LaboratorioR

Alberto Armijo Ruiz

19 de noviembre de 2018

```r
require(MASS)
```

```
## Loading required package: MASS
```

```r
require(ISLR)
```

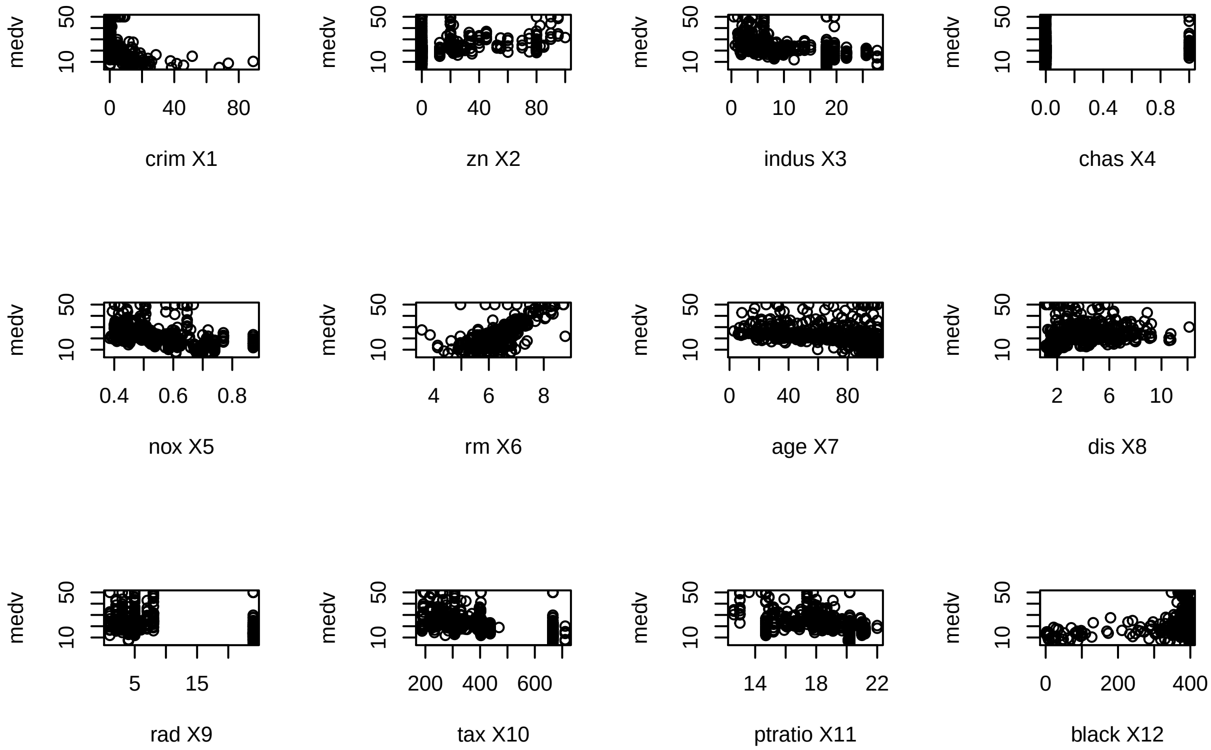
```
## Loading required package: ISLR
```

```r
?Boston
attach(Boston)
lstat
```

