Homework1

Nam Jun Lee

09/21/2021

Q1. 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.

```
# Auto dataset into df
df <- ISLR::Auto
# linear model mpg ~ horsepower
lm.fit <- lm(mpg ~ horsepower, data = df)
summary(lm.fit)
```

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

(i) Is there a relationship between the predictor and the response?

Since the p value is less than 0.05, it means that mpg and horsepower are **statistically significant**.

(ii) How strong is the relationship between the predictor and the response?

The value of R^2 indicates that the response mpg is due to **60.59%** predictor horsepower.

(iii) Is the relationship between the predictor and the response positive or negative?

Since the coefficients of predictor horsepower is negative, the relationship is also **negative**.

(iv) What is the predicted mpg associated with a horsepower of 98? What are the associated 95 % confidence and prediction intervals?

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

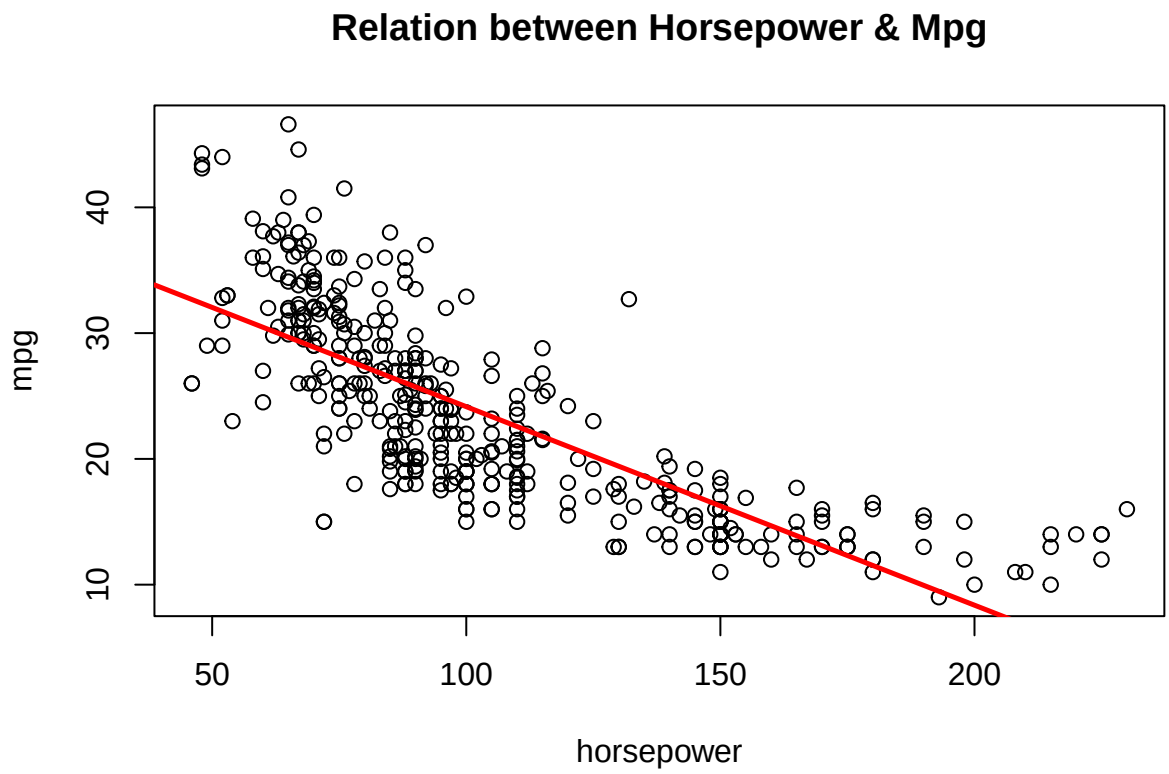
```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

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

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

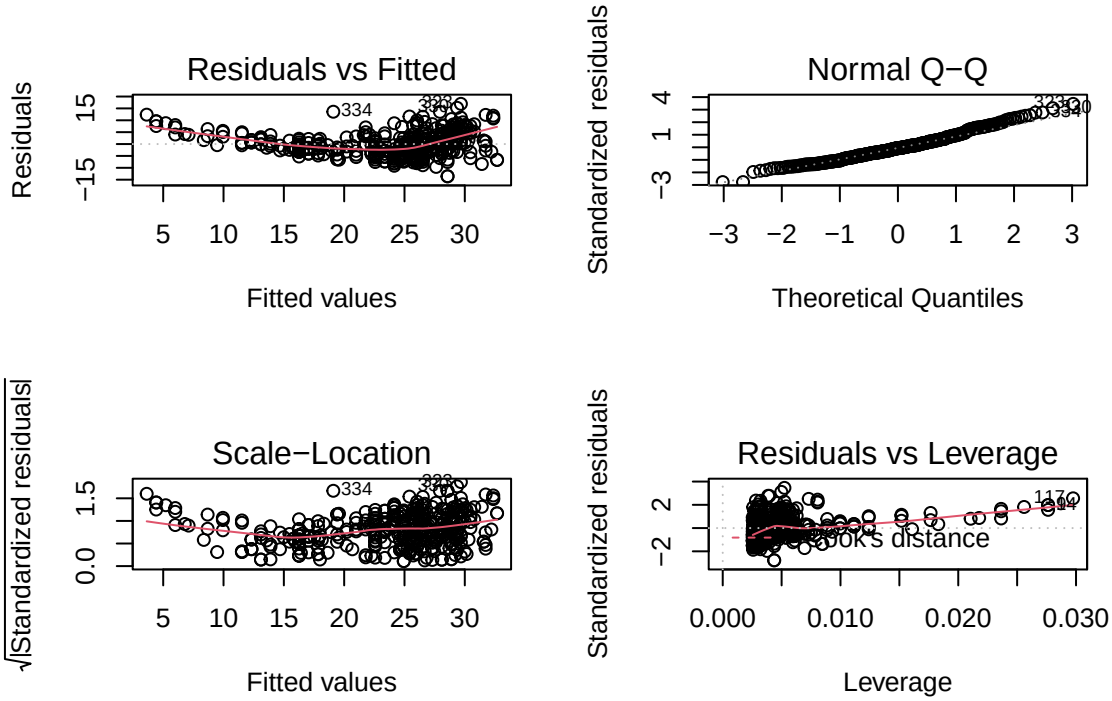
b. Plot the response and the predictor. Use the abline() function to display the least squares regression line.

```
# plot the model
plot(Auto$horsepower, Auto$mpg, main = "Relation between Horsepower & Mpg", xlab = "horsepower",
    ylab = "mpg")
abline(lm.fit, lwd = 2.5, col = "red")
```



c. Use the plot() function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.

```
# diagnostic plots of the least squares regression
par(mfrow = c(2, 2))
plot(lm.fit)
```



The Residuals versus Fitted plot does not follow a normal distribution with constant variance. The Scale-Location plot has several outliers, and the figure of Residuals versus Leverage plot indicates that there are several leverage points.

Q2. In this exercise you will create some simulated data and will fit a 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.

```
# vector X
set.seed(1)
X <- rnorm(100, mean = 0, sd = 1)
```

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

```
# vector $\epsilon$
E <- rnorm(100, mean = 0, sd = sqrt(0.25))
```

c. Using x and ϵ, generate a vector y according to the model $Y = -1 + 0.5X + \epsilon$. What is the length of the vector y? What are the values of $\beta 0$, $\beta 1$ in this linear model?

