Machine Learning - Lab 5 - Solution

Linear regression

Souhaib BEN TAIEB

9 March 2020

Simple Linear Regression

Exercise 1

We observe a dataset $\mathcal{D} = \{(y_i, x_i)\}_{i=1}^n$ where $y_i, x_i \in \mathbb{R}$. We consider the following optimization problem:

$$\underset{\beta_0, \beta_1 \in \mathbb{R}}{\text{Minimize}} \ \text{RSS}(\beta_0, \beta_1),$$

where

$$\text{RSS}(\beta_0, \beta_1) = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2.$$

The solution to the previous optimization problem is given by

$$\begin{aligned}
\hat{\beta}_1 &= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \\
\hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x}.
\end{aligned}$$

where $\bar{y} = \frac{1}{n} \sum_i y_i$ and $\bar{x} = \frac{1}{n} \sum_i x_i$.

We ask you to prove that $(\hat{\beta}_0, \hat{\beta}_1)$ can be derived by solving the following equations

$$\begin{aligned}
\frac{\partial \text{RSS}}{\partial \beta_0} &= 0, \\
\frac{\partial \text{RSS}}{\partial \beta_1} &= 0.
\end{aligned}$$

Solution:

$$\begin{aligned}
\frac{\partial \text{RSS}}{\partial \beta_0} &= -2 \sum_i (y_i - (\beta_0 + \beta_1 x_i)) = 0, \\
\frac{\partial \text{RSS}}{\partial \beta_1} &= -2 \sum_i x_i (y_i - (\beta_0 + \beta_1 x_i)) = 0
\end{aligned}$$

$$\begin{aligned}
\sum_i y_i &= n\beta_0 + \beta_1 \sum_i x_i, \\
\sum_i x_i y_i &= -2 \sum_i x_i (y_i - (\beta_0 + \beta_1 x_i)) = 0
\end{aligned}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$
$$\hat{\beta}_1 = \frac{n \sum_i x_i y_i - (\sum_i x_i)(\sum_i y_i)}{n \sum_i x_i^2 - (\sum_i x_i)^2}$$

Using the following equalities proves the result.

$$\sum_i (x_i - \bar{x})^2 = \sum_i x_i^2 - n\bar{x}^2$$
$$\sum_i (x_i - \bar{x})(y_i - \bar{y}) = \sum_i x_i y_i - n\bar{x}\bar{y}$$

Exercise 2

Do Exercise 5 in Chapter 3.7 of ISLR.

We have $\hat{y}_i = x_i \hat{\beta}$ and $\hat{\beta} = (\sum_{i=1}^n x_i y_i)/(\sum_{i'=1}^n x_i'^2)$

$$\begin{aligned}
\hat{y}_i &= x_i \frac{\sum_{i=1}^n x_i y_i}{\sum_{i'=1}^n x_i'^2} && (1)\\
&= x_i \frac{\sum_{i'=1}^n x_{i'} y_{i'}}{\sum_{k=1}^n x_k^2} && (2)\\
&= \sum_{i'=1}^n \frac{x_i x_{i'}}{\sum_{k=1}^n x_k^2} y_{i'} && (3)\\
& && (4)
\end{aligned}$$

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

Multiple Linear Regression

Exercise 4

Do Exercise 3 in Chapter 3.7 of ISLR.

Salary = 50 + 20 GPA + 0.07 IQ + 35 Gender + 0.01 (GPA * IQ) - 10 (GPA * Gender)

- (a)

Male: (Gender = 0)

Salary = 50 + 20 GPA_FIXED + 0.07 IQ_FIXED + 0.01 (GPA_FIXED * IQ_FIXED)

Female: (Gender = 1)

Salary = 50 + 20 GPA_FIXED + 0.07 IQ_FIXED + 0.01 (GPA_FIXED * IQ_FIXED) + 35 - 10 GPA_FIXED

When GPA_FIXED > 3.5, males earn more than females on average (iii).

- (b) Gender = 1, IQ = 110, GPA = 4.0

Salary = 50 + 20 * 4 + 0.07 * 110 + 35 + 0.01 (4 * 110) - 10 * 4 = 137.1

- (c)

False. We must examine the p-value of the regression coefficient to determine if the interaction term is statistically significant or not.

Exercise 5

Read and run the code in Sections 3.6.1 to 3.6.6 of ISLR. The goal is to understand the different R functions to fit and analyze linear models.

