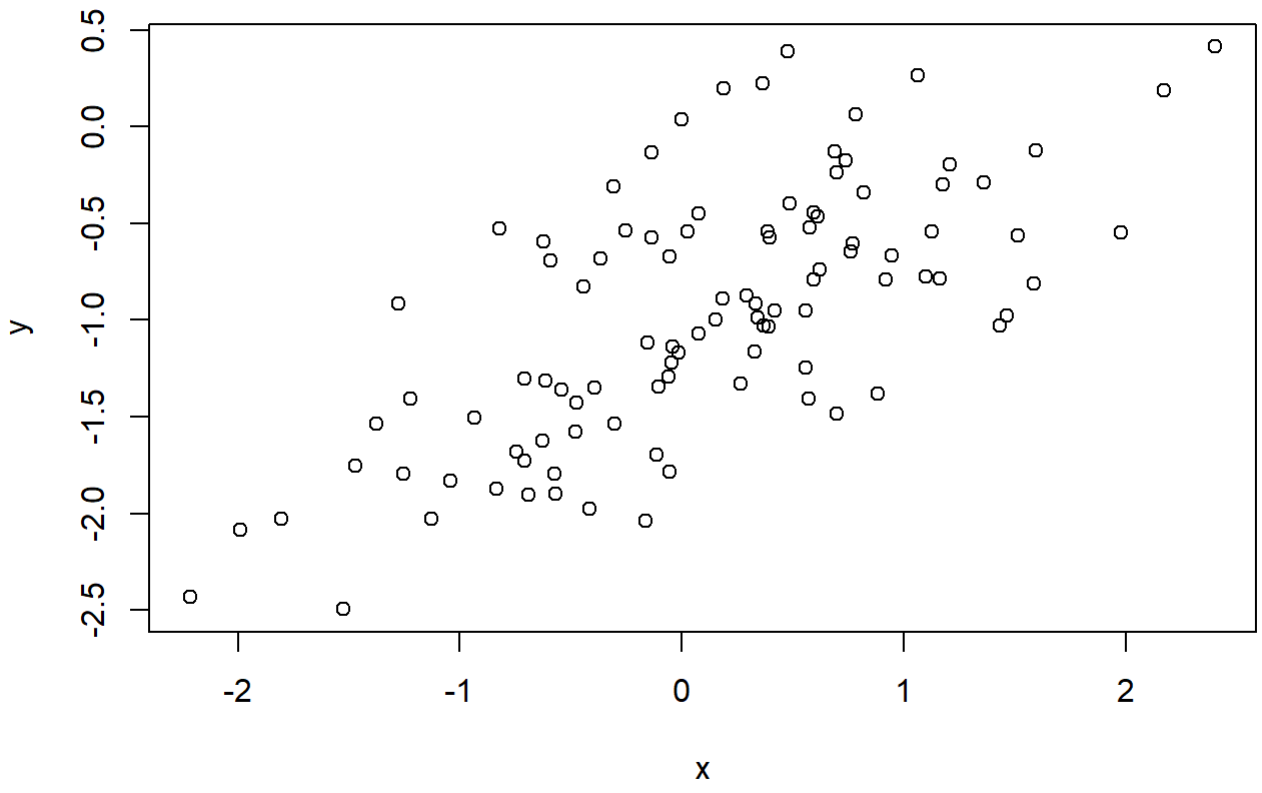
```
y = -1 + 0.5 * x + eps

length(y)
```

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

The values of Beta 0 and Beta 1 are -1 and 0.5.

d. Create a scatterplot displaying the relationship between “x” and “y”. Comment on what you observe.

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



The relationship between x and y looks linear with some noise from the epsilon variable

e. Fit a least squares linear model to predict “y” using “x”. Comment on the model obtained. How do β^0 and β^1 compare to β0 and β1 ?

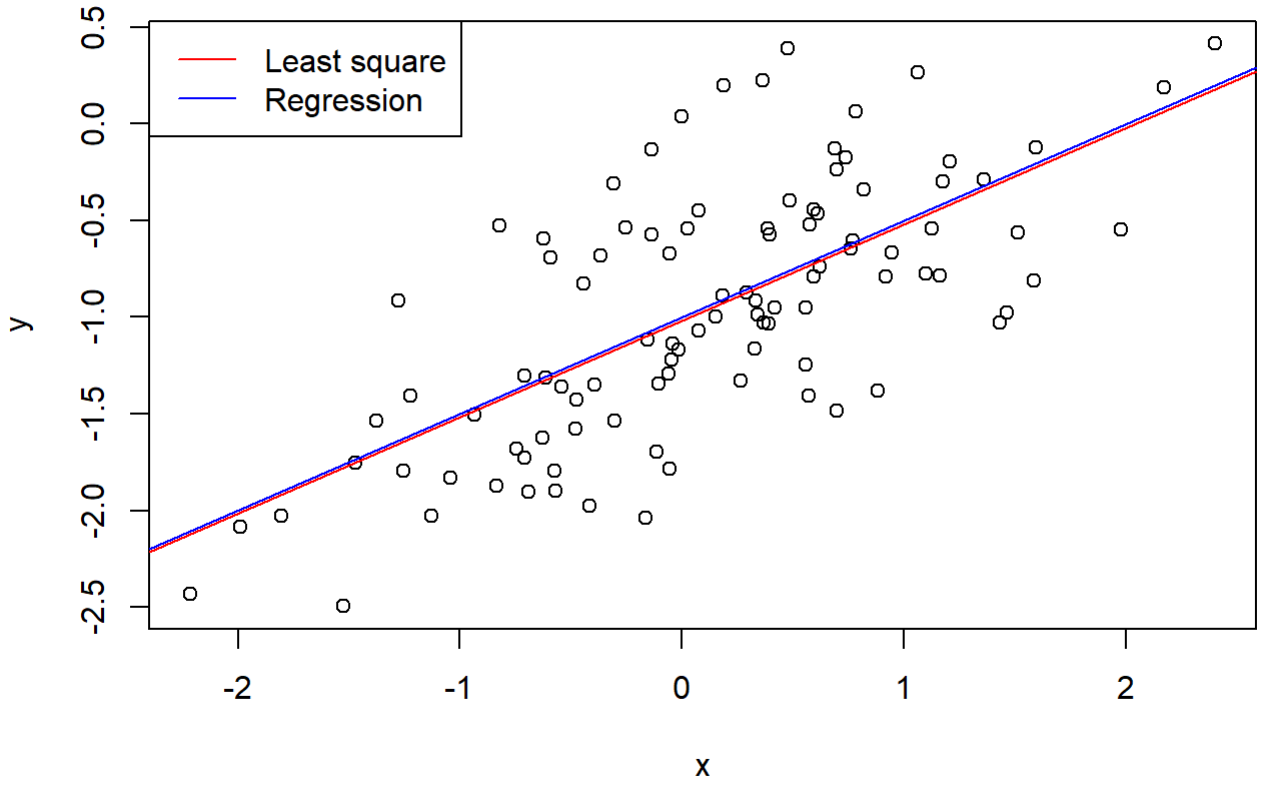
```
fit = lm(y ~ x)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93842 -0.30688 -0.06975  0.26970  1.17309
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01885    0.04849 -21.010  < 2e-16 ***
## x            0.49947    0.05386   9.273 4.58e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4814 on 98 degrees of freedom
## Multiple R-squared:  0.4674, Adjusted R-squared:  0.4619
## F-statistic: 85.99 on 1 and 98 DF,  p-value: 4.583e-15
```

the estimates for β^0 and β^1 are very close to the real values, and the p value is below the threshold (alpha = 0.05) so we are confident in rejecting the null hypothesis.

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() function to create an appropriate legend.

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



(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.