```
# vector Y
Y <- -1 + 0.5 * X + E
# length of the vector Y
length(Y)
```

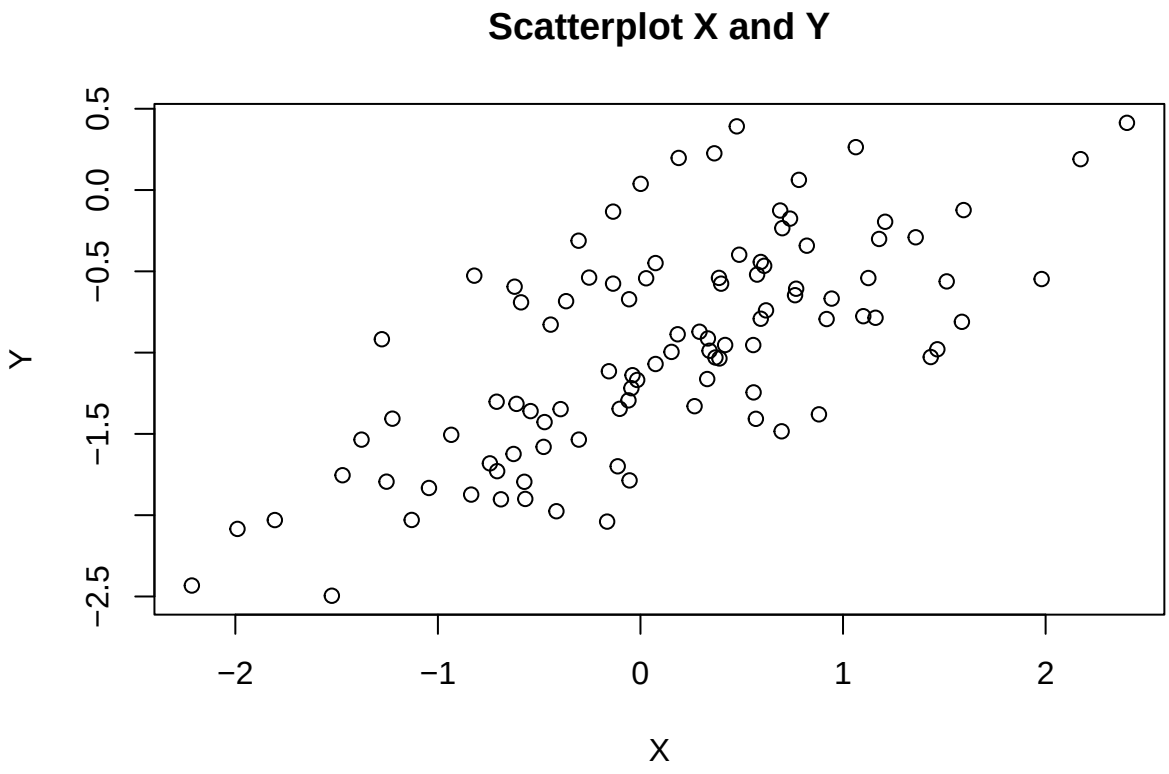
```
## [1] 100
```

Length of the vector Y is **100**. Also $\beta 0$ is **-1** and $\beta 1$ is **0.5**.

d.

(i) Create a scatterplot displaying the relationship between x and y.

```
# scatterplot model using X and Y
plot(X, Y, main = "Scatterplot X and Y")
```



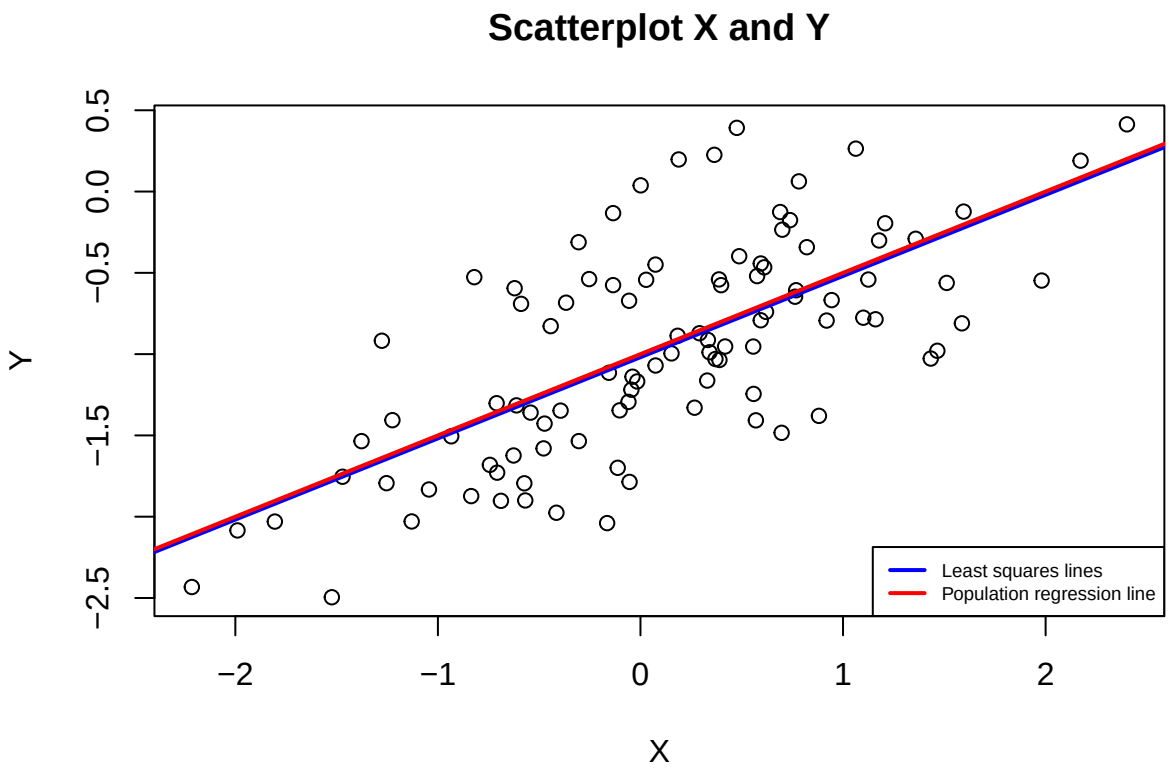
(ii) Fit a least squares linear model to predict y using x.

```
# linear model using Y and X
lm.fit1 <- lm(Y ~ X)
summary(lm.fit1)
```

```
##
## 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
```

(iii) Display the least squares line on the scatterplot. Draw the population regression line on the plot, in a different color. Use the legend() command to create an appropriate legend.

```
plot(X, Y, main = "Scatterplot X and Y")
# least squares line
abline(lm.fit1, col = "blue", lwd = 2)
# population regression line
abline(-1, 0.5, col = "red", lwd = 2)
legend("bottomright", c("Least squares lines", "Population regression line"), col = c("blue",
    "red"), lty = c(1, 1), lwd = 2, cex = 0.6)
```



e. Then fit a separate quadratic regression, i.e. $\mathbf{Y} = \beta 0 + \beta 1\mathbf{X} + \beta 2\mathbf{X2} + \epsilon$. Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the quadratic 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.

```
# quadratic regression linear model
lm.fit2 <- lm(Y ~ X + I(X^2))
summary(lm.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
```

Would we expect one to be lower than the other. Although R^2 has increased, it cannot be said that the model fit has increased because the p value of the t-statistics indicates that there is no relationship between Y and X^2.

f. Answer (e) using a test rather than RSS.

```
# test lm.fit and lm.fit2
anova(lm.fit1, lm.fit2)
```

```
## Analysis of Variance Table
##
## Model 1: Y ~ X
## Model 2: Y ~ X + I(X^2)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     98 22.709
## 2     97 22.257  1   0.45163 1.9682 0.1638
```

Model 1 represents a linear submodel including one predictor, and Model 2 corresponds to a larger quadratic model with two predictors. Model 2 is not better than Model 1, which includes only predictors. This is because p-value associated with the F statistic is greater than 0.05.

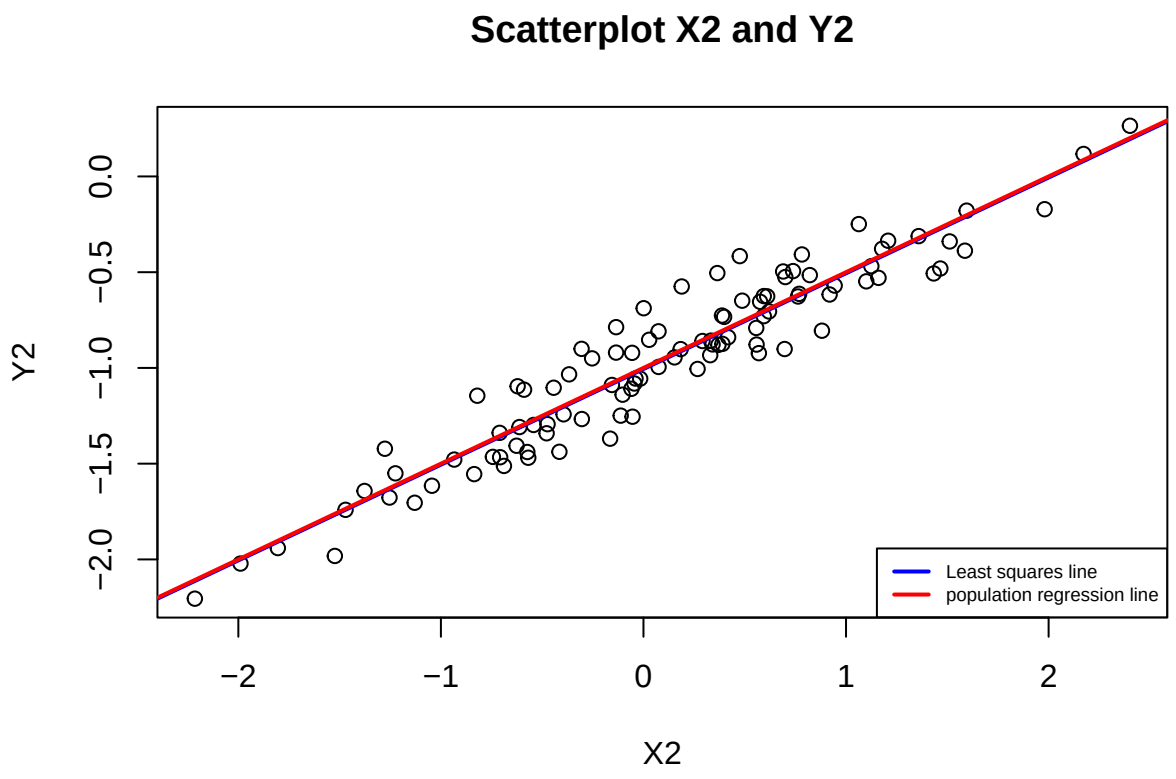
g. Repeat (a)-(f) after modifying the data generation process in such a way that there is less noise in the data. You can do this by decreasing the variance of the normal distribution used to generate the error term in (b). Describe your results.

