Chapter-03-lab

Fabiani Rafael

2025-02-25

Math 448 Chapter 03 Lab

This question involves the use of simple linear regression on the Auto data.

1) Perform regression with mpg as the response and horsepower as the predictor.Comment the output. For example

```
I.      Any relationship between the predictor and the response.

II.         Interpret each of the estimates of coefficients.

III.        Is the relationship between the predictor and the response positive or negative?

IV.         What is the predicted mpg associated with a horsepower of 98? What are the associated
```

```r
attach(Auto)
lm.fit=lm(mpg~horsepower)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower)
##
## 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
```

```r
#Call:
#lm(formula = mpg ~ horsepower)
#
#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. There is a negative relationship between the predictor and the response.

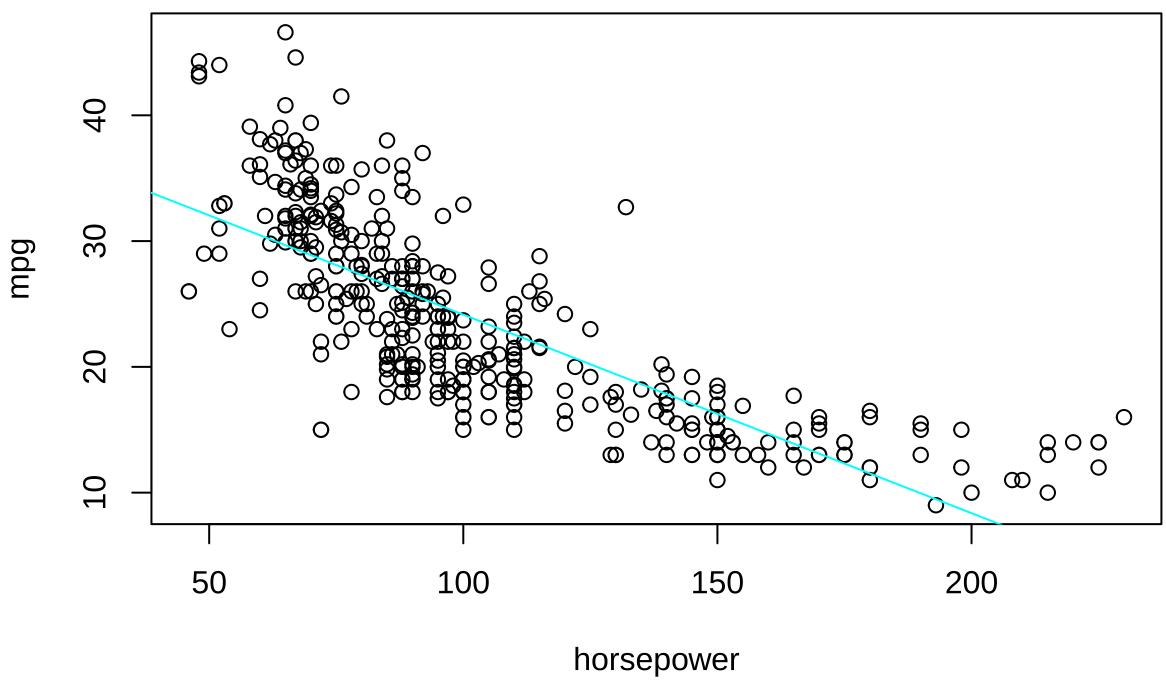
–II. The coefficient of horsepower is -0.157845. This means that for every one unit increase in horsepower, the mpg decreases by 0.157845.

–III. The relationship between the predictor and the response is negative.

–IV. The predicted mpg associated with a horsepower of 98 is 24.46708. The 95% confidence interval for the average mpg is (23.25274, 25.68142) and the 95% prediction interval for the exact mpg is (9.409285, 39.52487).

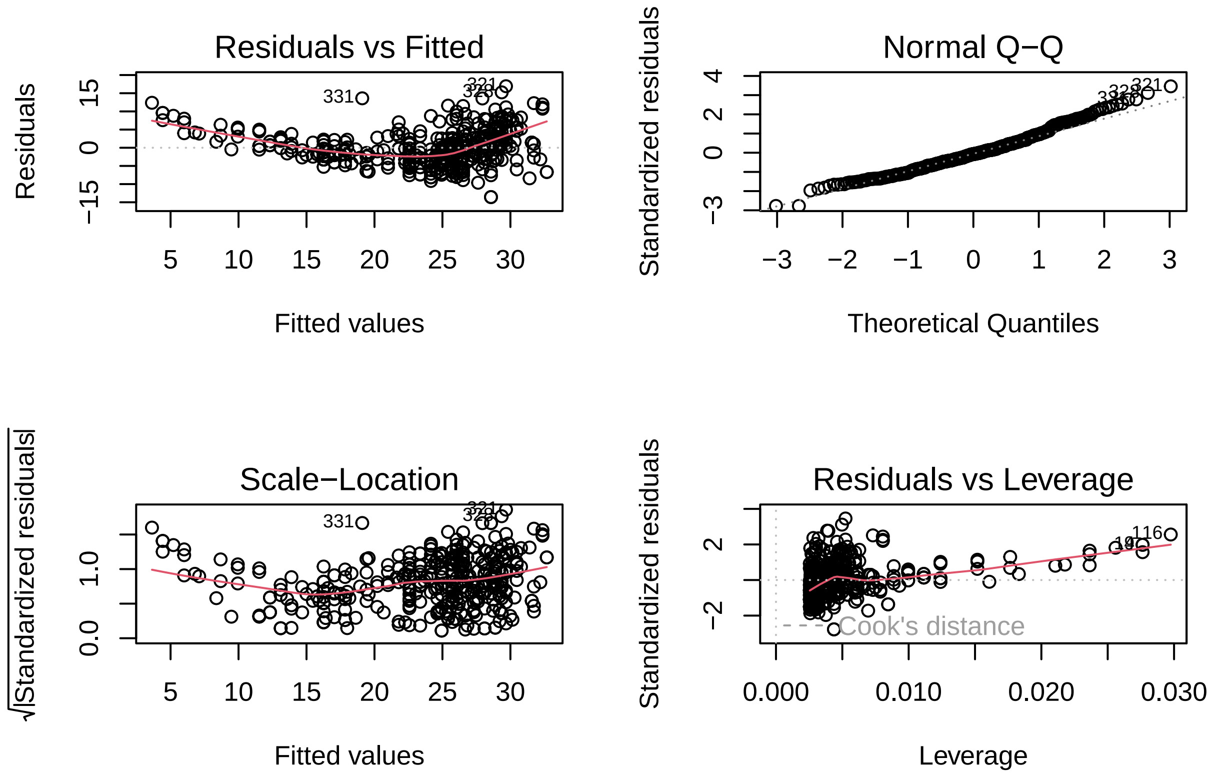
2) Plot the response and the predictor. Display the least squares regression line.