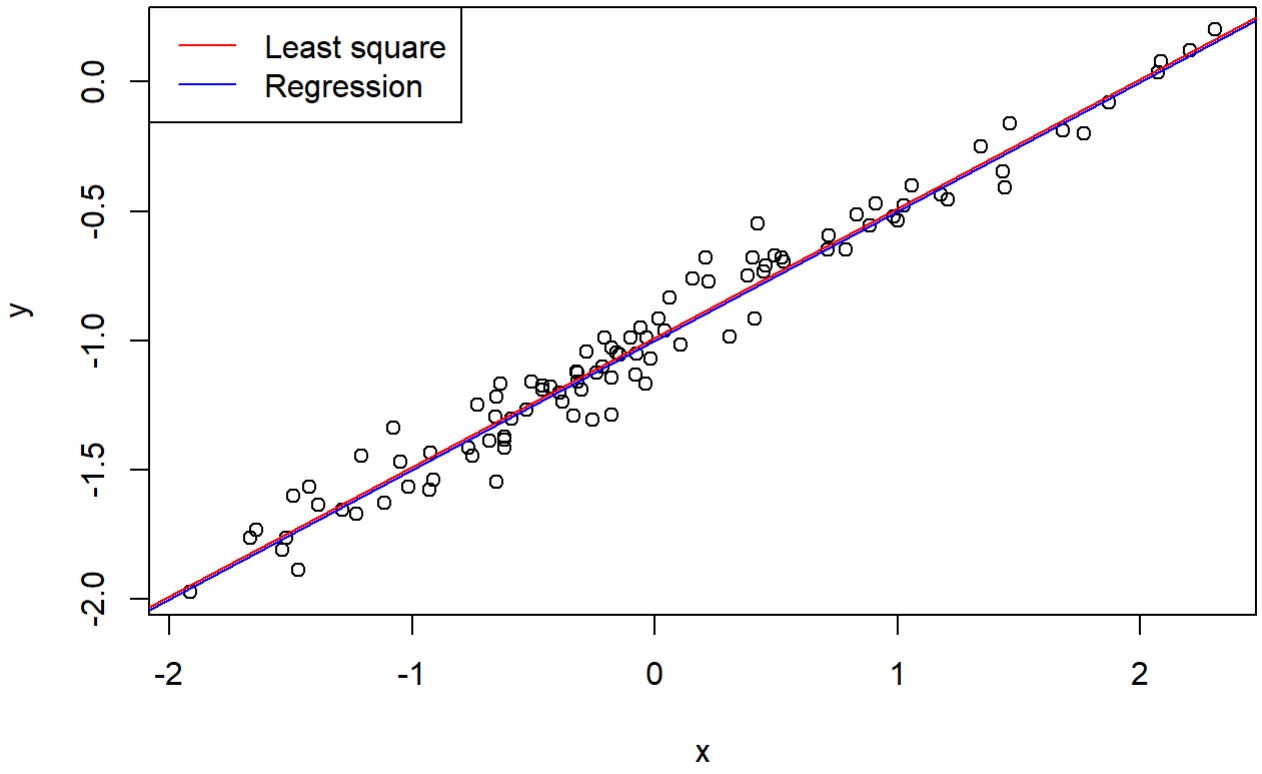
```
fit2 <- lm(y ~ x + I(x^2))
summary(fit2)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98252 -0.31270 -0.06441  0.29014  1.13500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.97164    0.05883 -16.517  < 2e-16 ***
## x            0.50858    0.05399   9.420  2.4e-15 ***
## I(x^2)      -0.05946    0.04238  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.479 on 97 degrees of freedom
## Multiple R-squared:  0.4779, Adjusted R-squared:  0.4672
## F-statistic:  44.4 on 2 and 97 DF,  p-value: 2.038e-14
```

The x^2 coefficient is not significant as its p-value is larger than the alpha of 0.05. There is not sufficient evidence that the quadratic term improves the model fit, even though the R squared is slightly better

h. Repeat (a)-(f) after modifying the data generation process in such a way that there is less noise in the data. The initial model should remain the same. Describe your results.

```
set.seed(1)
eps = rnorm(100, sd = 0.1)
x = rnorm(100)
y = -1 + 0.5 * x + eps
plot(x,y)
fit3 = lm(y ~ x)
abline(fit3, col = "red")
abline(-1, 0.5, col = "blue")
legend("topleft", c("Least square", "Regression"), col = c("red", "blue"), lty = c(1, 1))
```



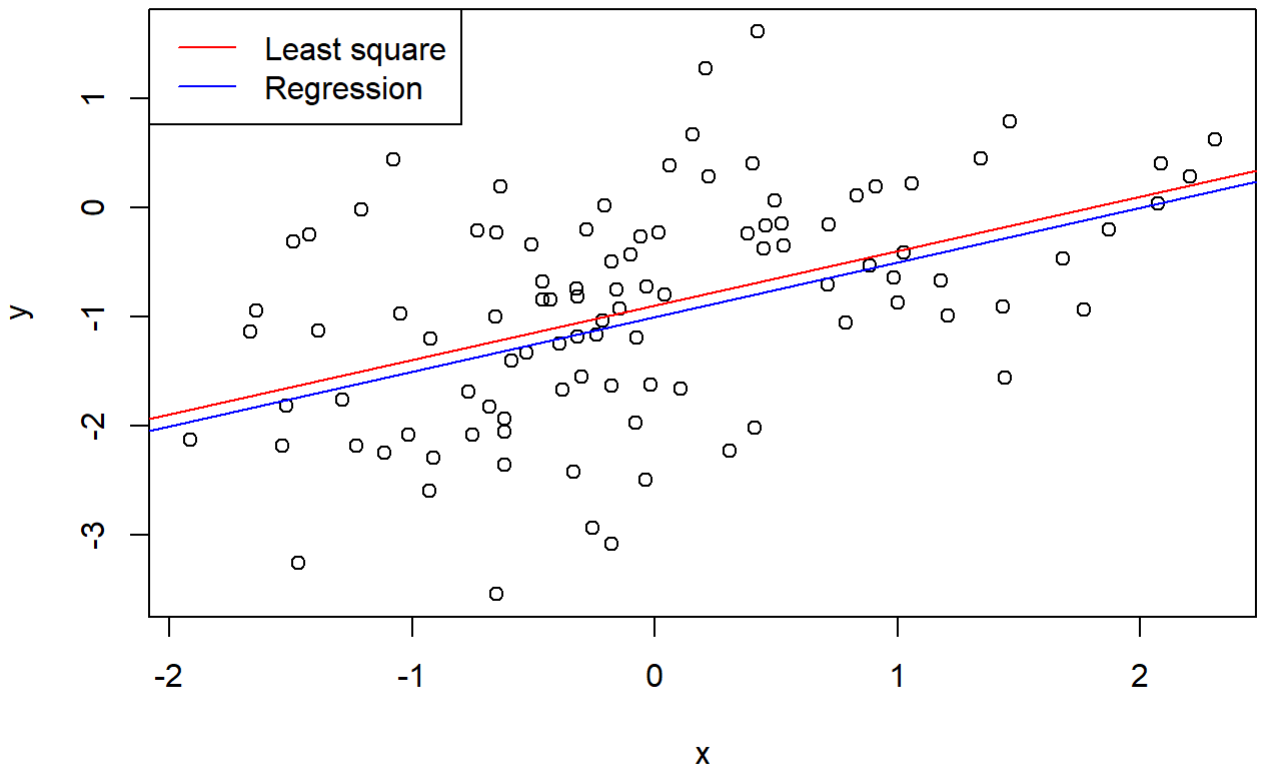

```
summary(fit3)
```

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

The data is less noisy since we reduced the standard deviation (i.e. the spread of the distribution) of the error term. The r squared, the proportion of variance explained by x, is nealy 100%, and the regression line and least square line are nealy identical

i. Repeat (a)-(f) after modifying the data generation process in such a way that there is more noise in the data. The initial model should remain the same. Describe your results.

```
set.seed(1)
eps = rnorm(100, sd = 1)
x = rnorm(100)
y = -1 + 0.5 * x + eps
plot(x,y)
fit4 = lm(y ~ x)
abline(fit4, col = "red")
abline(-1, 0.5, col = "blue")
legend("topleft", c("Least square", "Regression"), col = c("red", "blue"), lty = c(1, 1))
```




```
summary(fit4)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.32416 -0.60361  0.00536  0.58305  2.29316
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.89115    0.09035  -9.864 2.39e-16 ***
## x            0.49907    0.09472   5.269 8.16e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9028 on 98 degrees of freedom
## Multiple R-squared:  0.2207, Adjusted R-squared:  0.2128
## F-statistic: 27.76 on 1 and 98 DF,  p-value: 8.158e-07
```

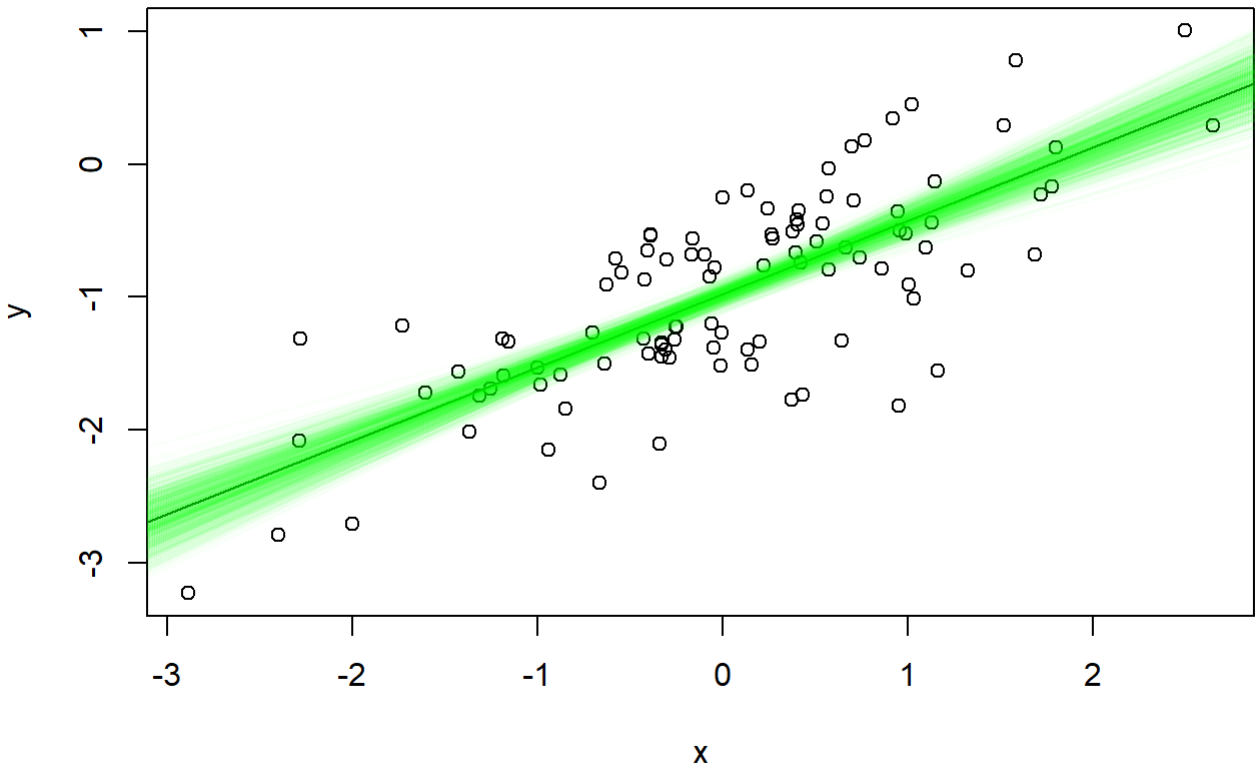
We increased the noise by increasing the variance of the normal distribution used to generate the error term. We may see that the coefficients are again very close to the previous ones, but now, as the relationship is not quite linear, we have a much lower R2 and much higher RSE. Moreover, the two lines

are wider apart but are still really close to each other as we have a fairly large data set.

j. What are the confidence intervals for β0 and β1 based on the original data set, the noisier data set, and the less noisy data set ? Comment on your results.