```
# Repeat (a) - (d)
set.seed(1)
X2 <- rnorm(100, mean = 0, sd = 1)
# decreasing the variance of the normal distribution 0.15
E2 <- rnorm(100, mean = 0, sd = 0.15)
Y2 <- -1 + 0.5 * X2 + E2
lm.fit3 <- lm(Y2 ~ X2)
summary(lm.fit3)
```

```
##
## Call:
## lm(formula = Y2 ~ X2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.28153 -0.09206 -0.02092  0.08091  0.35193
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00565    0.01455  -69.13   <2e-16 ***
## X2           0.49984    0.01616   30.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1444 on 98 degrees of freedom
## Multiple R-squared:  0.9071, Adjusted R-squared:  0.9061
## F-statistic: 956.8 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
plot(X2, Y2, main = "Scatterplot X2 and Y2")
abline(lm.fit3, col = "blue", lwd = 2)
abline(-1, 0.5, col = "red", lwd = 2)
legend("bottomright", legend = c("Least squares line", "population regression line"),
    cex = 0.6, border = "white", col = c("blue", "red"), lty = c(1, 1), lwd = 2)
```



The standard deviation of the error was change to 0.15. It is a little closer to the least squares model. Also, the RSE value is significantly reduced.

```r
# Repeat (e) - (f)
lm.fit4 <- lm(Y2 ~ X2 + I(X2^2))
summary(lm.fit4)
```

```
##
## Call:
## lm(formula = Y2 ~ X2 + I(X2^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29475 -0.09381 -0.01932  0.08704  0.34050
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.99149    0.01765 -56.181   <2e-16 ***
## X2           0.50257    0.01620  31.028   <2e-16 ***
## I(X2^2)     -0.01784    0.01271  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1437 on 97 degrees of freedom
## Multiple R-squared:  0.9089, Adjusted R-squared:  0.9071
## F-statistic: 484.1 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
anova(lm.fit3, lm.fit4)
```

```
## Analysis of Variance Table
##
## Model 1: Y2 ~ X2
## Model 2: Y2 ~ X2 + I(X2^2)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     98 2.0438
## 2     97 2.0032  1  0.040646 1.9682 0.1638
```

Although R^2 has increased, it cannot be said that the model fit has increased because the p value of the t-statistics indicates that there is no relationship between Y and X^2. Model 2 is not better than Model 1, which includes only predictors, which includes only predictors. This is because p-value associated with the F statistic is greater than 0.05.

Q3. This problem involves the Boston data set, which we saw in class. 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.

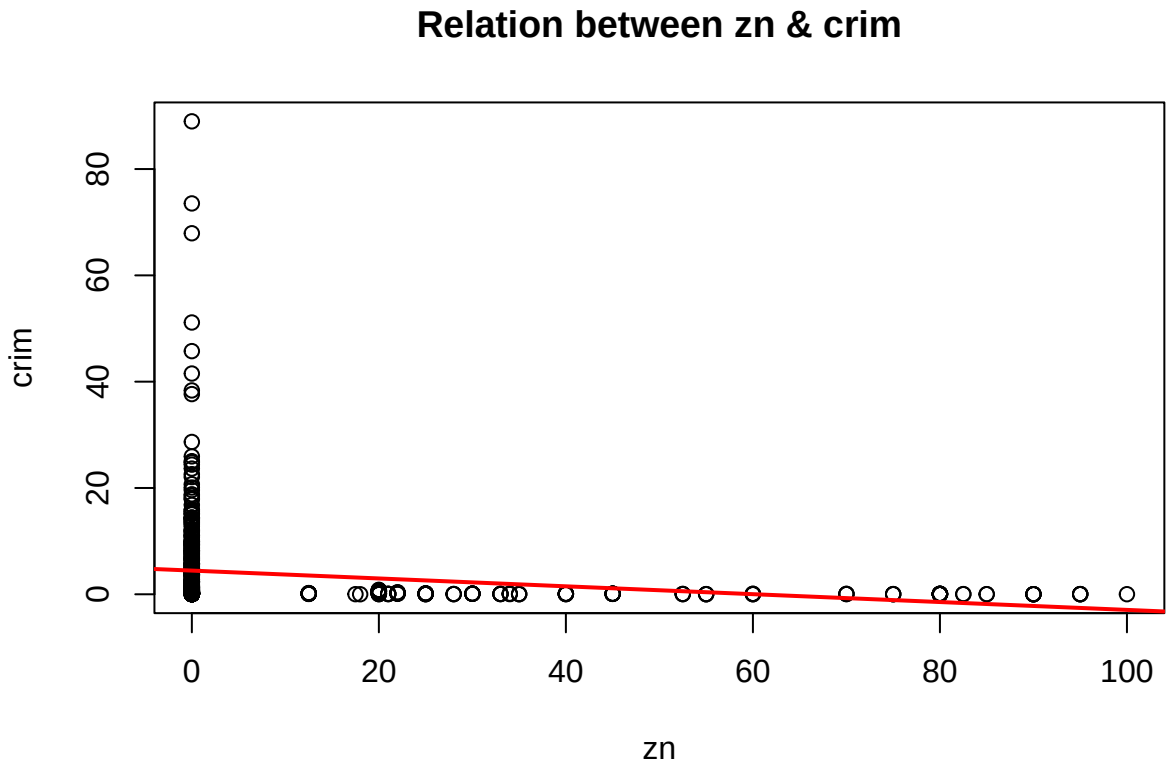
```
# Boston dataset into boston
boston <- MASS::Boston
# columns in boston
names(boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

```
# model crim~zn
fit.zn <- lm(crim ~ zn, data = boston)
summary(fit.zn)
```

```
##
## Call:
## lm(formula = crim ~ zn, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

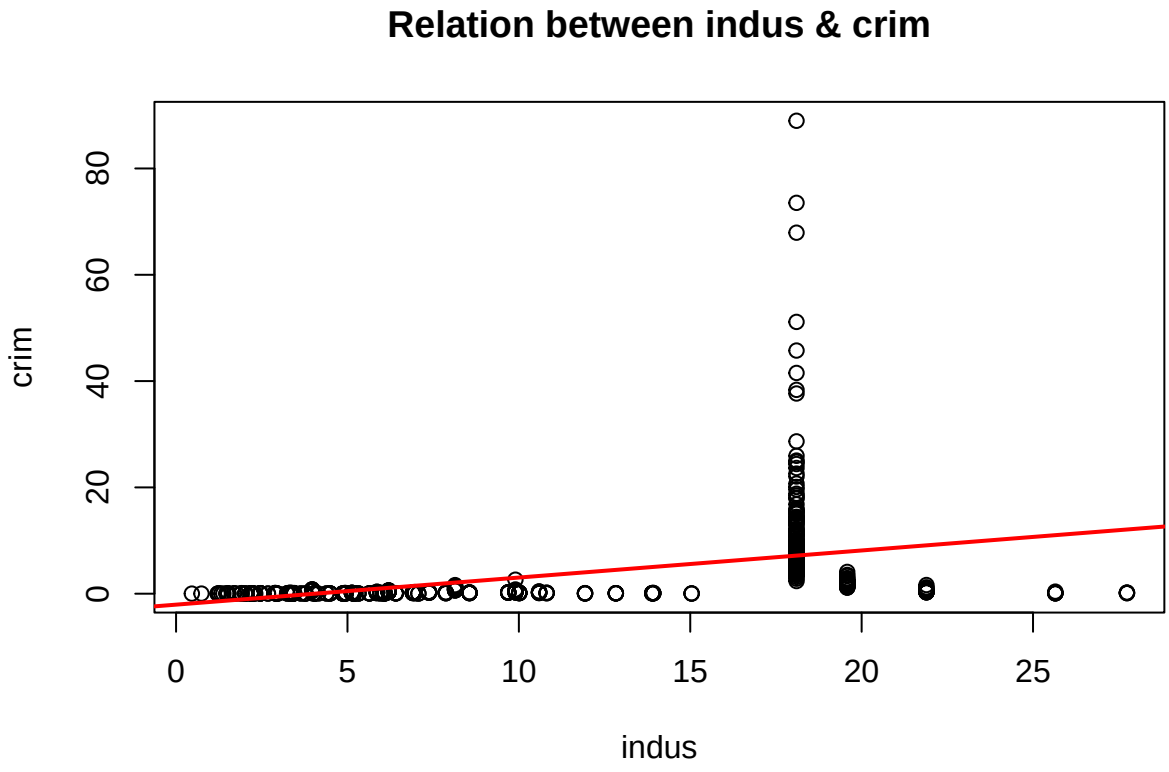
```
plot(boston$zn, boston$crim, main = "Relation between zn & crim", xlab = "zn", ylab = "crim")
abline(fit.zn, col = "red", lwd = 2)
```




