Regresión Lineal Simple SLR1

Santiago Sanchez Varela

2024-11-02

Descripción

Este ejercicio tiene como objetivo revisar el uso de R para generar un modelo de regresión lineal y comprender los principales conceptos que se desprenden del modelamiento.

Caso

Para este ejemplo se utilizará como fuente de información la librería ISLR2 que contiene el data set Boston. Éste contiene 506 observaciones de censos realizados. Se busca predecir medv (median house value) utilizando alguno de los 12 predictores restantes. Para este ejercicio se utilizará lstat (percent of households with low socioeconomic status). Para lo anterior se usará la función de R lm(). La sintáxis básica es lm(y ~ x, data), donde y es la respuesta, x es el predictor, y data es el conjunto de datos donde se mentienen las dos variables.

Esto se puede presentar de la siguiente manera:

$Y = \beta_0 + \beta_1 X$

```
library(MASS)
library(ISLR2)
```

```
##
## Adjuntando el paquete: 'ISLR2'
```

```
## The following object is masked from 'package:MASS':
##
##     Boston
```

```
head(Boston)
```

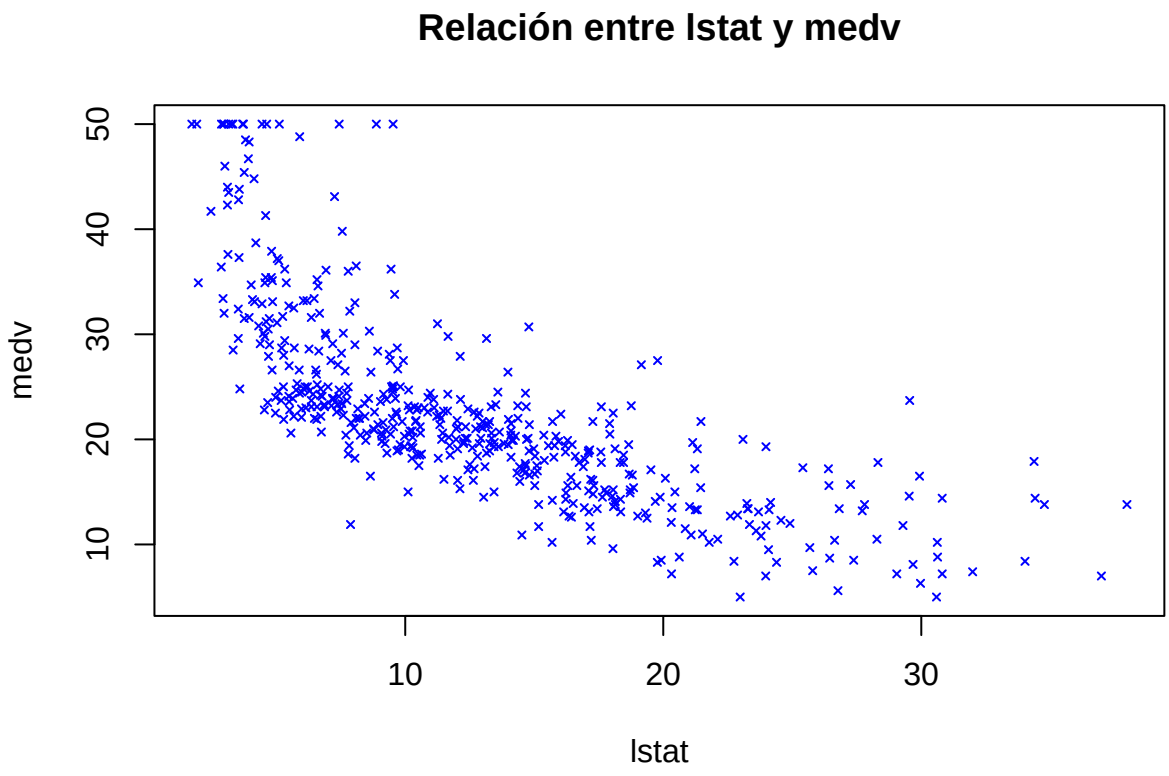
```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
```

```
attach(Boston)
```