Original

```
x = rnorm(100)
eps = rnorm(100, sd = sqrt(0.25))
y = -1 + 0.5 * x + eps

origFit = lm(y ~ x)
plot(x,y)
abline(origFit)

beta0 = NULL
beta1 = NULL

# Bootstrap
for(i in 1:1000) {
  mySample = sample(length(x),length(x),replace=TRUE)
  myFit = lm(y[mySample] ~ x[mySample])
  abline(myFit,col = rgb(0,1,0,0.02))
  beta0[i] = coef(myFit)[1]
  beta1[i] = coef(myFit)[2]
}
```



```
quantile(beta0, prob=c(0.025, 0.975))
```

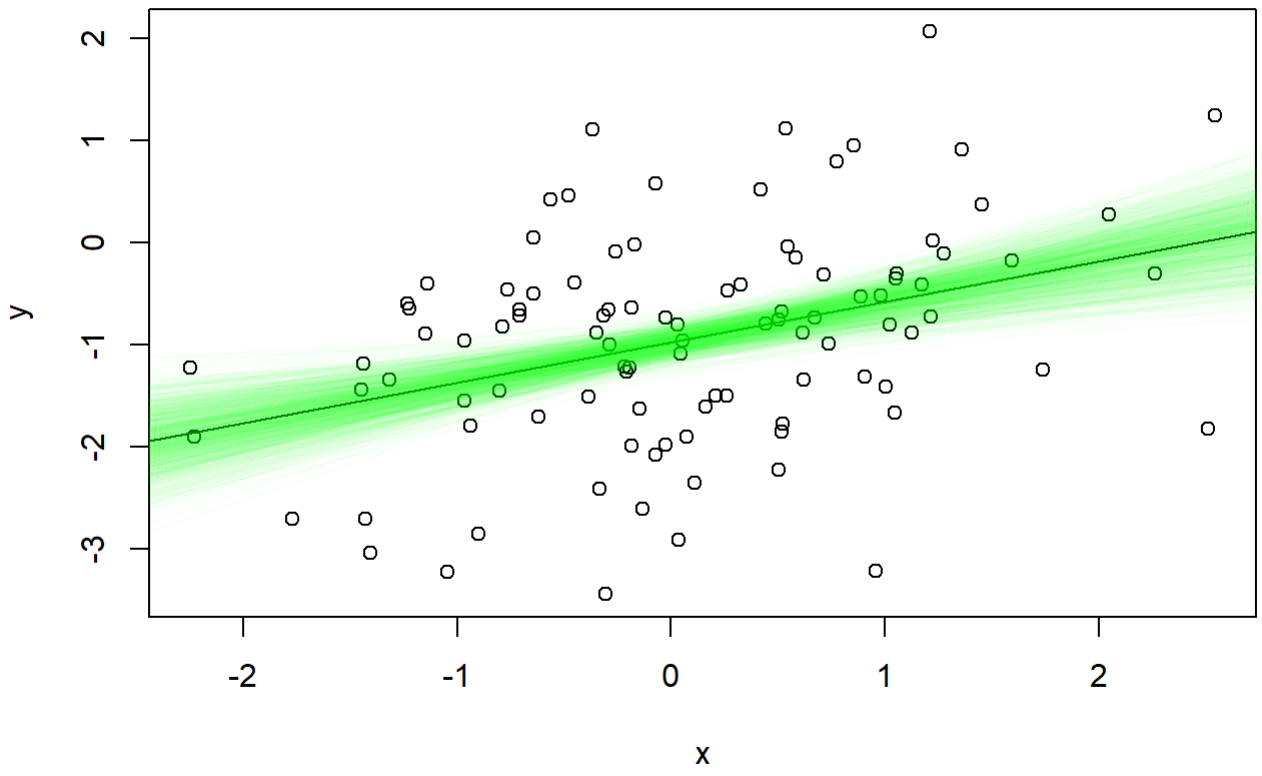
```
##       2.5%      97.5%
## -1.0720627 -0.8789268
```

```
quantile(beta1, prob=c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.4501213 0.6493910
```

noisier data

```
x = rnorm(100)
eps = rnorm(100, sd = 1)
y = -1 + 0.5 * x + eps

origFit = lm(y ~ x)
plot(x,y)
abline(origFit)

beta0 = NULL
beta1 = NULL

# Bootstrap
for(i in 1:1000) {
  mySample = sample(length(x),length(x),replace=TRUE)
  myFit = lm(y[mySample] ~ x[mySample])
  abline(myFit,col = rgb(0,1,0,0.02))
  beta0[i] = coef(myFit)[1]
  beta1[i] = coef(myFit)[2]
}
```




```
quantile(beta0, prob=c(0.025, 0.975))
```

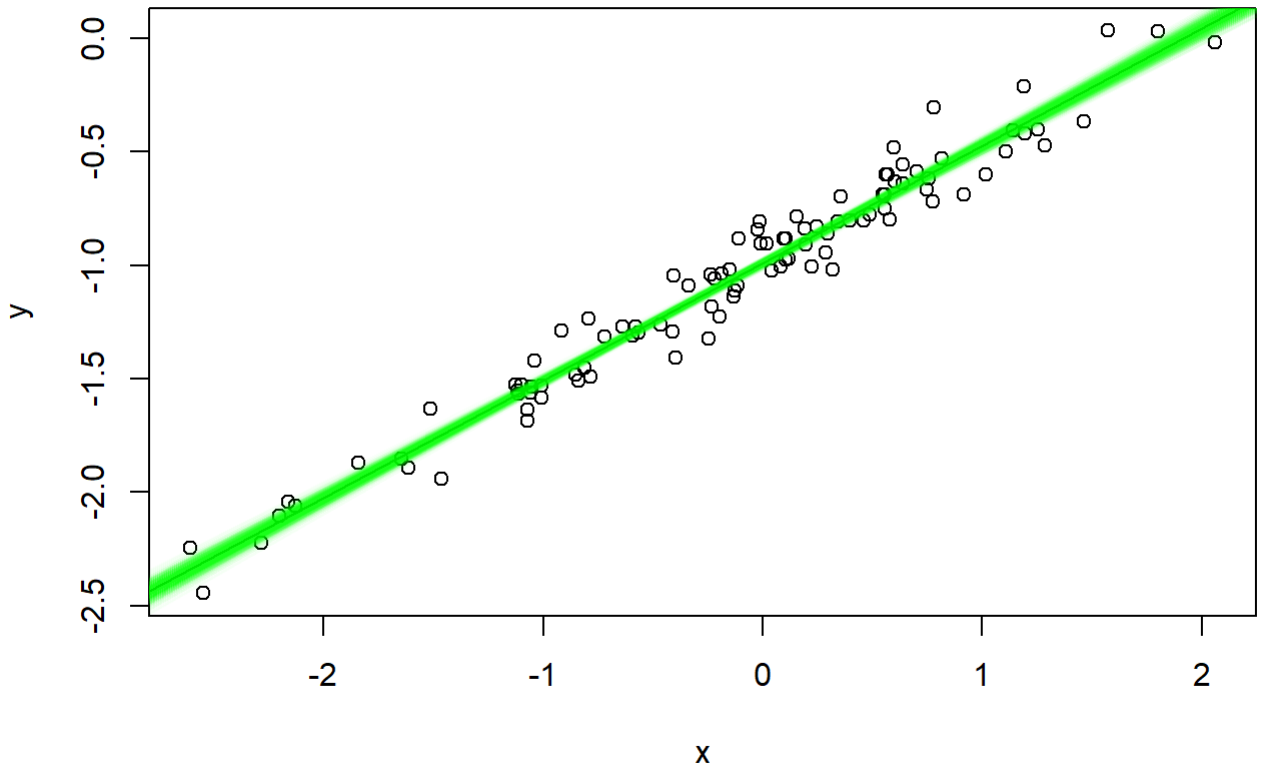
```
##       2.5%      97.5%
## -1.1772095 -0.7855965
```

```
quantile(beta1, prob=c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.1939098 0.5914576
```

less noisy data

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

origFit = lm(y ~ x)
plot(x,y)
abline(origFit)

beta0 = NULL
beta1 = NULL

# Bootstrap
for(i in 1:1000) {
  mySample = sample(length(x),length(x),replace=TRUE)
  myFit = lm(y[mySample] ~ x[mySample])
  abline(myFit,col = rgb(0,1,0,0.02))
  beta0[i] = coef(myFit)[1]
  beta1[i] = coef(myFit)[2]
}
```