```
# model crim~indus
fit.indus <- lm(crim ~ indus, data = boston)
summary(fit.indus)
```

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

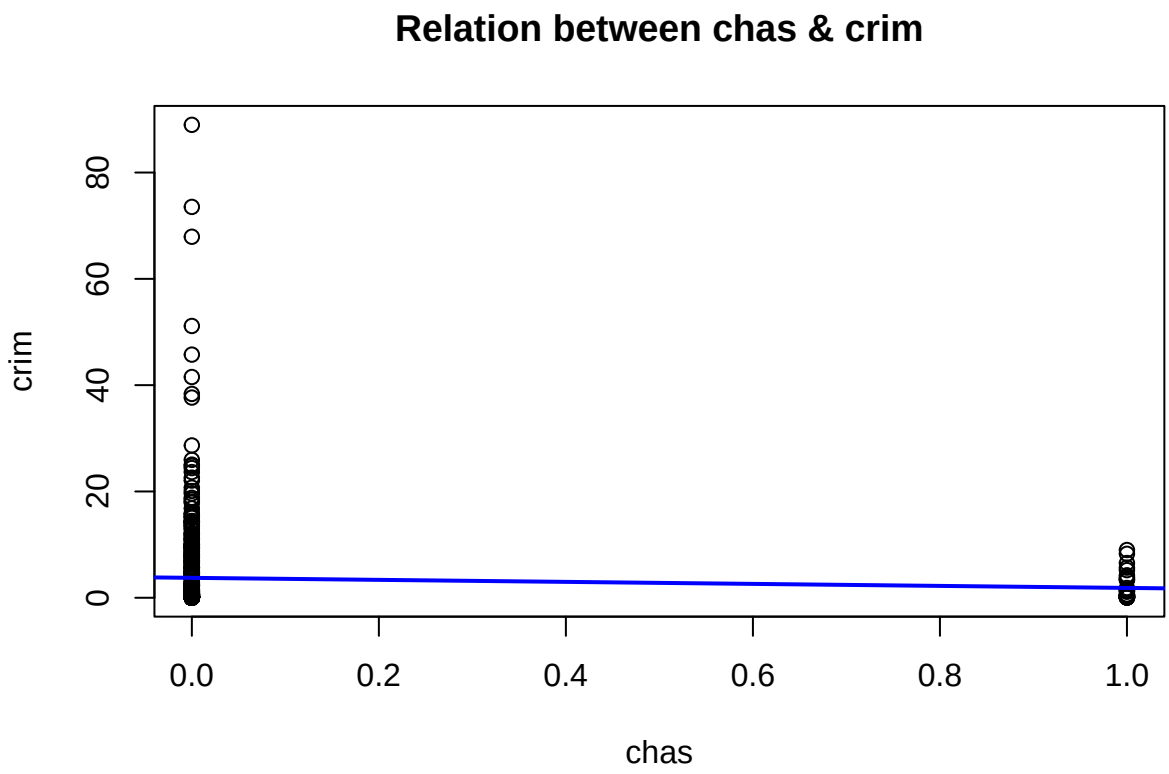
```
plot(boston$indus, boston$crim, main = "Relation between indus & crim", xlab = "indus",
    ylab = "crim")
abline(fit.indus, col = "red", lwd = 2)
```




```
# model crim~chas
fit.chas <- lm(crim ~ chas, data = boston)
summary(fit.chas)
```

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

```
plot(boston$chas, boston$crim, main = "Relation between chas & crim", xlab = "chas",
    ylab = "crim")
abline(fit.chas, col = "blue", lwd = 2)
```

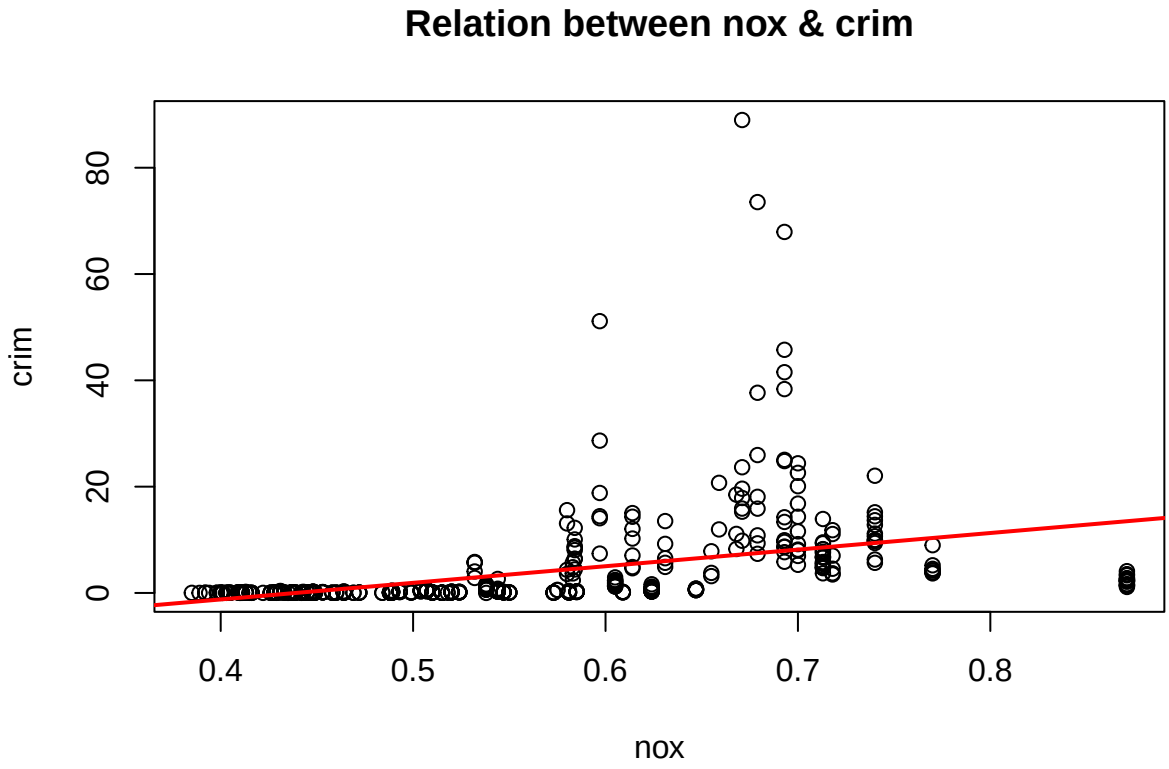



```
# model crim~nox
fit.nox <- lm(crim ~ nox, data = boston)
summary(fit.nox)
```

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