Gráfico medv/lstat

A continuación se puede observar como se presentan los datos asociados a las dos variables mecionadas en el caso

```
plot(lstat, medv, pch = 4, col = "blue", cex = 0.5, main = "Relación entre lstat y medv")
```




```
lm.fit <- lm(medv ~ lstat)
lm.fit
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
## Coefficients:
## (Intercept)        lstat
##       34.55        -0.95
```

```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
names(lm.fit)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
coef(lm.fit)
```

```
## (Intercept)       lstat
##  34.5538409  -0.9500494
```

```
confint(lm.fit)
```

```
##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505
```

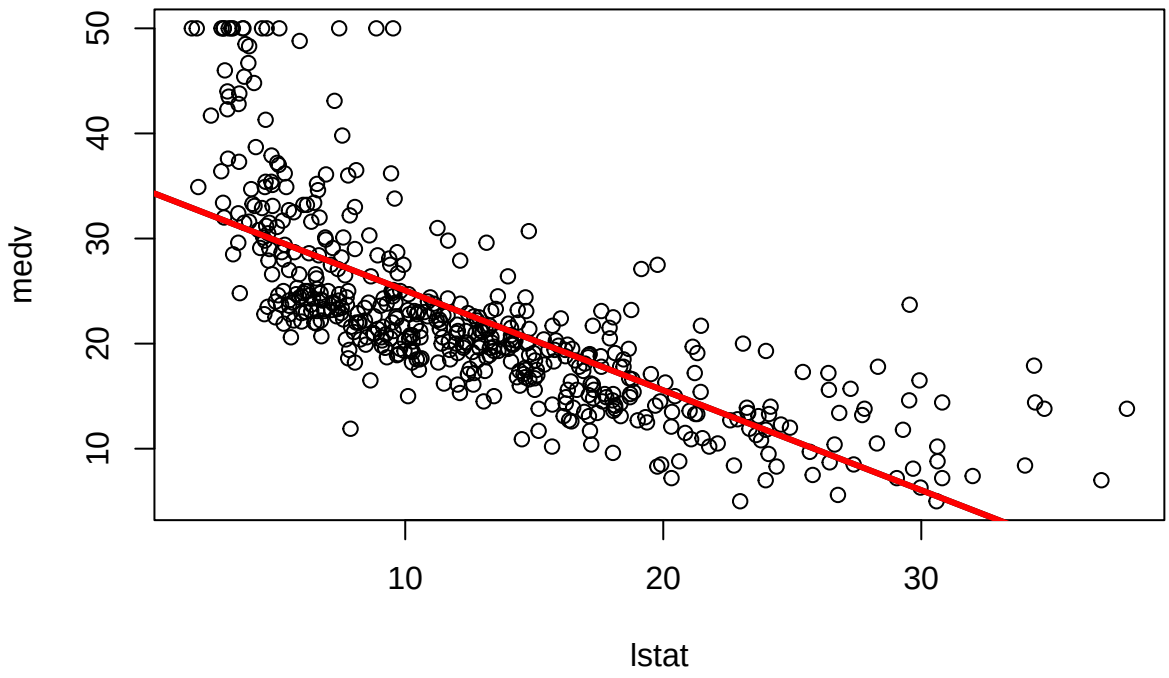
```
predict(lm.fit, data.frame(lstat = (c(5, 10, 15))), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

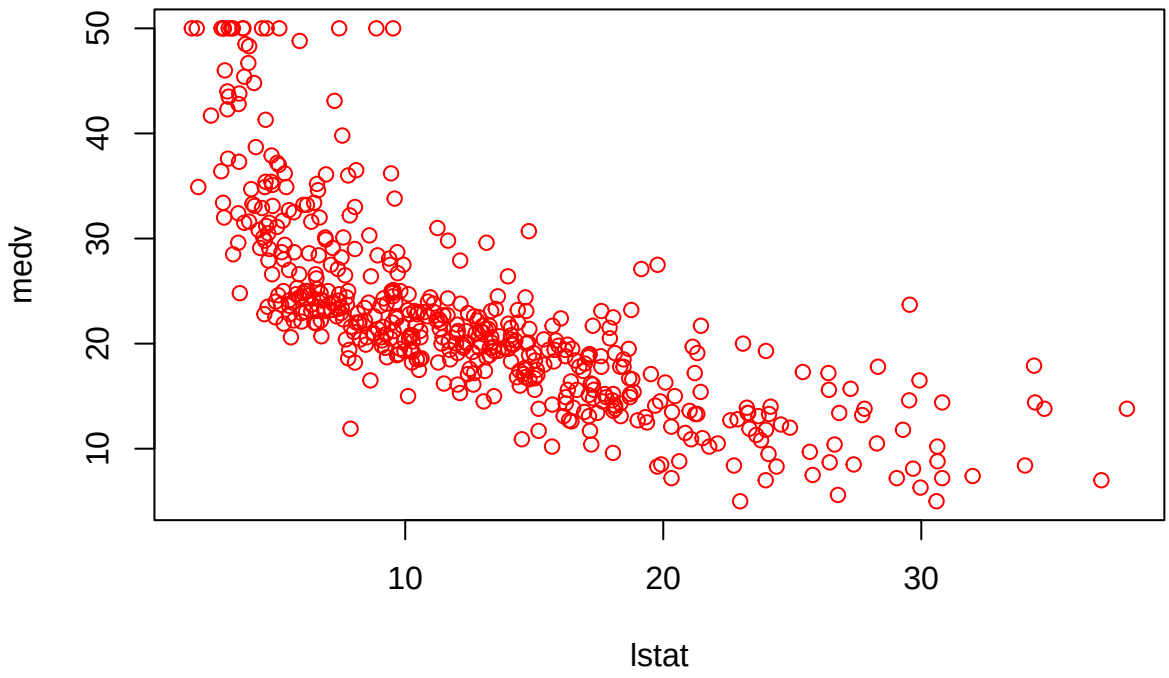
```
predict(lm.fit, data.frame(lstat = (c(5, 10, 15))), interval = "prediction")
```