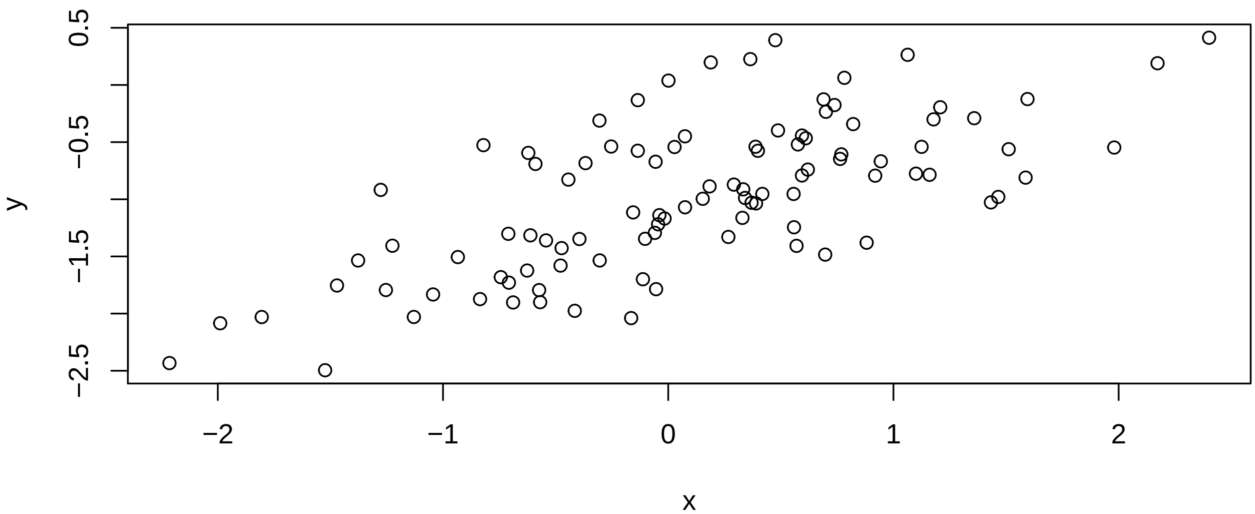
Exercise 6

Do Exercise 13 in Chapter 3.7 of ISLR.

```
set.seed(1)
x <- rnorm(100)
eps <- rnorm(100, 0, sqrt(0.25))
y <- -1 + 0.5*x + eps

print(length(y))
# [1] 100

plot(x,y)
```

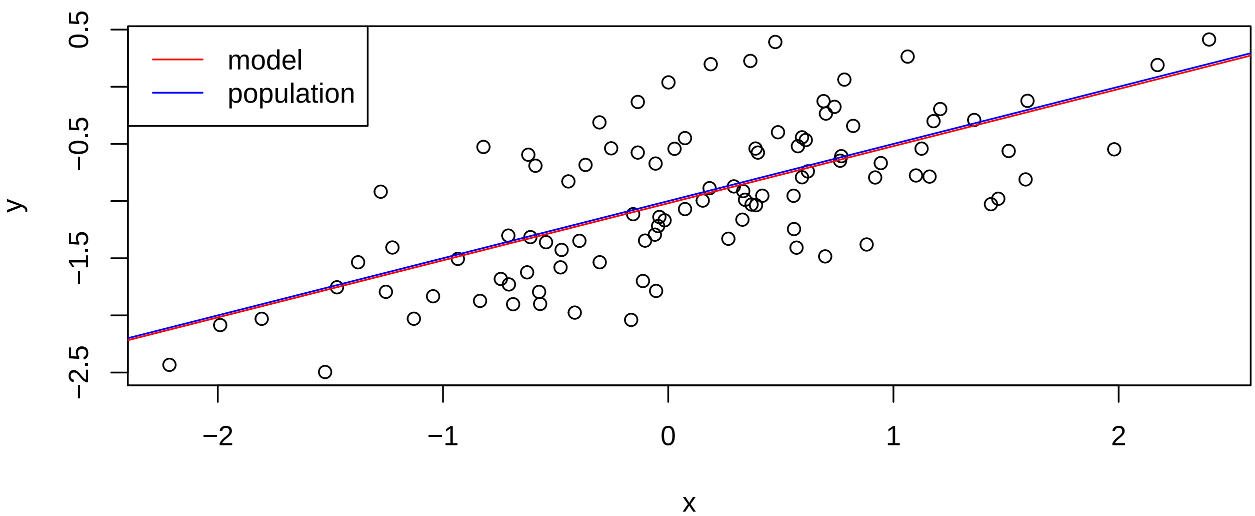



```
lm.fit <- lm(y~x)
summary(lm.fit)
#
# Call:
# lm(formula = y ~ x)
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -0.93842 -0.30688 -0.06975  0.26970  1.17309
#
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) -1.01885    0.04849 -21.010  < 2e-16 ***
# 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 linear regression fits a model close to the true value of the coefficients. The model has a large F-statistic with a near-zero p-value so the null hypothesis can be rejected.