```
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
plot(boston$nox, boston$crim, main = "Relation between nox & crim", xlab = "nox",
    ylab = "crim")
abline(fit.nox, col = "red", lwd = 2)
```

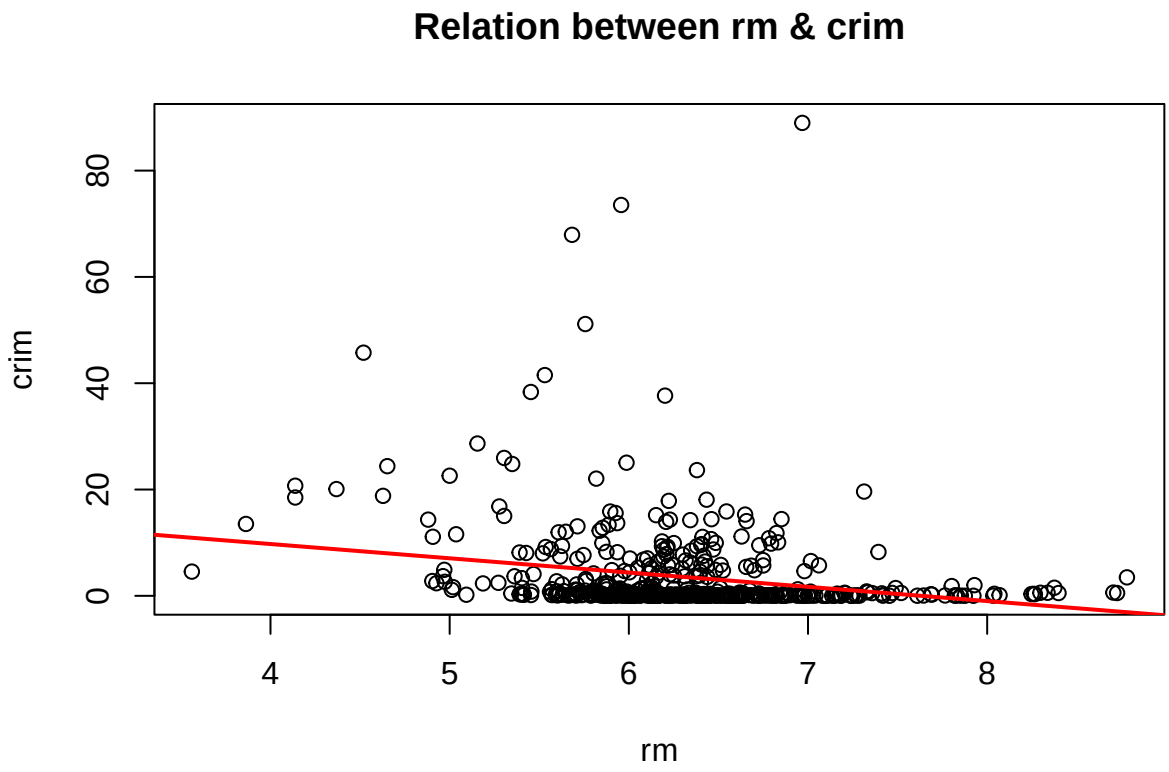



```
# model crim~rm
fit.rm <- lm(crim ~ rm, data = boston)
summary(fit.rm)
```

```
##
## Call:
## lm(formula = crim ~ rm, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

```
plot(boston$rm, boston$crim, main = "Relation between rm & crim", xlab = "rm", ylab = "crim")
abline(fit.rm, col = "red", lwd = 2)
```

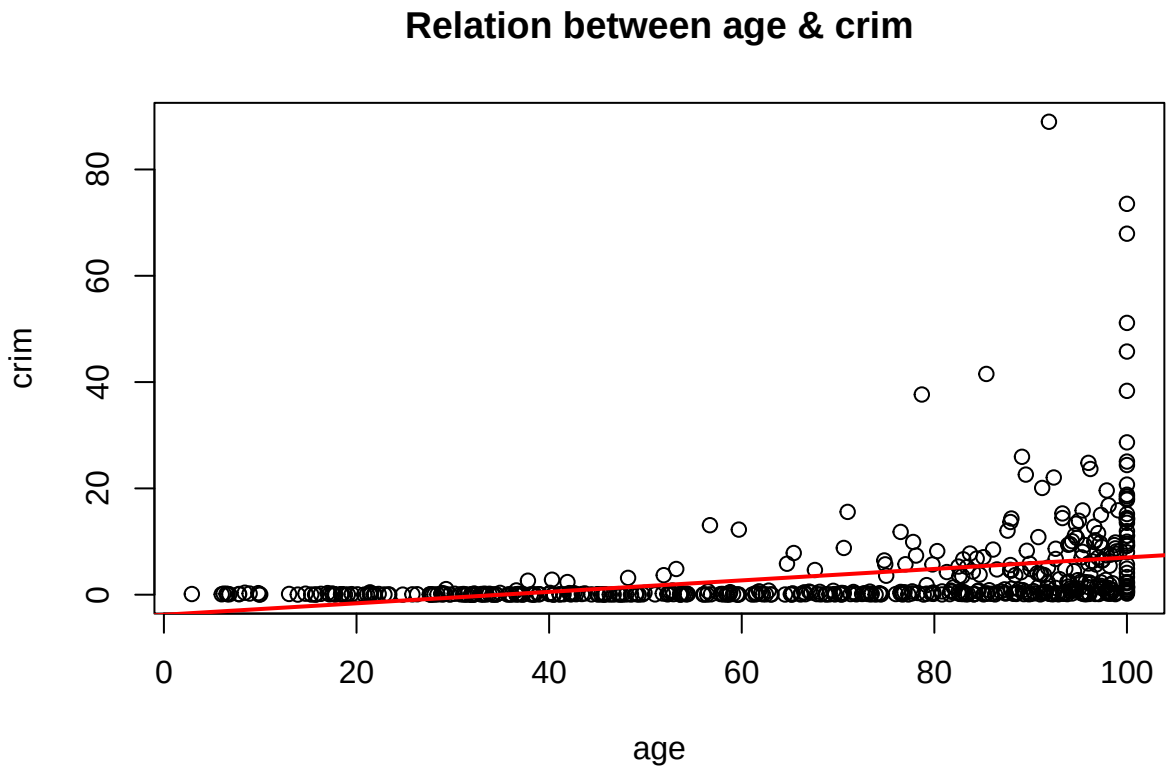



```
# model crim~age
fit.age <- lm(crim ~ age, data = boston)
summary(fit.age)
```

```
##
## Call:
## lm(formula = crim ~ age, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

```
plot(boston$age, boston$crim, main = "Relation between age & crim", xlab = "age",
    ylab = "crim")
abline(fit.age, col = "red", lwd = 2)
```

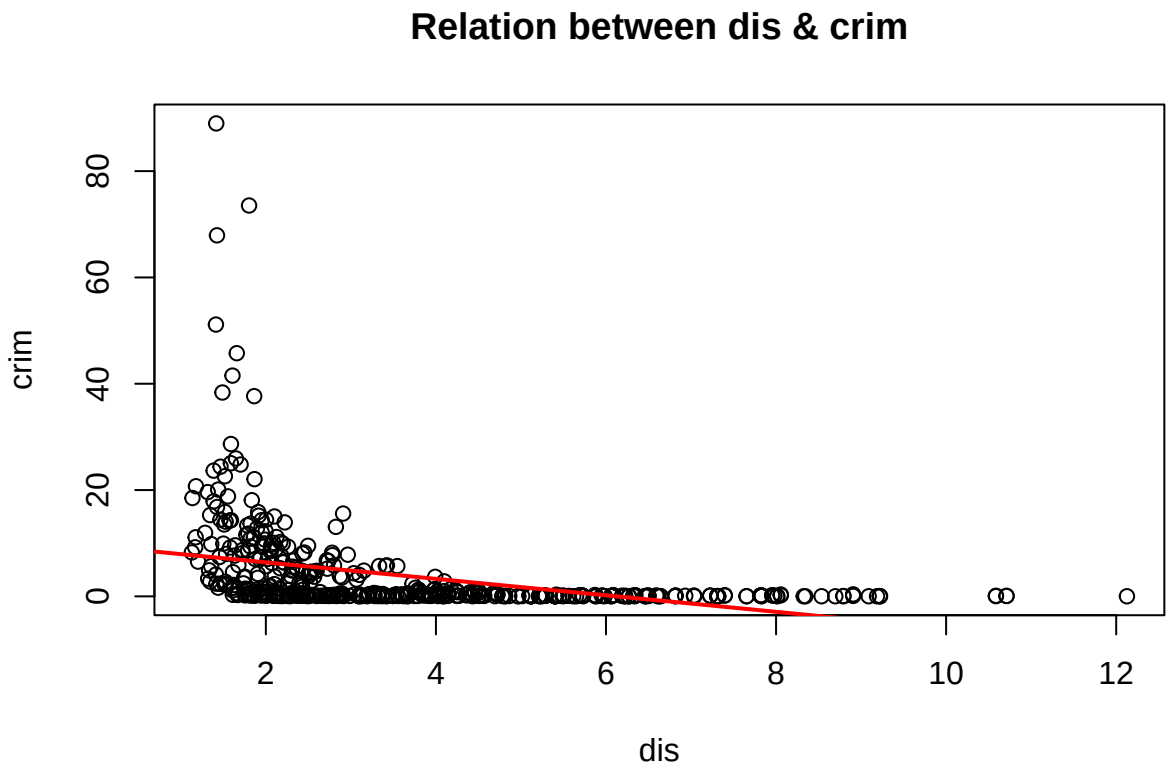