```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

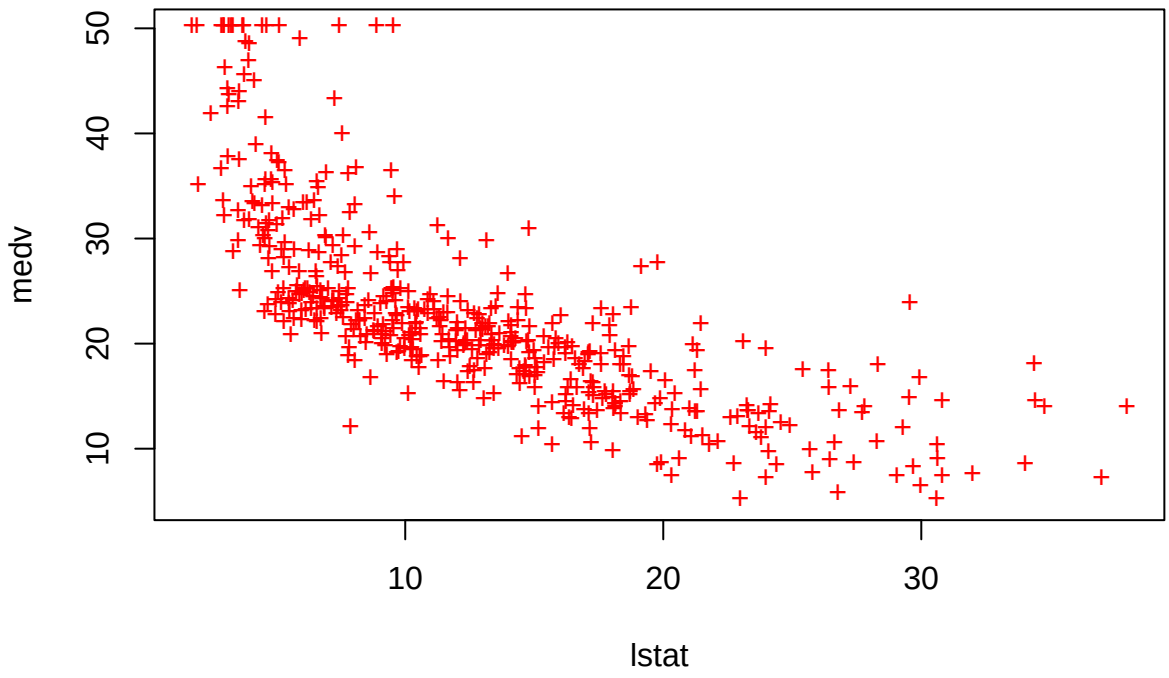
```
plot(lstat, medv)
abline(lm.fit)

abline(lm.fit, lwd = 3)
abline(lm.fit, lwd = 3, col = "red")
```