```
plot(x, y)
abline(lm.fit, col = 'red')
abline(-1, 0.5, col = 'blue')
legend('topleft', legend = c('model', 'population'), col = c('red', 'blue'), lty = 1)
```

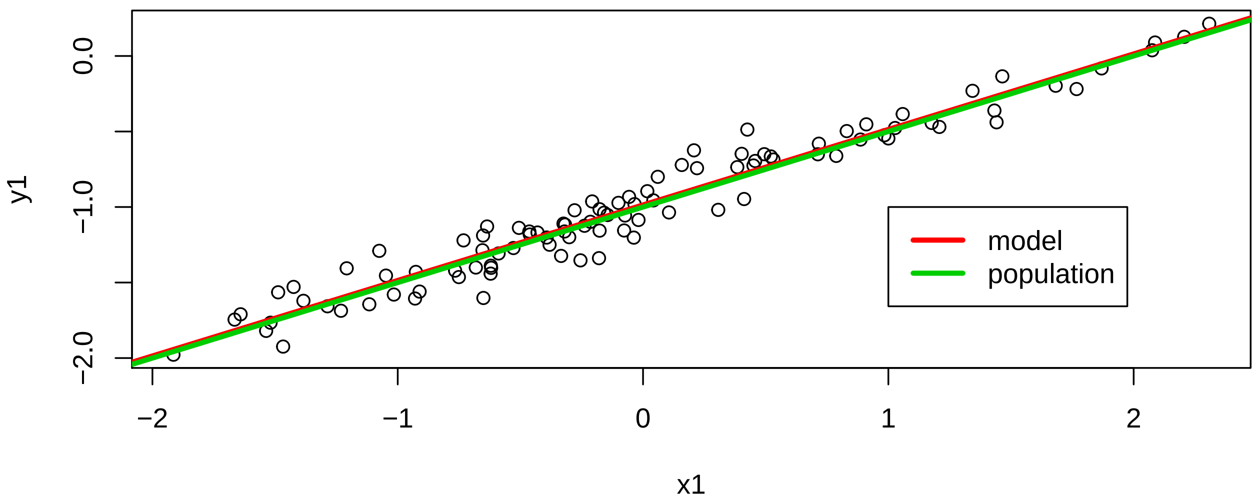



```
lm.fit_sq = lm(y~x+I(x^2))
summary(lm.fit_sq)
#
# Call:
# lm(formula = y ~ x + I(x^2))
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -0.98252 -0.31270 -0.06441  0.29014  1.13500
#
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) -0.97164    0.05883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

There is evidence that model fit is better given the slight increase in R2 and RSE. However, the p-value of the t-statistic suggests that there isn't a relationship between y and x^2.

```
set.seed(1)
eps1 = rnorm(100, 0, 0.125)
x1 = rnorm(100)
```

