**EXPERIMENT NO.-2**

**AIM: Implement the code using 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.**
**b. Plot the response and the predictor. Use the abline() function to display the least squares regression line.**
**c. Use the plot() function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.**
**d. Produce a scatterplot matrix which includes all of the variables in the data set.**

**(2.a) AIM: Use the lm() function to perform a simple linear regression with mpg as the response and horsepower as the predictor.**

**DESCRIPTION:** The library() function is used to load libraries, or groups of functions and library() data sets that are not included in the base R distribution. Basic functions that perform least squares linear regression and other simple analyses come standard with the base distribution, but more exotic functions require additional libraries. Here we load the MASS package, which is a very large collection of data sets and functions. . Alternatively, this can be done at the R command line via install.packages("ISLR").The lm() function to fit a simple linear regression lm() model, with medv as the response and lstat as the predictor. The basic syntax is lm(y∼x,data), where y is the response, x is the predictor, and data is the data set in which these two variables are kept

**CODE:**

```
library(ISLR)
library(MASS)
data("auto")
head(auto)
lm.fit<-lm(mpg~horsepower,data=Auto)
summary(lm.fit)
```

**OUTPUT:**

```
  mpg cylinders displacement horsepower weight acceleration year
1  18         8          307        130   3504         12.0   70
2  15         8          350        165   3693         11.5   70
3  18         8          318        150   3436         11.0   70
4  16         8          304        150   3433         12.0   70
5  17         8          302        140   3449         10.5   70
6  15         8          429        198   4341         10.0   70
  origin                       name
1      1 chevrolet chevelle malibu
2      1         buick skylark 320
3      1        plymouth satellite
4      1             amc rebel sst
5      1               ford torino
6      1          ford galaxie 500
```

```
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
Multiple R-squared:  0.6059,    Adjusted R-squared:  0.6049
F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

**CODE:**

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

**OUTPUT:**

```
> predict(lm.fit,data.frame(horsepower=c(98)),interval="prediction")
       fit     lwr      upr
1 24.46708 14.8094 34.12476
> predict(lm.fit,data.frame(horsepower=c(98)),interval="confidence")
       fit      lwr      upr
1 24.46708 23.97308 24.96108
```

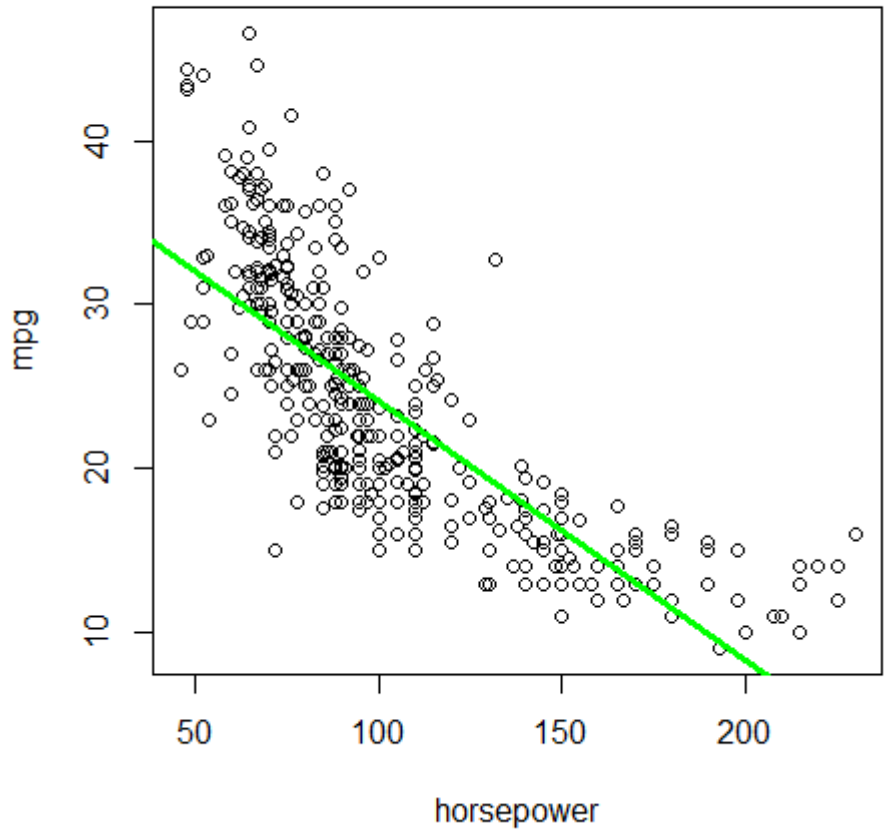
**(2.b) AIM : Plot the response and the predictor. Use the abline() function to display the least squares regression line.**

**DESCRIPTION:** The abline() function can be used to draw any line, not just the least squares regression line. To draw a line with intercept a and slope b, we type abline(a,b). Below we experiment with some additional settings for plotting lines and points. The lwd=3 command causes the width of the regression line to be increased by a factor of 3; this works for the plot() and lines() functions also.

**CODE:**