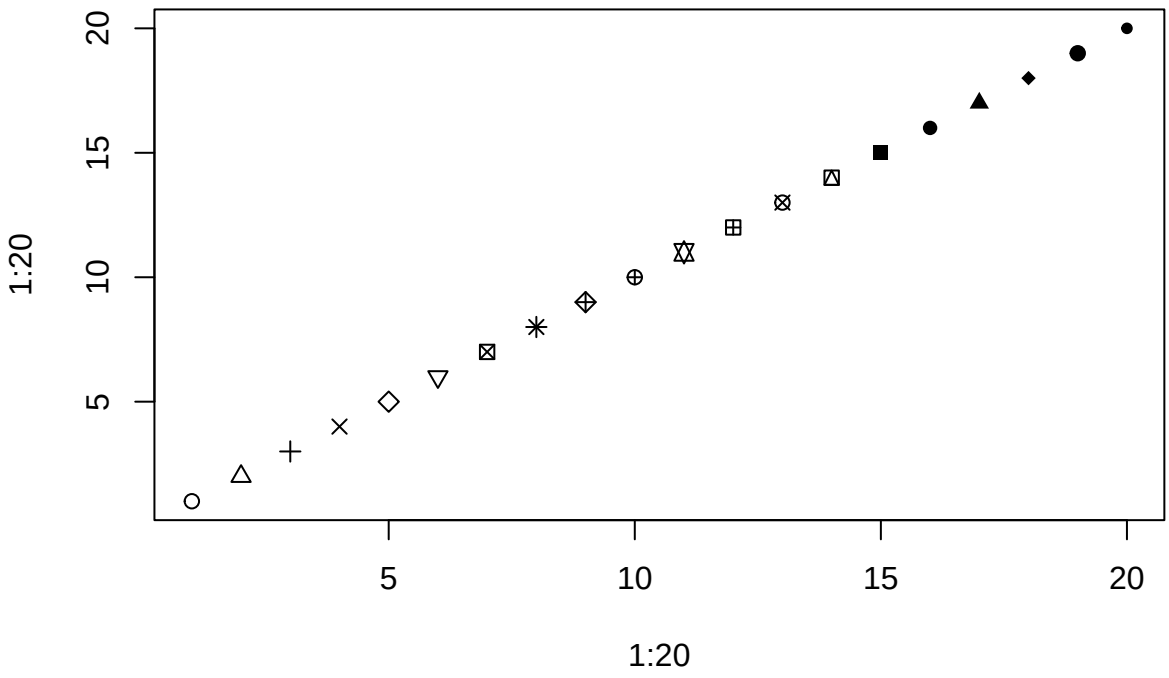

```
plot(lstat, medv, col = "red")
```



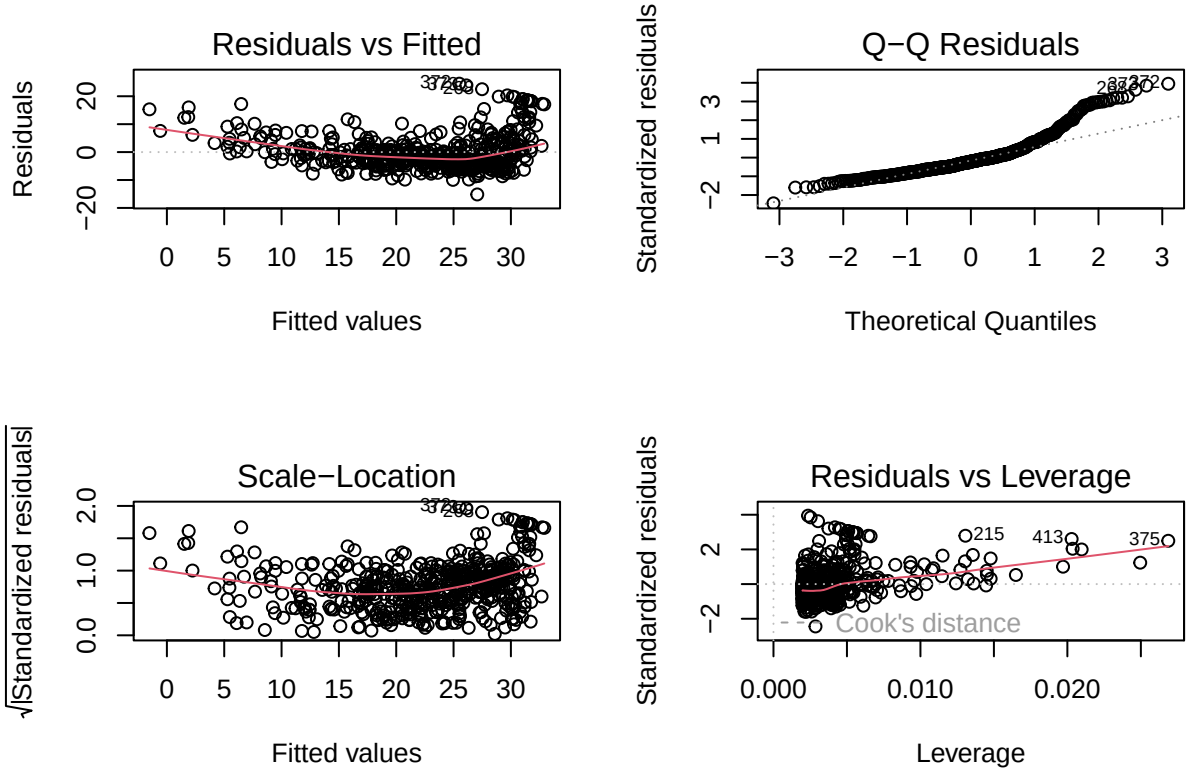

```
plot(lstat, medv, pch = "+", col = "red")
```




```
plot(1:20, 1:20, pch = 1:20)
```

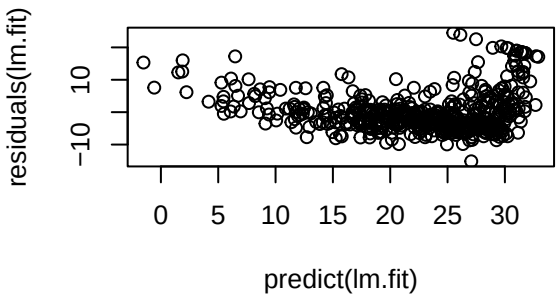


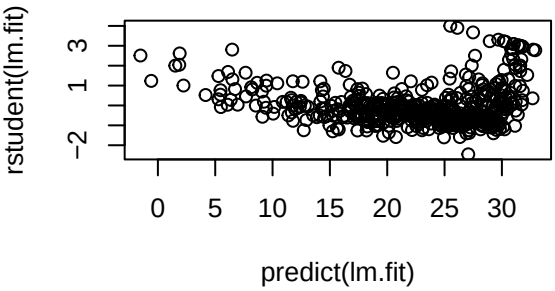

```
par(mfrow = c(2,2))
plot(lm.fit)
```

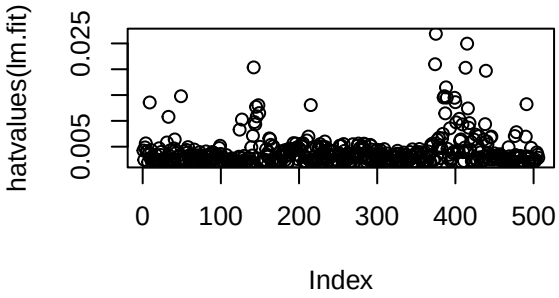



```
plot(predict(lm.fit), residuals(lm.fit))
plot(predict(lm.fit), rstudent(lm.fit))
plot(hatvalues(lm.fit))
which.max(hatvalues(lm.fit))
```

```
## 375
## 375
```


residuals(lm.fit)
−10
10
0
5
10
15
20
25
30
predict(lm.fit)


rstudent(lm.fit)
−2
1
3
0
5
10
15
20
25
30
predict(lm.fit)


hatvalues(lm.fit)
0.005
0.025
0
100
200
300
400
500
Index