```
quantile(beta0, prob=c(0.025, 0.975))
```

```
##      2.5%     97.5% 
## -1.010055 -0.968992
```

```
quantile(beta1, prob=c(0.025, 0.975))
```

```
##      2.5%     97.5% 
## 0.4970884 0.5396571
```

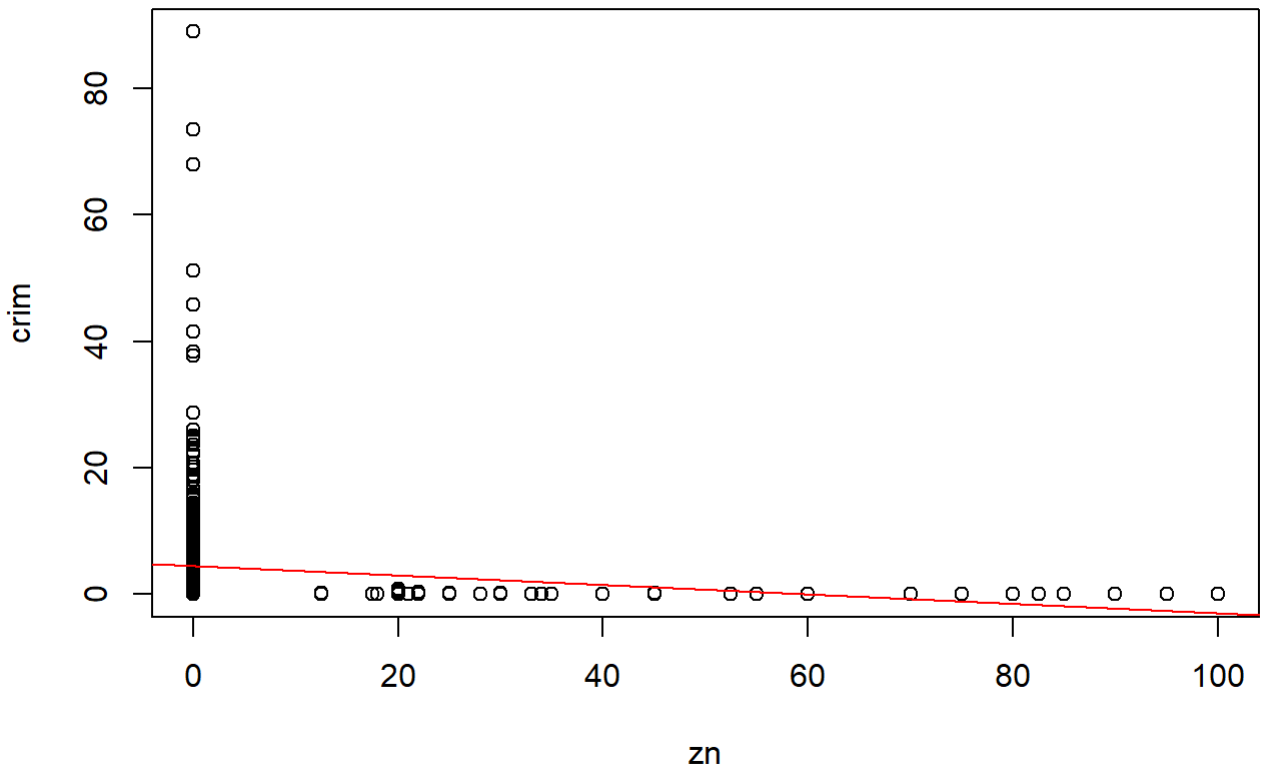
These confidence intervals conform with our expectation, as the data gets noisier, the confidence interval grows, while as it becomes less noisy, the confidence interval is narrower. The intervals are still centered on the true values of beta 0 and beta 1, but are less predicatble as the noise increases.

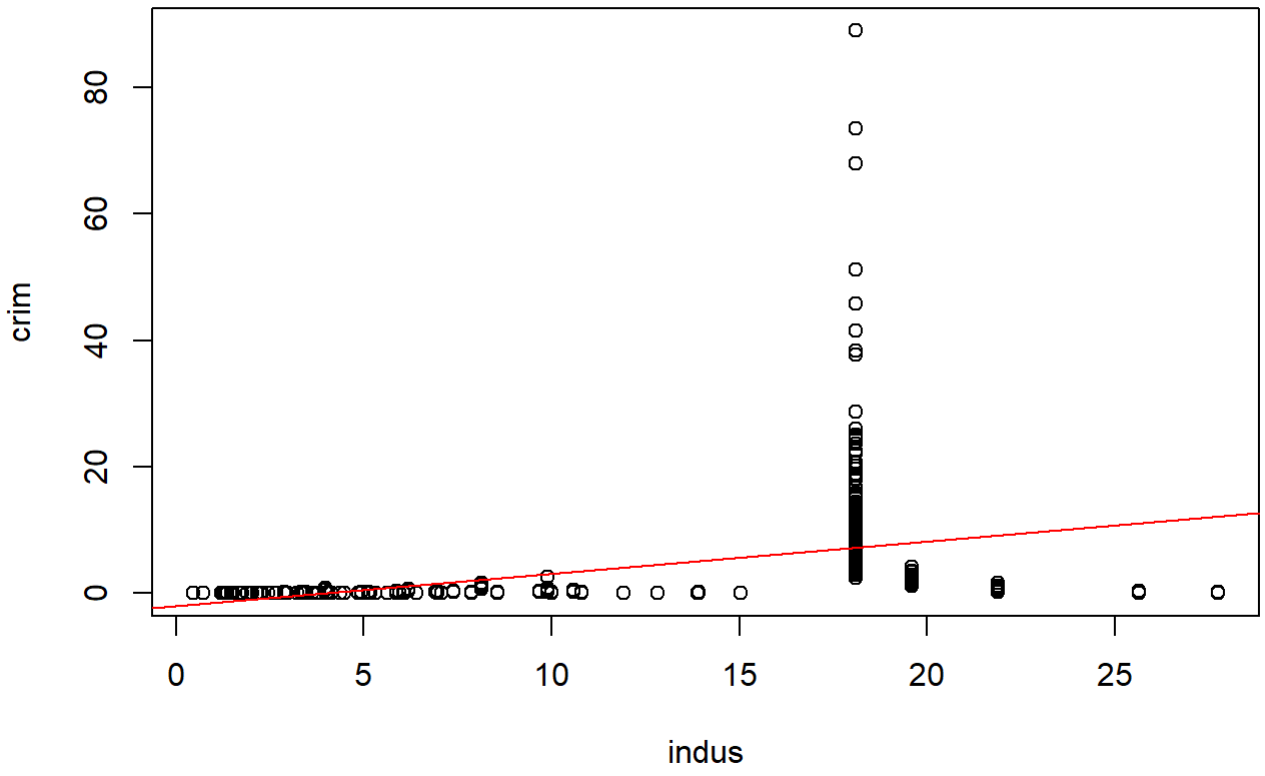
Section 3.7, page 126, question 15 part a and d

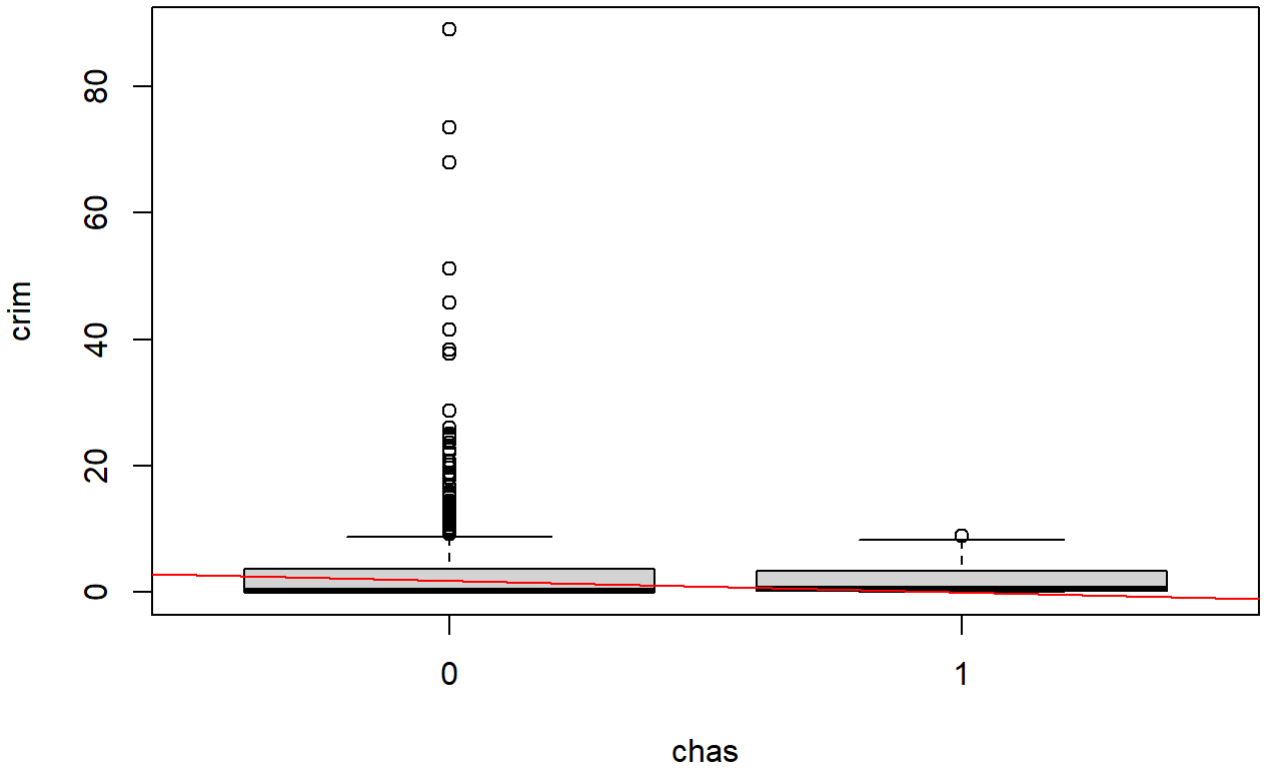
15. This problem involves the Boston data set, which we saw in the lab for this chapter. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.

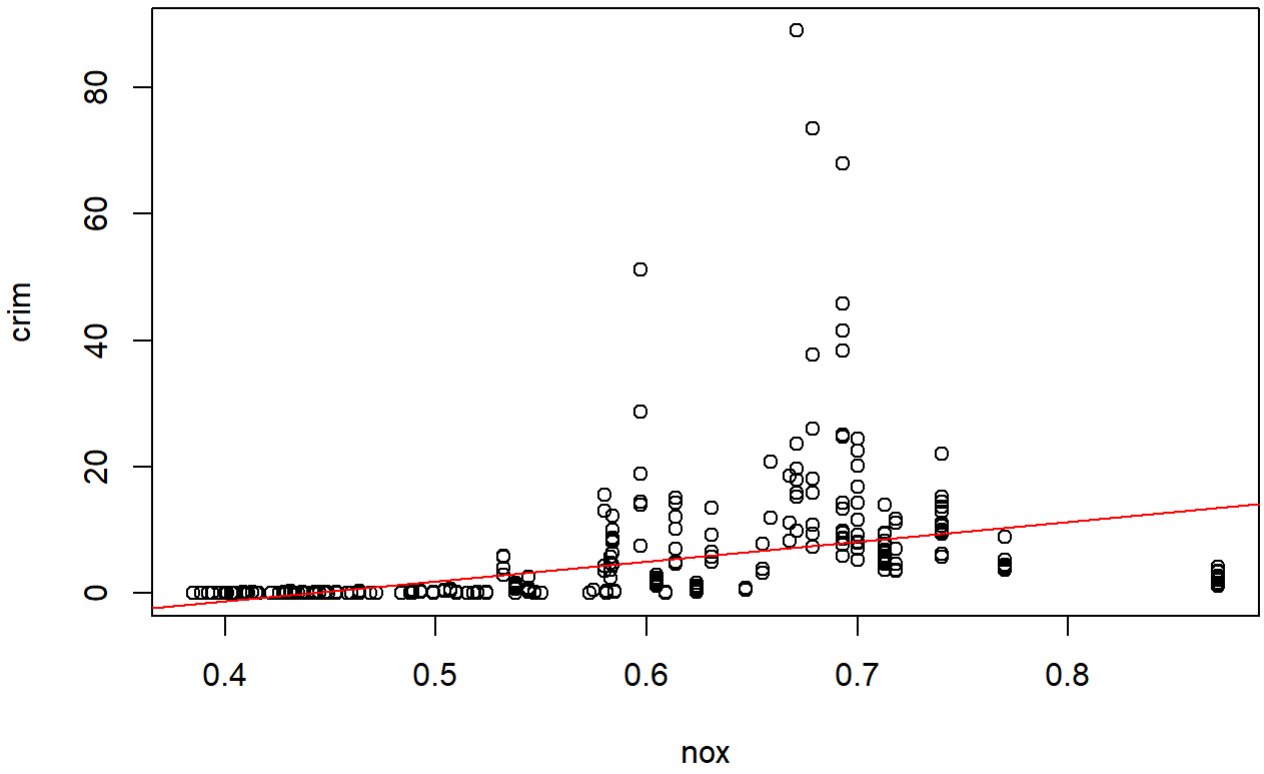
    a. For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions.

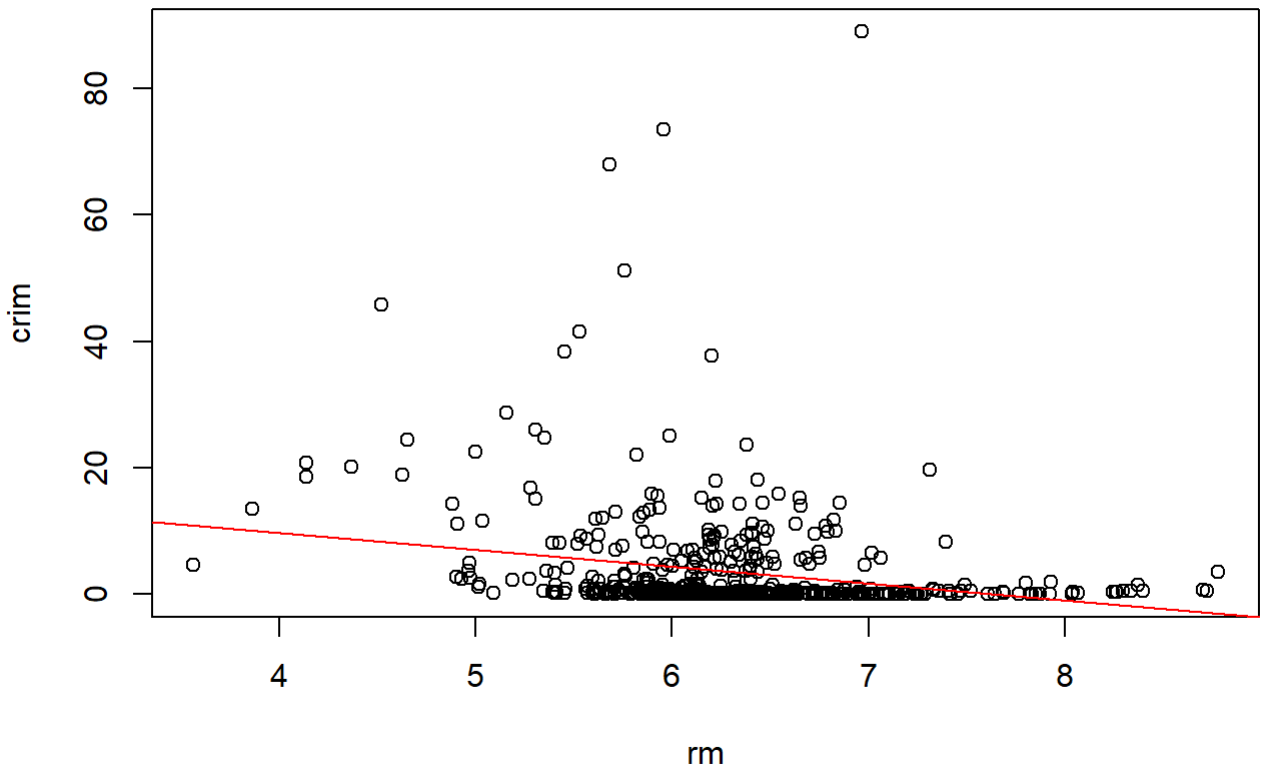
```
Boston = MASS::Boston
Boston$chas = as.factor(Boston$chas)