```
attach(Auto)
plot(horsepower,mpg)
abline(lm.fit,lwd=5,col="blue")
```

**OUTPUT:**



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

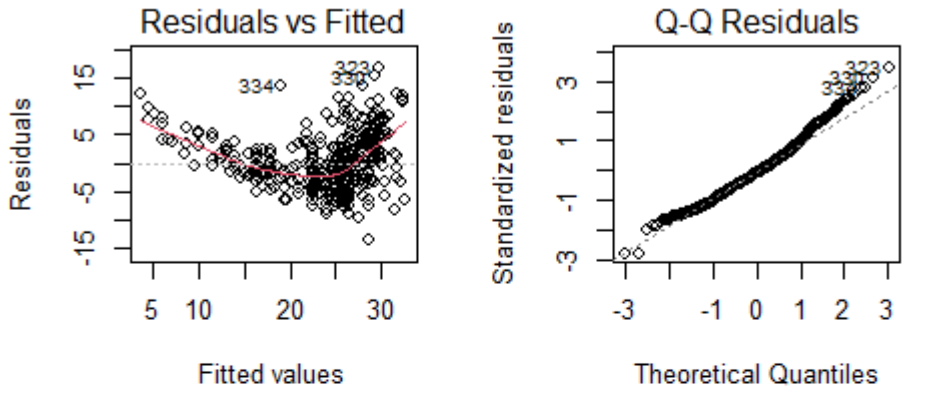
**DESCRIPTION:** Plot() function in R programming Language is defined as a generic function for plotting. **The plot() function in R is "used to create a plot". It has many options and arguments to control many things, such as the plot type, labels, titles, and colors. For example, plot(x, y) creates a scatter plot of x and y numeric vectors.**

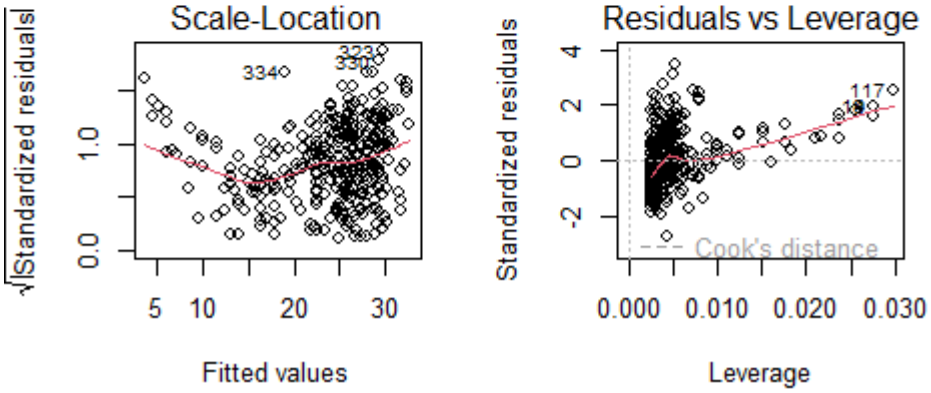
**CODE:**

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

**OUTPUT:**

```
> which.max(hatvalues(lm.fit))
117
116
```



**(2.d) AIM: . Produce a scatterplot matrix which includes all of the variables in the data set.**

**DESCRIPTION:** A scatterplot is a set of dotted points representing individual data pieces on the horizontal and vertical axis.In a graph in which the values of two variables are plotted along the axis X and Y,the pattern of the resulting points reveals a correlation between them.We can create a scatter plot in R programming Language using the plot() function.

**CODE:**
Pairs(Auto)

**OUTPUT:**