```
# model crim~dis
fit.dis <- lm(crim ~ dis, data = boston)
summary(fit.dis)
```

```
##
## Call:
## lm(formula = crim ~ dis, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
plot(boston$dis, boston$crim, main = "Relation between dis & crim", xlab = "dis",
    ylab = "crim")
abline(fit.dis, col = "red", lwd = 2)
```

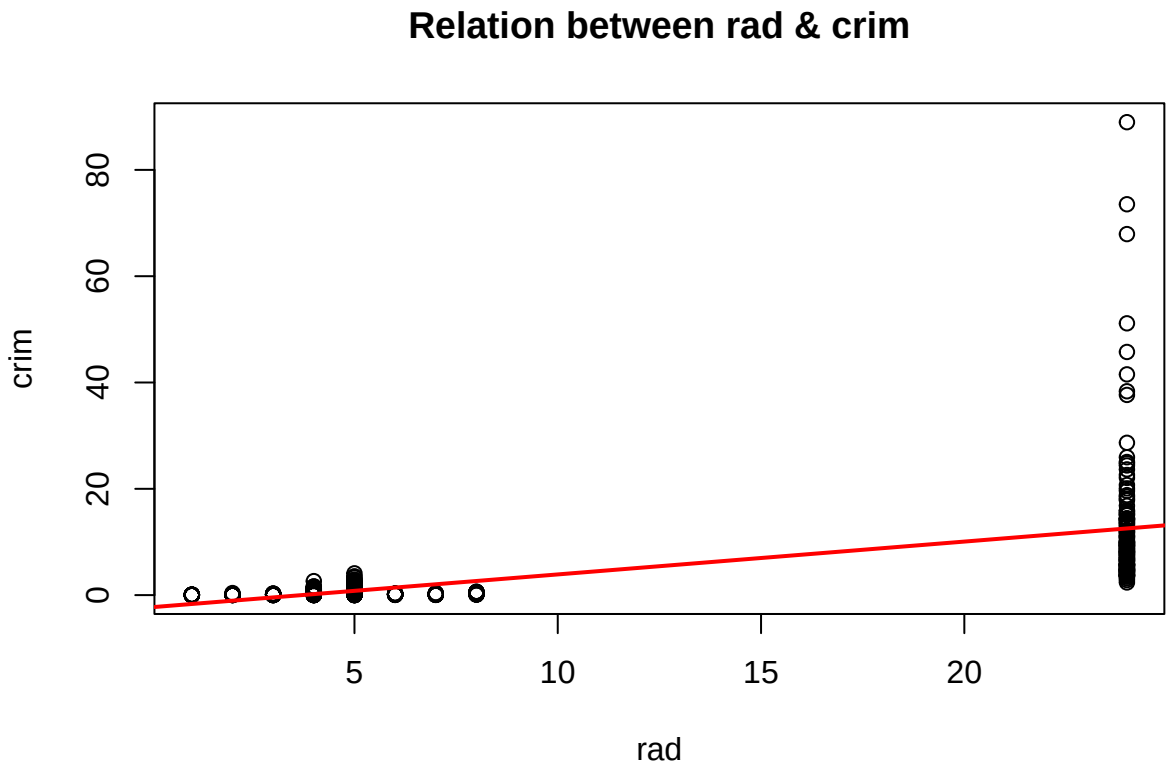



```
# model crim~rad
fit.rad <- lm(crim ~ rad, data = boston)
summary(fit.rad)
```

```
##
## Call:
## lm(formula = crim ~ rad, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
plot(boston$rad, boston$crim, main = "Relation between rad & crim", xlab = "rad",
    ylab = "crim")
abline(fit.rad, col = "red", lwd = 2)
```

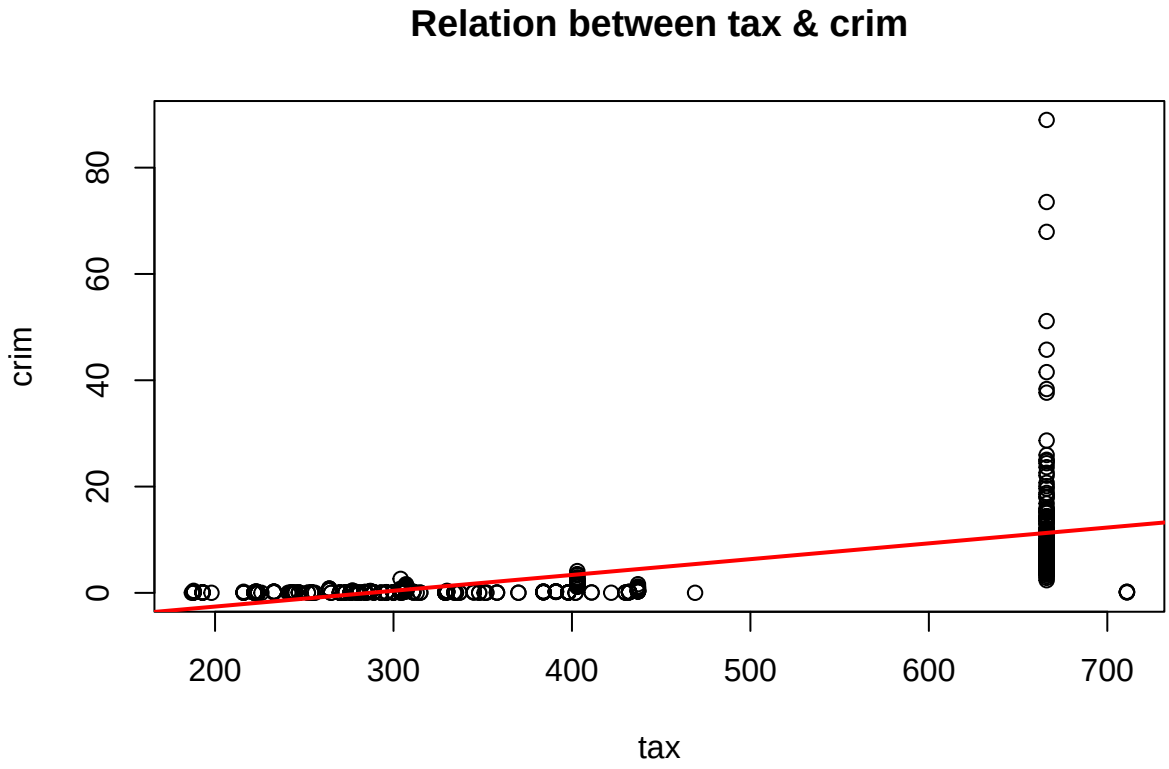



```r
# model crim~tax
fit.tax <- lm(crim ~ tax, data = boston)
summary(fit.tax)
```

```
##
## Call:
## lm(formula = crim ~ tax, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
plot(boston$tax, boston$crim, main = "Relation between tax & crim", xlab = "tax",
    ylab = "crim")
abline(fit.tax, col = "red", lwd = 2)
```

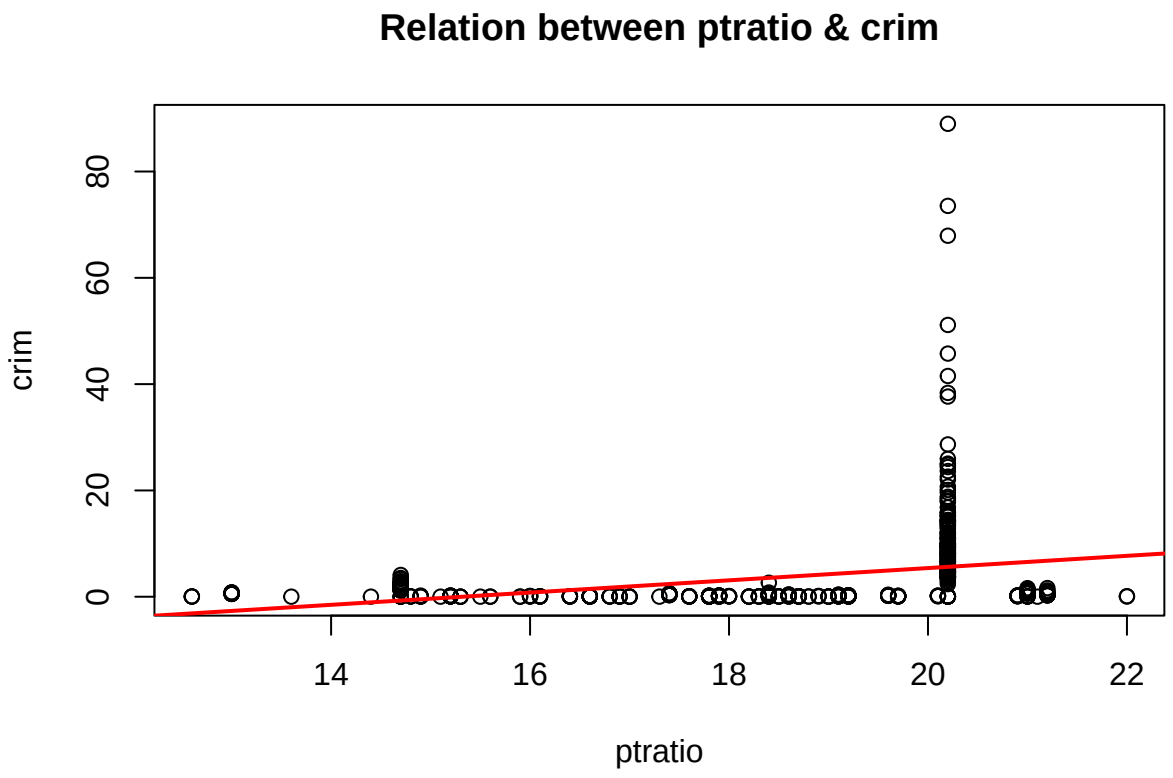