r2 = c()
p_value = c()

y = Boston$crim

for (i in 2:ncol(Boston)) {

  x = Boston[ ,i]

  m = lm(y ~ x)

  plot(x,y, xlab = names(Boston)[i], ylab = "crim")
  abline(m, col = "red")

  r2[i] = summary(m)$r.squared
  p_value[i] = summary(m)$coefficients[2,4]

}
```

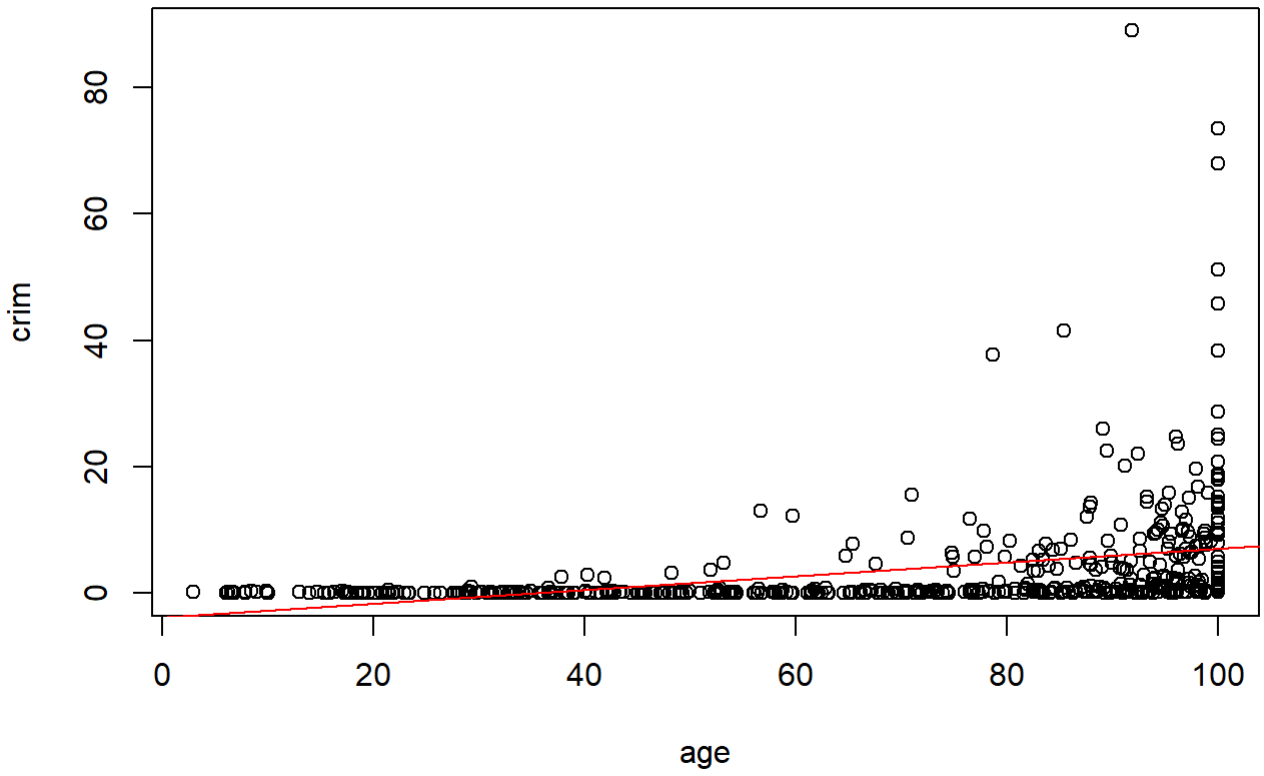

crim
0
20
40
60
80
0
20
40
60
80
100
zn