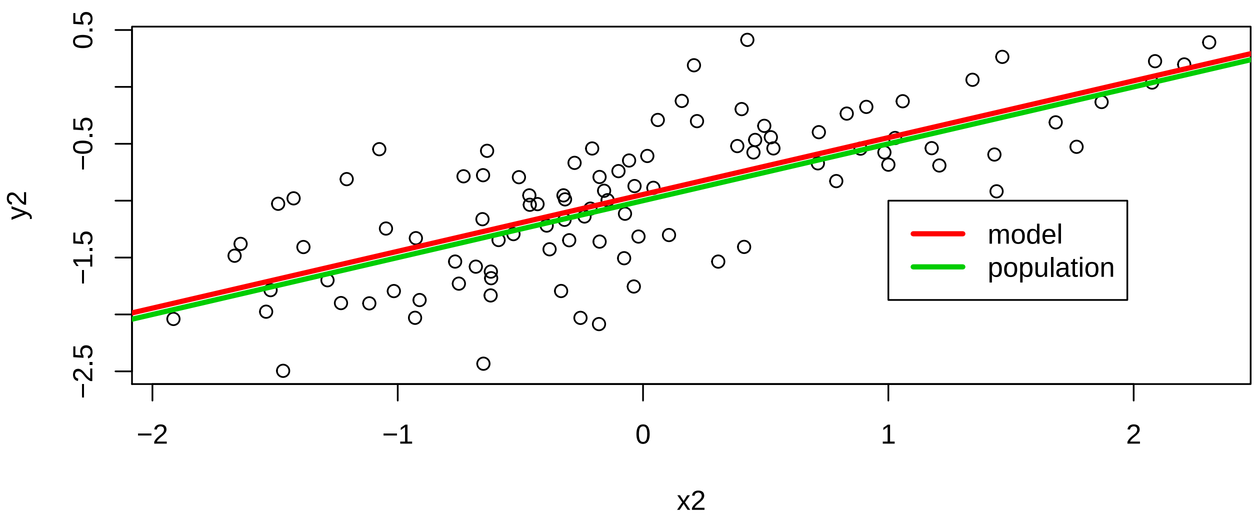
```r
y1 = -1 + 0.5*x1 + eps1
plot(x1, y1)
lm.fit1 = lm(y1~x1)
summary(lm.fit1)
#
# Call:
# lm(formula = y1 ~ x1)
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -0.29052 -0.07545  0.00067  0.07288  0.28664
#
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) -0.98639    0.01129  -87.34   <2e-16 ***
# x1           0.49988    0.01184   42.22   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.1128 on 98 degrees of freedom
# Multiple R-squared:  0.9479,  Adjusted R-squared:  0.9474
# F-statistic:  1782 on 1 and 98 DF,  p-value: < 2.2e-16
abline(lm.fit1, lwd=3, col=2)
abline(-1, 0.5, lwd=3, col=3)
legend(-1, legend = c("model", "population"), col=2:3, lwd=3)
```



As expected, the R^2 and RSE decreases considerably.

```r
set.seed(1)
eps2 = rnorm(100, 0, 0.5)
x2 = rnorm(100)
y2 = -1 + 0.5*x2 + eps2
plot(x2, y2)
lm.fit2 = lm(y2~x2)
summary(lm.fit2)
#
# Call:
# lm(formula = y2 ~ x2)
```

```
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -1.16208 -0.30181  0.00268  0.29152  1.14658
#
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) -0.94557    0.04517  -20.93   <2e-16 ***
# x2           0.49953    0.04736   10.55   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.4514 on 98 degrees of freedom
# Multiple R-squared:  0.5317,  Adjusted R-squared:  0.5269
# F-statistic: 111.2 on 1 and 98 DF,  p-value: < 2.2e-16
abline(lm.fit2, lwd=3, col=2)
abline(-1, 0.5, lwd=3, col=3)
legend(-1, legend = c("model", "population"), col=2:3, lwd=3)
```



As expected, the R^2 and RSE increases considerably.

```
confint(lm.fit)
#                  2.5 %     97.5 %
# (Intercept) -1.1150804 -0.9226122
# x            0.3925794  0.6063602
confint(lm.fit1)
#                 2.5 %     97.5 %
# (Intercept) -1.008805 -0.9639819
# x1           0.476387  0.5233799
confint(lm.fit2)
#                  2.5 %     97.5 %
# (Intercept) -1.0352203 -0.8559276
# x2           0.4055479  0.5935197
```

All intervals seem to be centered on approximately 0.5, with the second fit's interval being narrower than the first fit's interval and the last fit's interval being wider than the first fit's interval.

Exercise 7

Do Exercise 15 in chapter 3.7 of ISLR.

```r
library(MASS)
summary(Boston)
#       crim                zn             indus            chas
#  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
#  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
#  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
#  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
#  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
#  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
#       nox               rm             age              dis
#  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
#  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
#  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
#  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
#  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
#  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
#       rad              tax           ptratio          black
#  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
#  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
#  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
#  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
#  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
#  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
#      lstat            medv
#  Min.   : 1.73   Min.   : 5.00
#  1st Qu.: 6.95   1st Qu.:17.02
#  Median :11.36   Median :21.20
#  Mean   :12.65   Mean   :22.53
#  3rd Qu.:16.95   3rd Qu.:25.00
#  Max.   :37.97   Max.   :50.00
Boston$chas <- factor(Boston$chas, labels = c("N","Y"))
X <- Boston[-1]
crim <- Boston$crim
coefs <- numeric(length(X))
for (i in seq_along(X)) {
  pred <- X[, i]
  name <- colnames(X)[i]
  model_summary <- summary(lm(crim ~ pred))

  print(model_summary$coefficients[2, 4])
}
# [1] 5.506472e-06
# [1] 1.450349e-21
# [1] 0.2094345
# [1] 3.751739e-23
# [1] 6.346703e-07
# [1] 2.854869e-16
# [1] 8.519949e-19
# [1] 2.693844e-56
# [1] 2.357127e-47
# [1] 2.942922e-11
```