```
# model crim~ptratio
fit.ptratio <- lm(crim ~ ptratio, data = boston)
summary(fit.ptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

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

```
plot(boston$ptratio, boston$crim, main = "Relation between ptratio & crim", xlab = "ptratio",
    ylab = "crim")
abline(fit.ptratio, col = "red", lwd = 2)
```

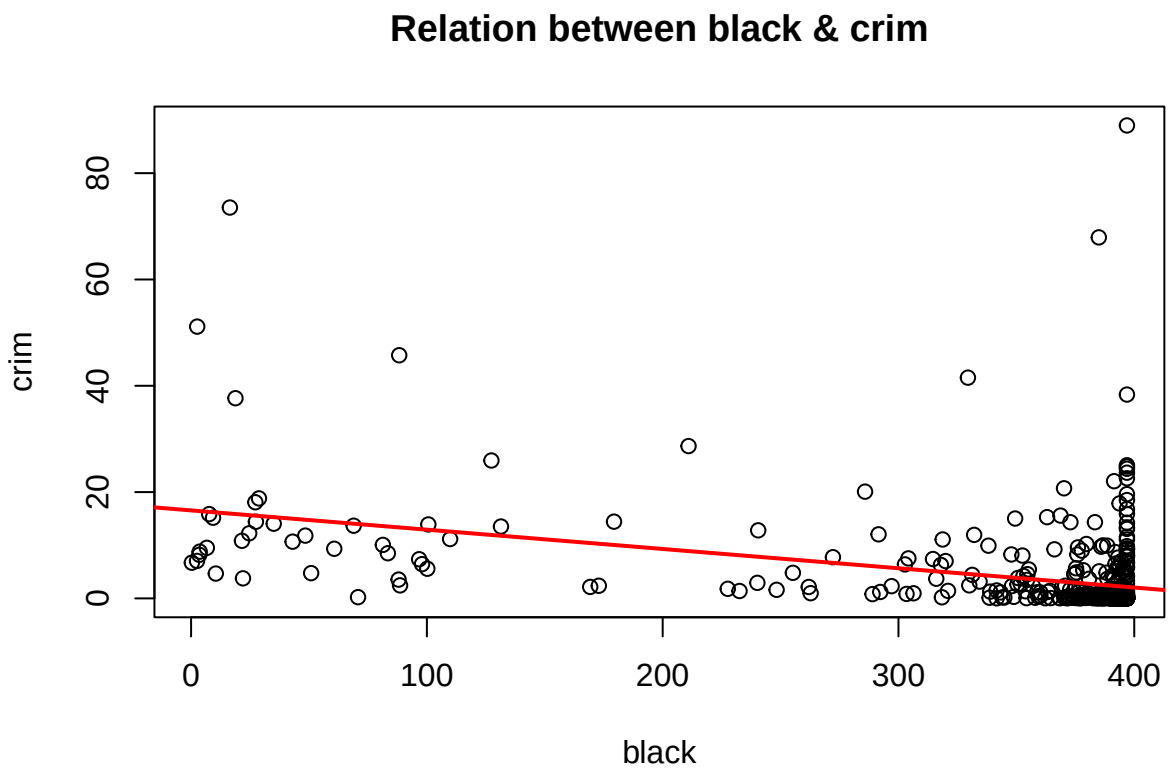



```
# model crim~black
fit.black <- lm(crim ~ black, data = boston)
summary(fit.black)
```

```
##
## Call:
## lm(formula = crim ~ black, data = boston)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   0.003873  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
plot(boston$black, boston$crim, main = "Relation between black & crim", xlab = "black",
    ylab = "crim")
abline(fit.black, col = "red", lwd = 2)
```

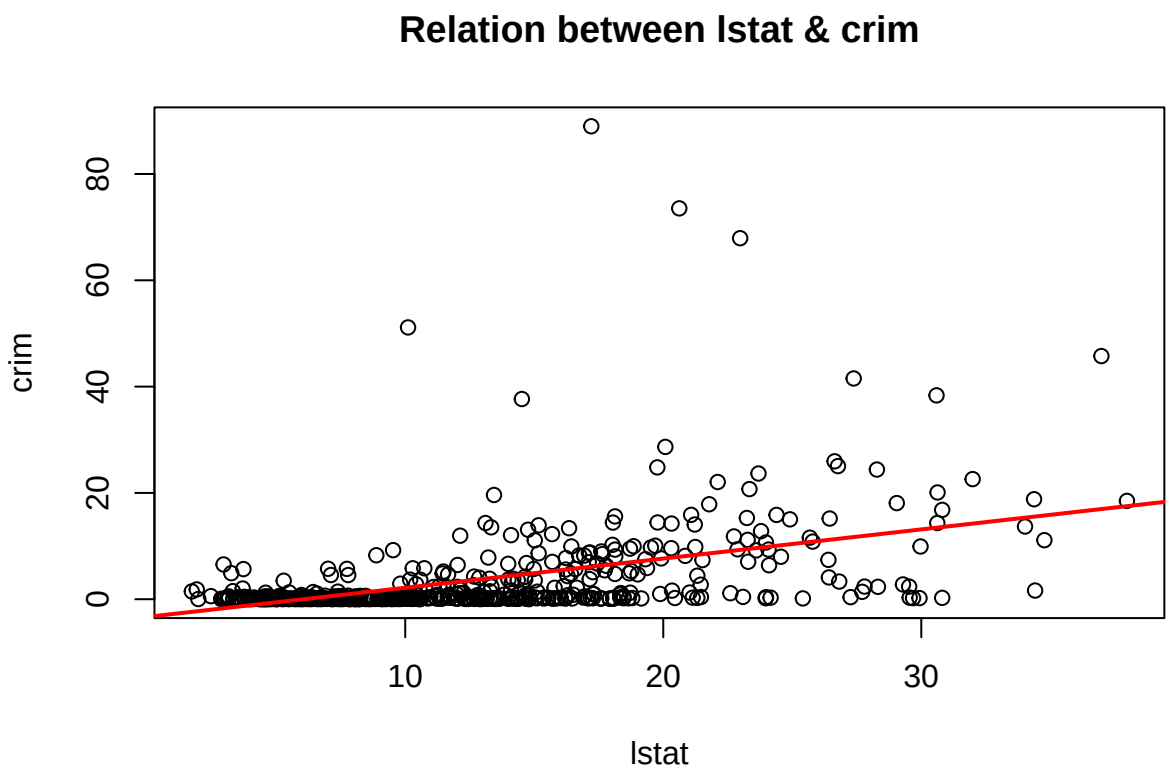



```
# model crim~lstat
fit.lstat <- lm(crim ~ lstat, data = boston)
summary(fit.lstat)
```

```
##
## Call:
```

```
## lm(formula = crim ~ lstat, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
plot(boston$lstat, boston$crim, main = "Relation between lstat & crim", xlab = "lstat",
    ylab = "crim")
abline(fit.lstat, col = "red", lwd = 2)
```