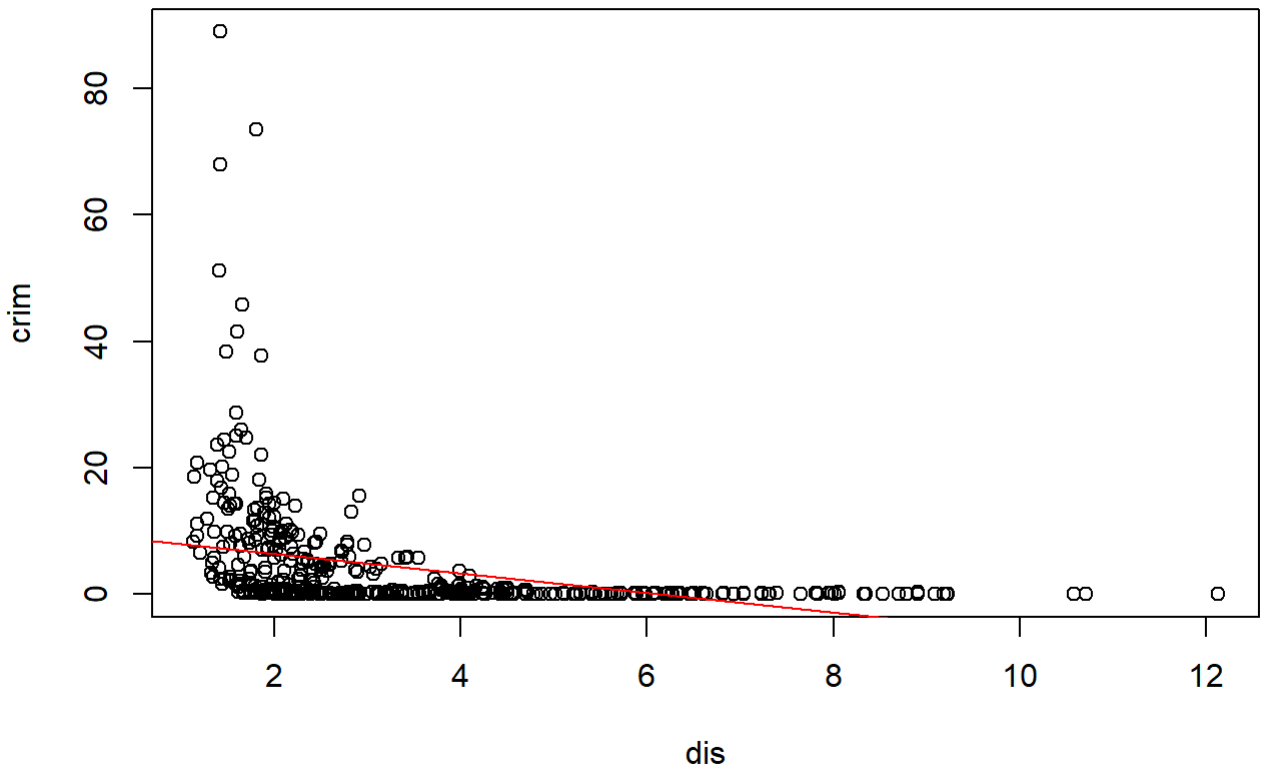

crim
0
20
40
60
80
0
5
10
15
20
25
indus

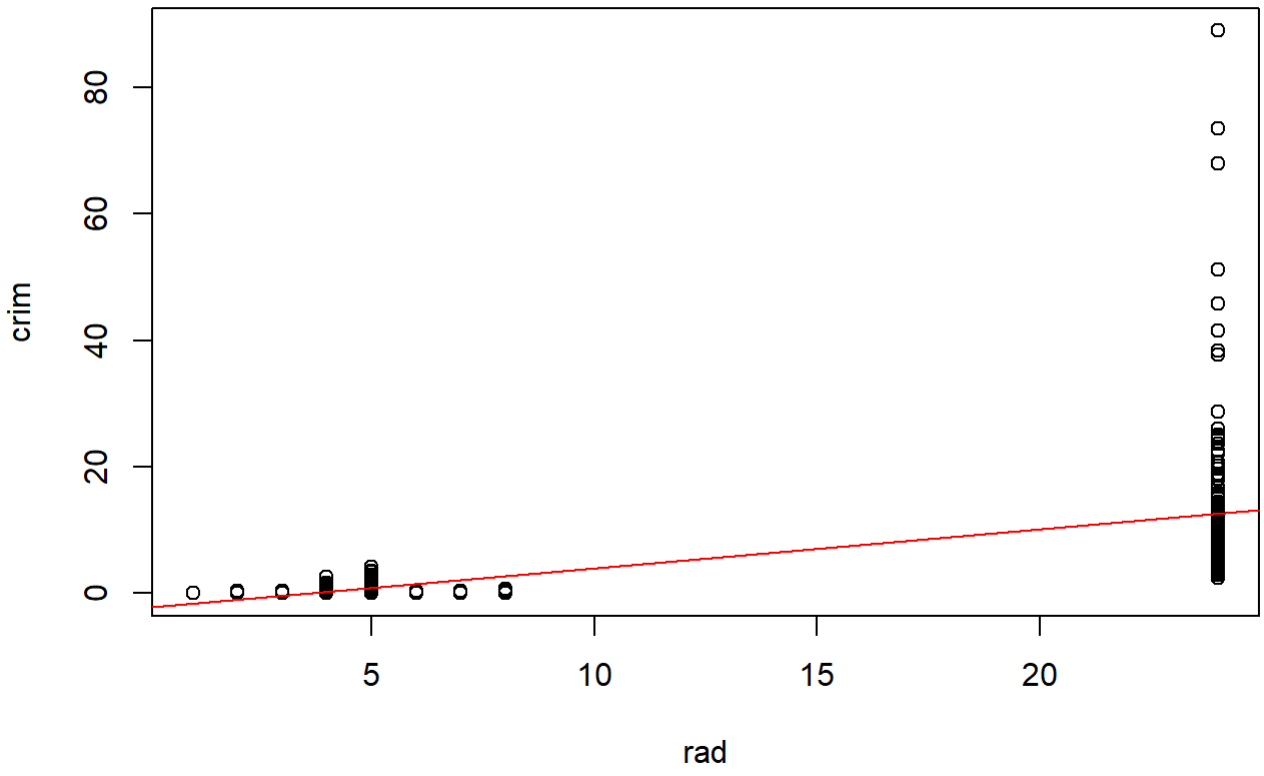

crim
0
20
40
60
80
0
1
chas

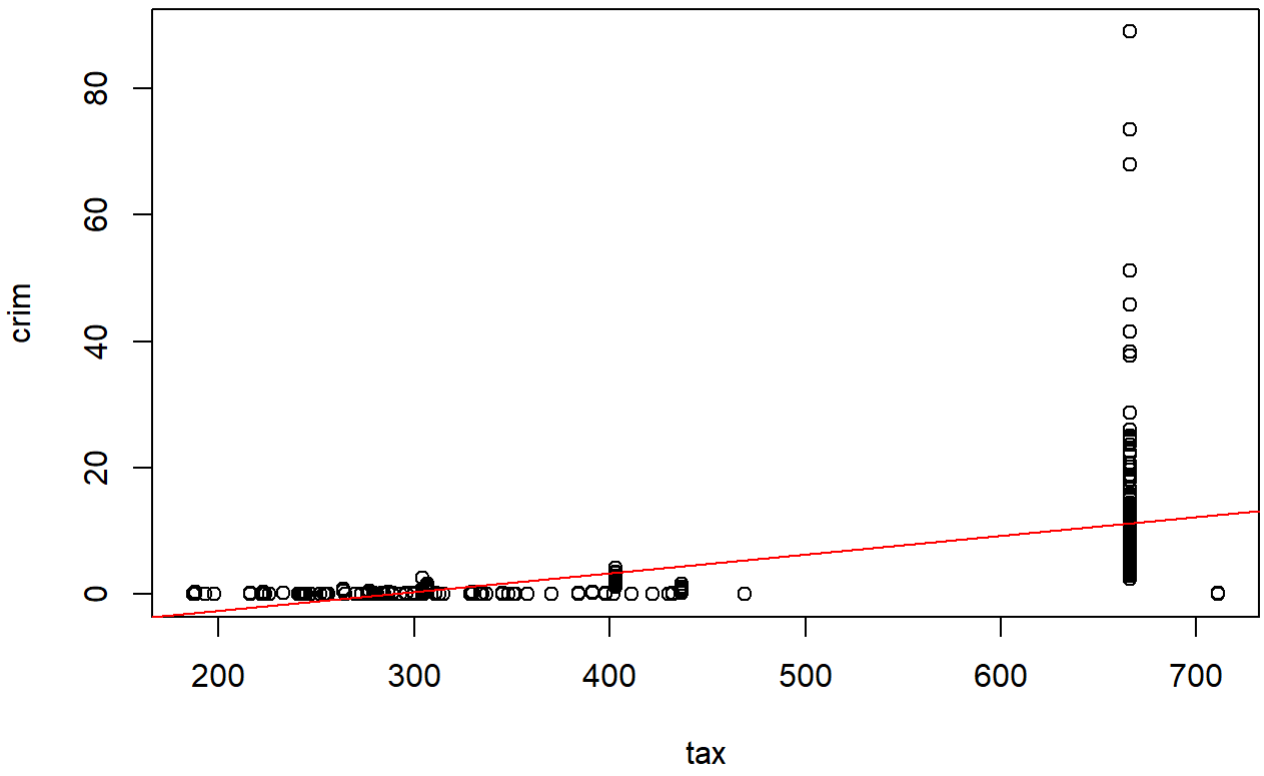

crim
0
20
40
60
80
0.4
0.5
0.6
0.7
0.8
nox

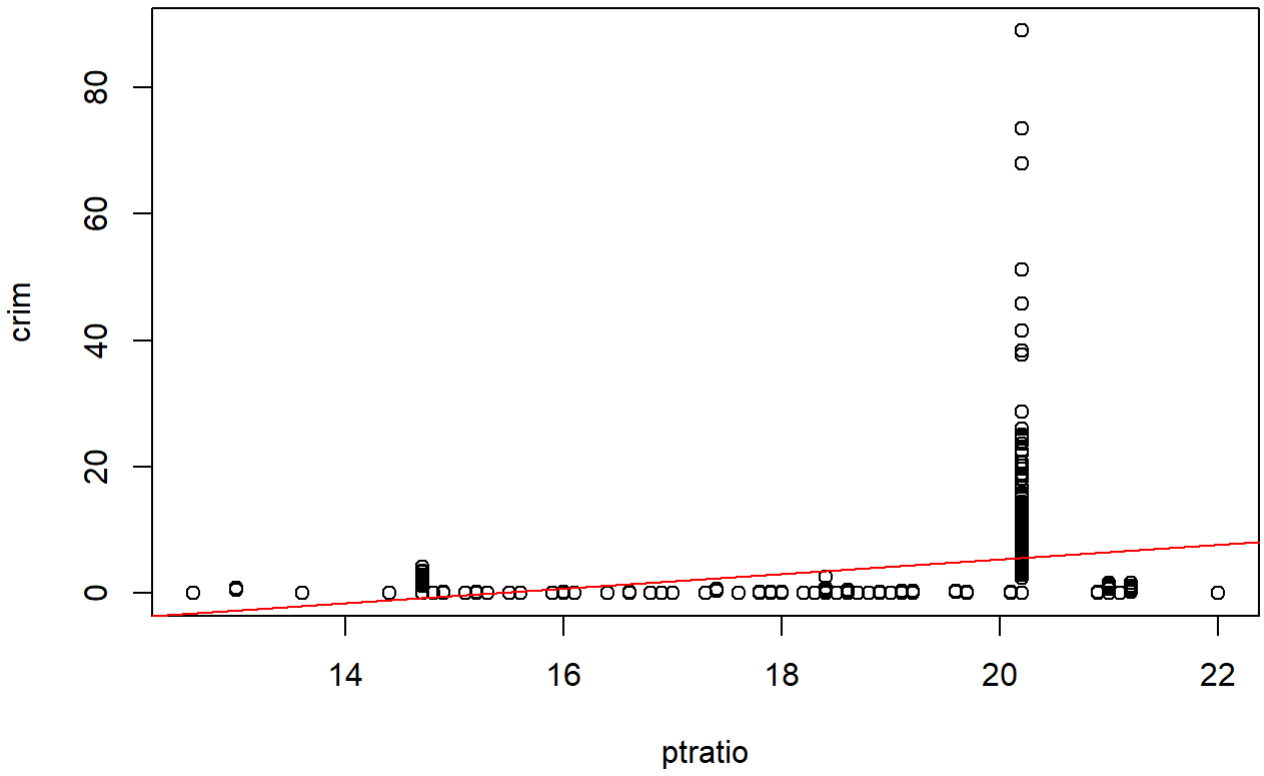

crim
0
20
40
60
80
4
5
6
7
8
rm

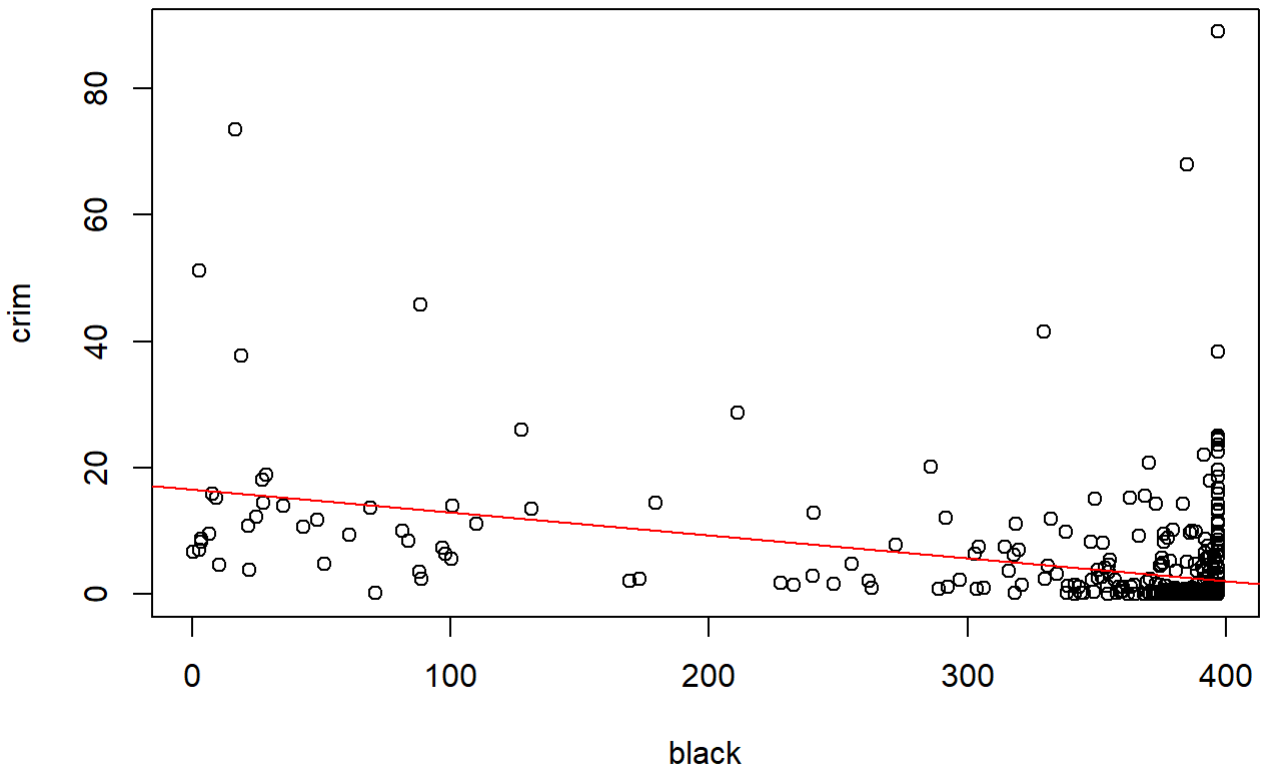

crim
0
20
40
60