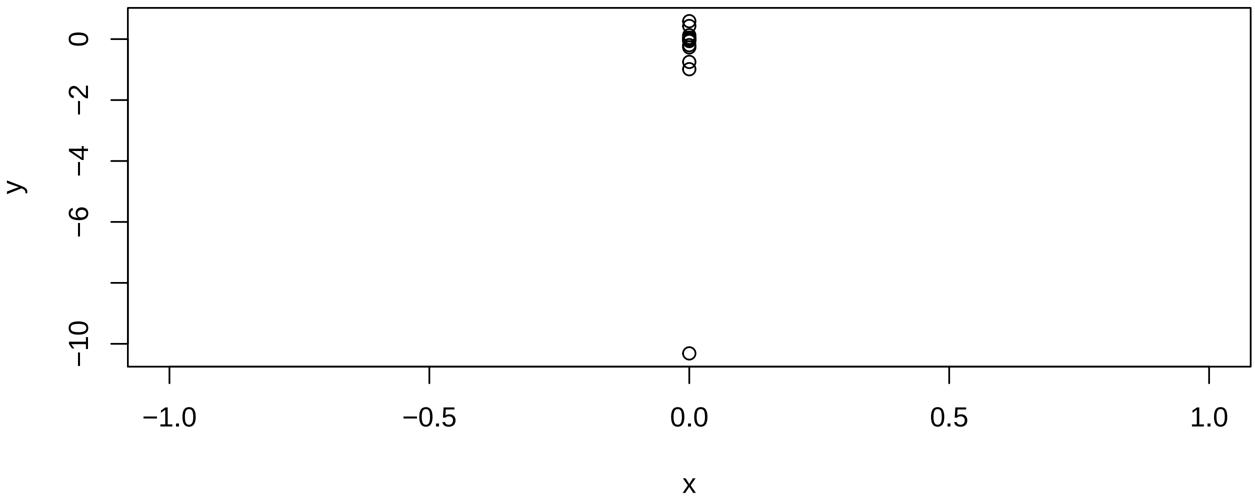
```
# [1] 2.487274e-19
# [1] 2.654277e-27
# [1] 1.173987e-19
```

All, except chas.

```
lm.all = lm(crim~., data=Boston)
summary(lm.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
# chasY        -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
```

zn, dis, rad, black, medv

```
x <- coefs
y <- coefficients(lm.all)[-1]
plot(x,y)
```



There is a large difference for the coefficient of nox.

```
for (i in seq_along(X)) {
  pred <- X[, i]
  name <- colnames(X)[i]
  if(name != "chas"){
    print(name)
    model <- lm(crim~poly(pred,3))
    model_summary <- summary(lm(crim ~ poly(pred,3)  ))

    pvalues <- model_summary$coefficients[3:4, 4]

    res <- ifelse(sum(pvalues < 0.05), "YES", "NO")
    print(sprintf('Predictor %s : significant (at 5 percent) non-linear relationship? = %s', name, res))
  }
}
# [1] "zn"
# [1] "Predictor zn : significant (at 5 percent) non-linear relationship? = YES"
# [1] "indus"
# [1] "Predictor indus : significant (at 5 percent) non-linear relationship? = YES"
# [1] "nox"
# [1] "Predictor nox : significant (at 5 percent) non-linear relationship? = YES"
# [1] "rm"
# [1] "Predictor rm : significant (at 5 percent) non-linear relationship? = YES"
# [1] "age"
# [1] "Predictor age : significant (at 5 percent) non-linear relationship? = YES"
# [1] "dis"
# [1] "Predictor dis : significant (at 5 percent) non-linear relationship? = YES"
# [1] "rad"
# [1] "Predictor rad : significant (at 5 percent) non-linear relationship? = YES"
# [1] "tax"
# [1] "Predictor tax : significant (at 5 percent) non-linear relationship? = YES"
# [1] "ptratio"
# [1] "Predictor ptratio : significant (at 5 percent) non-linear relationship? = YES"
# [1] "black"
# [1] "Predictor black : significant (at 5 percent) non-linear relationship? = NO"
# [1] "lstat"
# [1] "Predictor lstat : significant (at 5 percent) non-linear relationship? = YES"
```

```
# [1] "medv"
# [1] "Predictor medv : significant (at 5 percent) non-linear relationship? = YES"
```