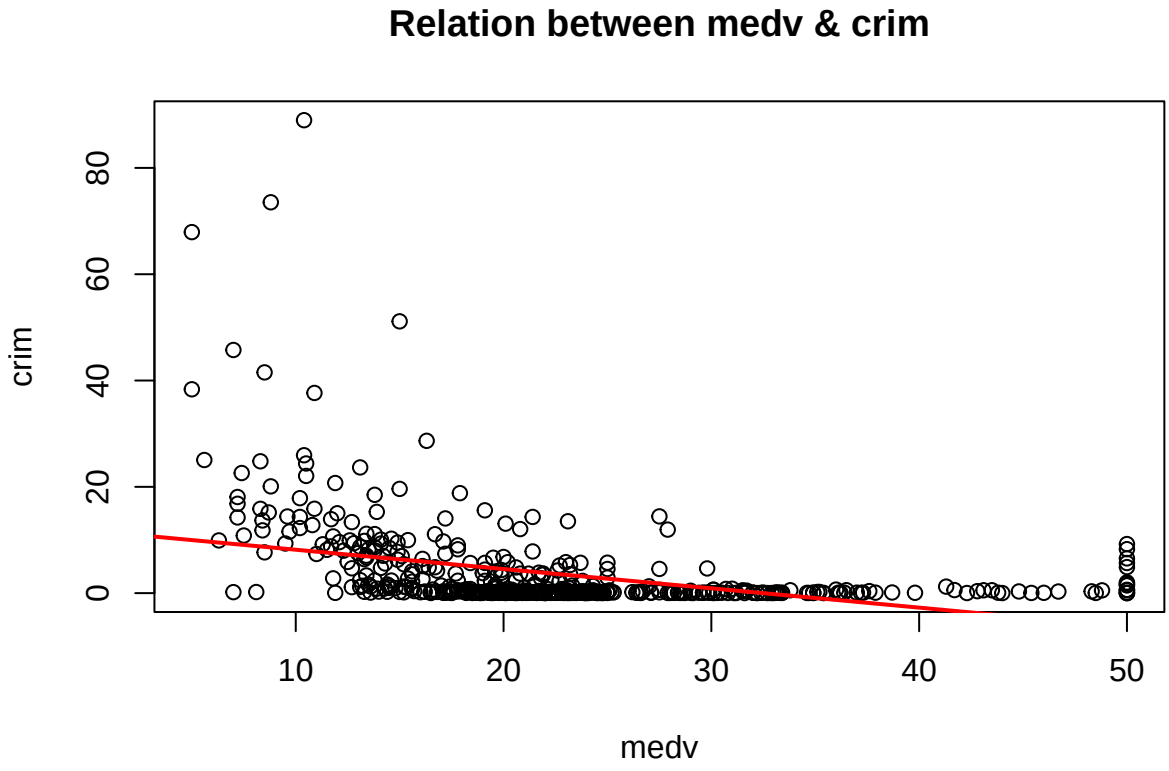



```
# model crim~medv
fit.medv <- lm(crim ~ medv, data = boston)
summary(fit.medv)
```

```
##
```

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

```
plot(boston$medv, boston$crim, main = "Relation between medv & crim", xlab = "medv",
    ylab = "crim")
abline(fit.medv, col = "red", lwd = 2)
```



Except for the **chas** predictor, each predictor is statistically significant to the response variable.

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 : βj = 0?

```
# multiple regression model all predictors
fit.all <- lm(crim ~ ., data = boston)
summary(fit.all)
```

```
##
## Call:
## lm(formula = crim ~ ., data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

It may reject the null hypothesis for **zn**, **dis**, **rad**, **black**, **medv**. zn and black at the 0.05 level, medv at the 0.01 level, dis and rad at the 0.001 level.

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.

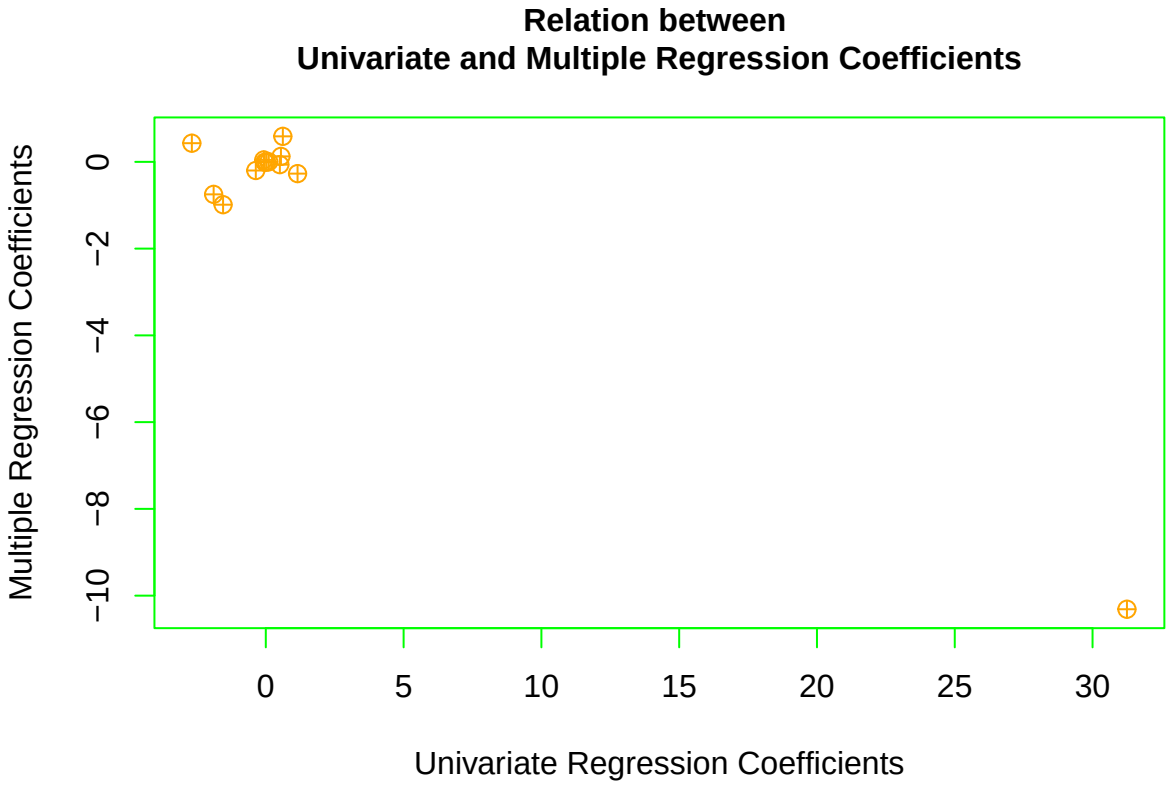
```r
# univariate regression coefficients
univariate <- vector("numeric", 0)
# except (Intercept)
univariate <- c(fit.zn$coefficients[2], fit.indus$coefficients[2], fit.chas$coefficients[2],
    fit.nox$coefficients[2], fit.rm$coefficients[2], fit.age$coefficients[2], fit.dis$coefficients[2],
    fit.rad$coefficients[2], fit.tax$coefficients[2], fit.ptratio$coefficients[2],
    fit.black$coefficients[2], fit.lstat$coefficients[2], fit.medv$coefficients[2])
univariate
```

```
##          zn       indus        chas         nox          rm         age
## -0.07393498  0.50977633 -1.89277655 31.24853120 -2.68405122  0.10778623
##         dis         rad         tax     ptratio       black       lstat
## -1.55090168  0.61791093  0.02974225  1.15198279 -0.03627964  0.54880478
##        medv
## -0.36315992
```

```r
# multiple regression cofficients
multiple <- vector("numeric", 0)
# except (Intercept)
multiple <- c(fit.all$coefficients[-1])
multiple
```

```
##            zn         indus          chas           nox            rm
##   0.044855215  -0.063854824  -0.749133611 -10.313534912   0.430130506
##           age           dis           rad           tax       ptratio
##   0.001451643  -0.987175726   0.588208591  -0.003780016  -0.271080558
##         black         lstat          medv
##  -0.007537505   0.126211376  -0.198886821
```

```r
# univariate regression model is shown on the x-axis, multiple regression model
# is shown on the y-axis
plot(univariate, multiple, main = "Relation between \nUnivariate and Multiple Regression Coefficients",
    xlab = "Univariate Regression Coefficients", ylab = "Multiple Regression Coefficients",
    col = "orange", pch = 10, cex = 1.2, fg = "green", cex.main = 1)
```



The **nox coefficient** is below -10 in the univariate regression model and above 30 in the multiple regression model, which is very far from other predictors.

There is a difference between a univariate regression coefficient and a multiple regression coefficient. This difference shows that the slope term of a univariate regression ignores other predictors and shows the average effect of increasing predictors, and the slope term of multiple regression holds other predictors and shows the average of increasing predictors.