```
plot(horsepower,mpg)
abline(lm.fit, col="cyan")
```



3) Produce diagnostic plots of the least square regression fit. Comment on any problems you see with the fit.

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



–The residuals vs fitted plot shows a pattern that doesnt appear to be entirely random. This could mean that the relationship between the predictor and the response is non linear. From the Q-Q plot, the residuals don't look normally distributed. The scale-location plot shows that the residuals have different variance values. Lastly the residuals vs leverage plot shows that there are some outliers that could be influencing the model.

##This question involves a simple implement of KNN regression. 1) Create a function f(x) calculating -x^2+2*x.

```
f <- function(x) {## function to calc x^2+2x
  return(-x^2 + 2*x)
}
```

2) Generate a sequence 0,0.1,0.2,. . . ,1.5 and name it as x. Count the length and name it as n.

```
x <- seq(0,1.5,0.1)
n <- length(x)
```

3) Set seed as "448". R: set.seed(448) Python: import numpy as np; np.random.seed(448)

```
set.seed(448)
```

4) Generate n random numbers from normal distribution with mean 0 and standard deviation 0.1 and save it as error.

```
error <- rnorm(n, mean=0, sd=0.1)
```

5) Generate a n by 1 vector y using relation y=f(x)+error.

```
y <- f(x) + error
print(y)
```

```
##  [1]  0.006034717 -0.011762504  0.367778649  0.649463568  0.711163735
##  [6]  0.730006386  0.907004952  0.985899345  0.937757190  0.865327759
## [11]  1.042369518  0.938892315  0.881746968  0.783449924  0.855340785
```

```
## [16]  0.835842963
```

6) Create a function knn.pred with two augments: K and an observation x0. This function will fit a KNN regression with parameter K based on (x,y) and predict response for the observation x0. Test your function using K=5 and x0=1.
7) Use the function you created in 6) to calculate the training MSE for the data (x,y) we have. (Set K=5)
8) Use your code in 7) to create a function with one argument K. The output is the training MSE of data.
9) Make a plot to see the relation between 1/K and the training MSE.

Suppose we have a test data set where x_test=c(0.05,0.15,…,1.45), generate random error error_test and y_test using the true relationship. Can you make a plot to see the relationship between 1/K and the test MSE?