80
0
20
40
60
80
100
age

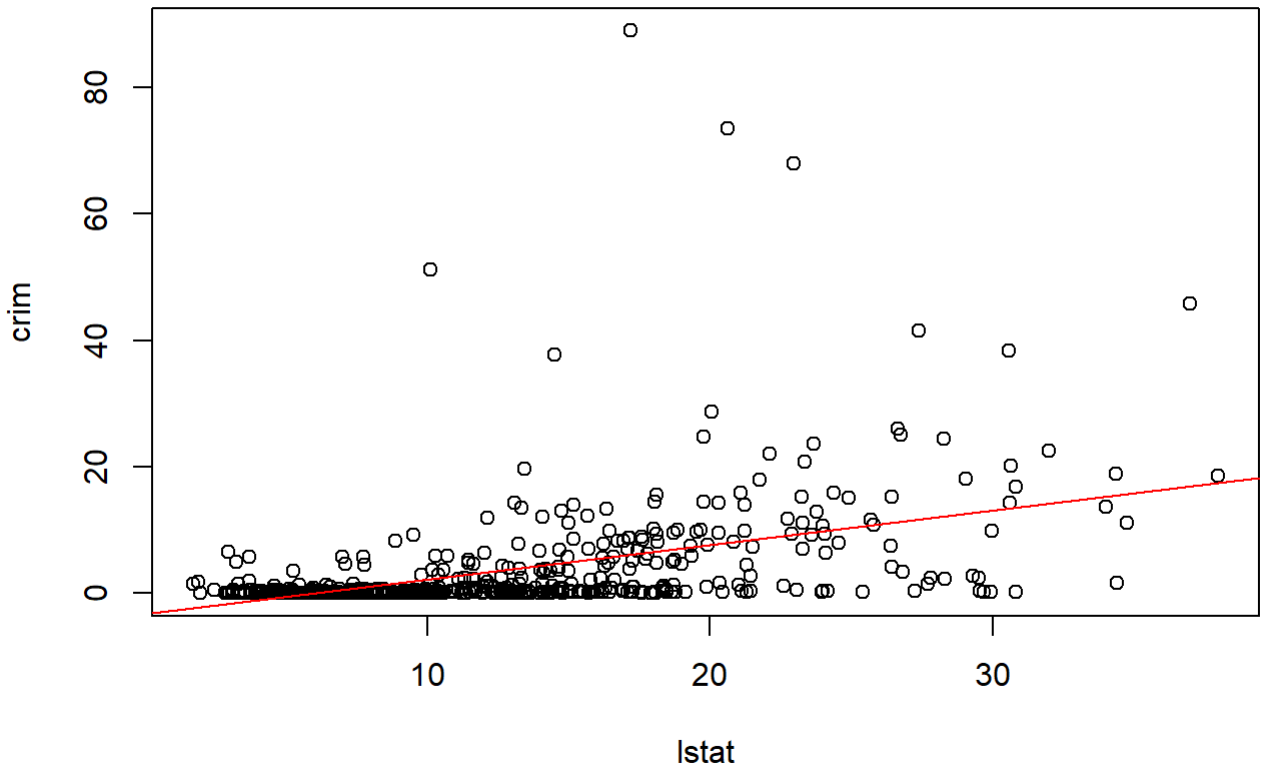

crim
0
20
40
60
80
2
4
6
8
10
12
dis

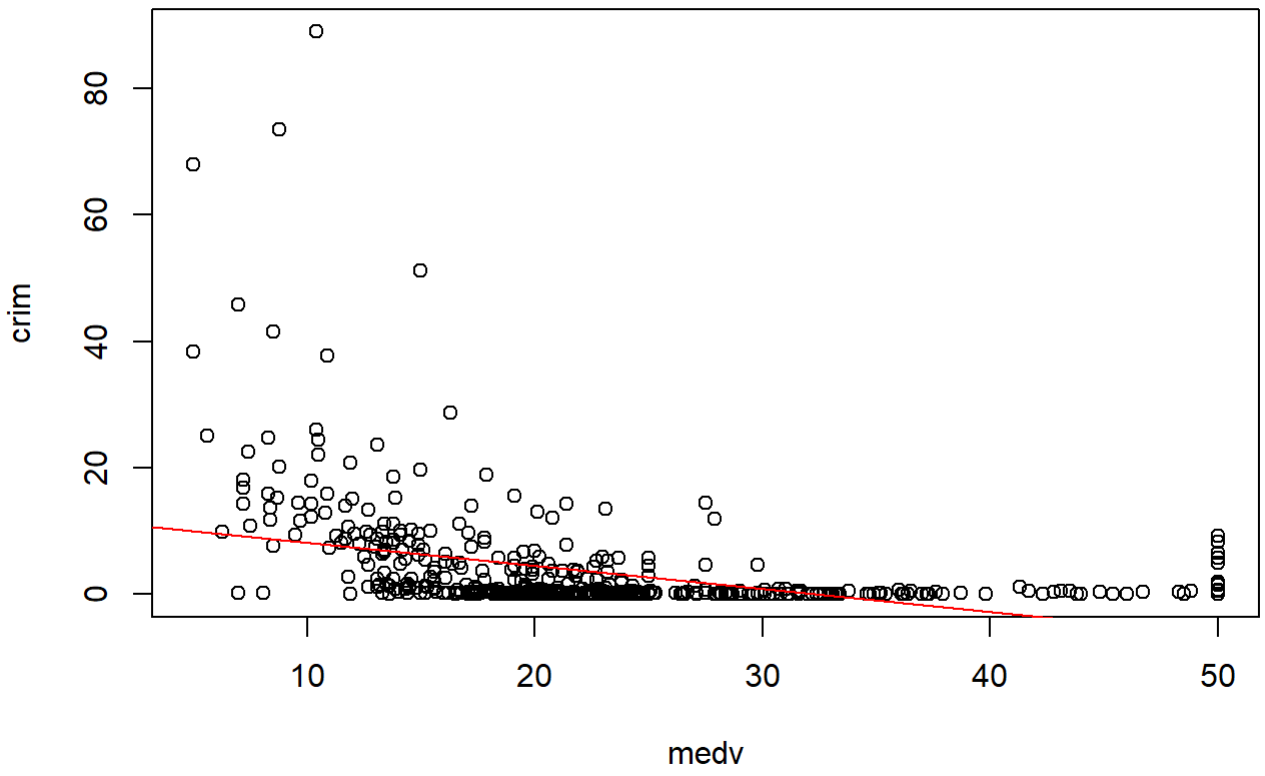

crim
0
20
40
60
80
5
10
15
20
rad


crim
0
20
40
60
80
200
300
400
500
600
700
tax


crim
0
20
40
60
80
14
16
18
20
22
ptratio


crim
0
20
40
60
80
0
100
200
300
400
black


crim
0
20
40
60
80
10
20
30
lstat




```
preds = data.frame(feature = names(Boston),
                   R2 = round(r2, 5),
                   P_value = round(p_value, 10))

preds
```

```
##    feature      R2      P_value
## 1     crim      NA           NA
## 2       zn 0.04019 0.0000055065
## 3    indus 0.16531 0.0000000000
## 4     chas 0.00312 0.2094345015
## 5      nox 0.17722 0.0000000000
## 6       rm 0.04807 0.0000006347
## 7      age 0.12442 0.0000000000
## 8      dis 0.14415 0.0000000000
## 9      rad 0.39126 0.0000000000
## 10     tax 0.33961 0.0000000000
## 11 ptratio 0.08407 0.0000000000
## 12   black 0.14827 0.0000000000
## 13   lstat 0.20759 0.0000000000
## 14    medv 0.15078 0.0000000000
```

By hypothesis testing the significance of the predictors, using an alpha of 0.05, we can see that all predictors excel "chas" have a low enough p value that we can reject the null hypothesis and conclude statistical significance between the predictor and the response

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

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon.$$

```
r2 = c()
p_value_x = c()
p_value_x2 = c()
p_value_x3 = c()

y = Boston$crim

for (i in 2:ncol(Boston)) {

  x = Boston[ ,i]
  if(is.numeric(x)){
      nl = lm(y ~ x + I(x^2) + I(x^3))

      r2[i] = summary(nl)$r.squared
      p_value_x[i] = summary(nl)$coefficients[2,4]
      p_value_x2[i] = summary(nl)$coefficients[3,4]
      p_value_x3[i] = summary(nl)$coefficients[4,4]

  }


}

preds = data.frame(feature = names(Boston),
                   R2 = round(r2, 5),
                   P_value_x = round(p_value_x, 10),
                   P_value_x2 = round(p_value_x2, 10),
                   P_value_x3 = round(p_value_x3, 10))

preds
```

```
##    feature      R2    P_value_x   P_value_x2   P_value_x3
## 1     crim      NA           NA           NA           NA
## 2       zn 0.05824 0.0026122963 0.0937504996 0.2295386205
## 3    indus 0.25966 0.0000529706 0.0000000003 0.0000000000
## 4     chas      NA           NA           NA           NA
## 5      nox 0.29698 0.0000000000 0.0000000000 0.0000000000
## 6       rm 0.06779 0.2117564139 0.3641093853 0.5085751094
## 7      age 0.17423 0.1426608270 0.0473773275 0.0066799154
## 8      dis 0.27782 0.0000000000 0.0000000000 0.0000000109
## 9      rad 0.40004 0.6234175212 0.6130098773 0.4823137740
## 10     tax 0.36888 0.1097075249 0.1374681578 0.2438506811
## 11 ptratio 0.11378 0.0030286627 0.0041195521 0.0063005136
## 12   black 0.14984 0.1385871340 0.4741750826 0.5436171817
## 13   lstat 0.21793 0.3345299858 0.0645873561 0.1298905873
## 14    medv 0.42020 0.0000000000 0.0000000000 0.0000000000
```

When we increase the flexibility of the model, we see that many predictors we thought significant are no longer useful, such as rm, rad, tax, black, and lstat. Some predictors show evidence that there is a nonlinear association with the response, like medv, nox, dis, indus, age, and ptratio. Finally, zn seems to maintain a linear relationship.