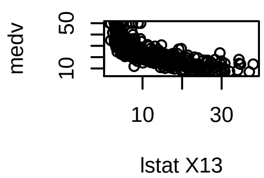
```
##   [1]  4.98  9.14  4.03  2.94  5.33  5.21 12.43 19.15 29.93 17.10 20.45
##  [12] 13.27 15.71  8.26 10.26  8.47  6.58 14.67 11.69 11.28 21.02 13.83
##  [23] 18.72 19.88 16.30 16.51 14.81 17.28 12.80 11.98 22.60 13.04 27.71
##  [34] 18.35 20.34  9.68 11.41  8.77 10.13  4.32  1.98  4.84  5.81  7.44
##  [45]  9.55 10.21 14.15 18.80 30.81 16.20 13.45  9.43  5.28  8.43 14.80
##  [56]  4.81  5.77  3.95  6.86  9.22 13.15 14.44  6.73  9.50  8.05  4.67
##  [67] 10.24  8.10 13.09  8.79  6.72  9.88  5.52  7.54  6.78  8.94 11.97
##  [78] 10.27 12.34  9.10  5.29  7.22  6.72  7.51  9.62  6.53 12.86  8.44
##  [89]  5.50  5.70  8.81  8.20  8.16  6.21 10.59  6.65 11.34  4.21  3.57
## [100]  6.19  9.42  7.67 10.63 13.44 12.33 16.47 18.66 14.09 12.27 15.55
## [111] 13.00 10.16 16.21 17.09 10.45 15.76 12.04 10.30 15.37 13.61 14.37
## [122] 14.27 17.93 25.41 17.58 14.81 27.26 17.19 15.39 18.34 12.60 12.26
## [133] 11.12 15.03 17.31 16.96 16.90 14.59 21.32 18.46 24.16 34.41 26.82
## [144] 26.42 29.29 27.80 16.65 29.53 28.32 21.45 14.10 13.28 12.12 15.79
## [155] 15.12 15.02 16.14  4.59  6.43  7.39  5.50  1.73  1.92  3.32 11.64
## [166]  9.81  3.70 12.14 11.10 11.32 14.43 12.03 14.69  9.04  9.64  5.33
## [177] 10.11  6.29  6.92  5.04  7.56  9.45  4.82  5.68 13.98 13.15  4.45
## [188]  6.68  4.56  5.39  5.10  4.69  2.87  5.03  4.38  2.97  4.08  8.61
## [199]  6.62  4.56  4.45  7.43  3.11  3.81  2.88 10.87 10.97 18.06 14.66
## [210] 23.09 17.27 23.98 16.03  9.38 29.55  9.47 13.51  9.69 17.92 10.50
## [221]  9.71 21.46  9.93  7.60  4.14  4.63  3.13  6.36  3.92  3.76 11.65
## [232]  5.25  2.47  3.95  8.05 10.88  9.54  4.73  6.36  7.37 11.38 12.40
## [243] 11.22  5.19 12.50 18.46  9.16 10.15  9.52  6.56  5.90  3.59  3.53
## [254]  3.54  6.57  9.25  3.11  5.12  7.79  6.90  9.59  7.26  5.91 11.25
## [265]  8.10 10.45 14.79  7.44  3.16 13.65 13.00  6.59  7.73  6.58  3.53
## [276]  2.98  6.05  4.16  7.19  4.85  3.76  4.59  3.01  3.16  7.85  8.23
## [287] 12.93  7.14  7.60  9.51  3.33  3.56  4.70  8.58 10.40  6.27  7.39
## [298] 15.84  4.97  4.74  6.07  9.50  8.67  4.86  6.93  8.93  6.47  7.53
## [309]  4.54  9.97 12.64  5.98 11.72  7.90  9.28 11.50 18.33 15.94 10.36
## [320] 12.73  7.20  6.87  7.70 11.74  6.12  5.08  6.15 12.79  9.97  7.34
## [331]  9.09 12.43  7.83  5.68  6.75  8.01  9.80 10.56  8.51  9.74  9.29
## [342]  5.49  8.65  7.18  4.61 10.53 12.67  6.36  5.99  5.89  5.98  5.49
## [353]  7.79  4.50  8.05  5.57 17.60 13.27 11.48 12.67  7.79 14.19 10.19
## [364] 14.64  5.29  7.12 14.00 13.33  3.26  3.73  2.96  9.53  8.88 34.77
## [375] 37.97 13.44 23.24 21.24 23.69 21.78 17.21 21.08 23.60 24.56 30.63
## [386] 30.81 28.28 31.99 30.62 20.85 17.11 18.76 25.68 15.17 16.35 17.12
## [397] 19.37 19.92 30.59 29.97 26.77 20.32 20.31 19.77 27.38 22.98 23.34
```

```
## [408] 12.13 26.40 19.78 10.11 21.22 34.37 20.08 36.98 29.05 25.79 26.64
## [419] 20.62 22.74 15.02 15.70 14.10 23.29 17.16 24.39 15.69 14.52 21.52
## [430] 24.08 17.64 19.69 12.03 16.22 15.17 23.27 18.05 26.45 34.02 22.88
## [441] 22.11 19.52 16.59 18.85 23.79 23.98 17.79 16.44 18.13 19.31 17.44
## [452] 17.73 17.27 16.74 18.71 18.13 19.01 16.94 16.23 14.70 16.42 14.65
## [463] 13.99 10.29 13.22 14.13 17.15 21.32 18.13 14.76 16.29 12.87 14.36
## [474] 11.66 18.14 24.10 18.68 24.91 18.03 13.11 10.74  7.74  7.01 10.42
## [485] 13.34 10.58 14.98 11.45 18.06 23.97 29.68 18.07 13.35 12.01 13.59
## [496] 17.60 21.14 14.10 12.92 15.10 14.33  9.67  9.08  5.64  6.48  7.88
```

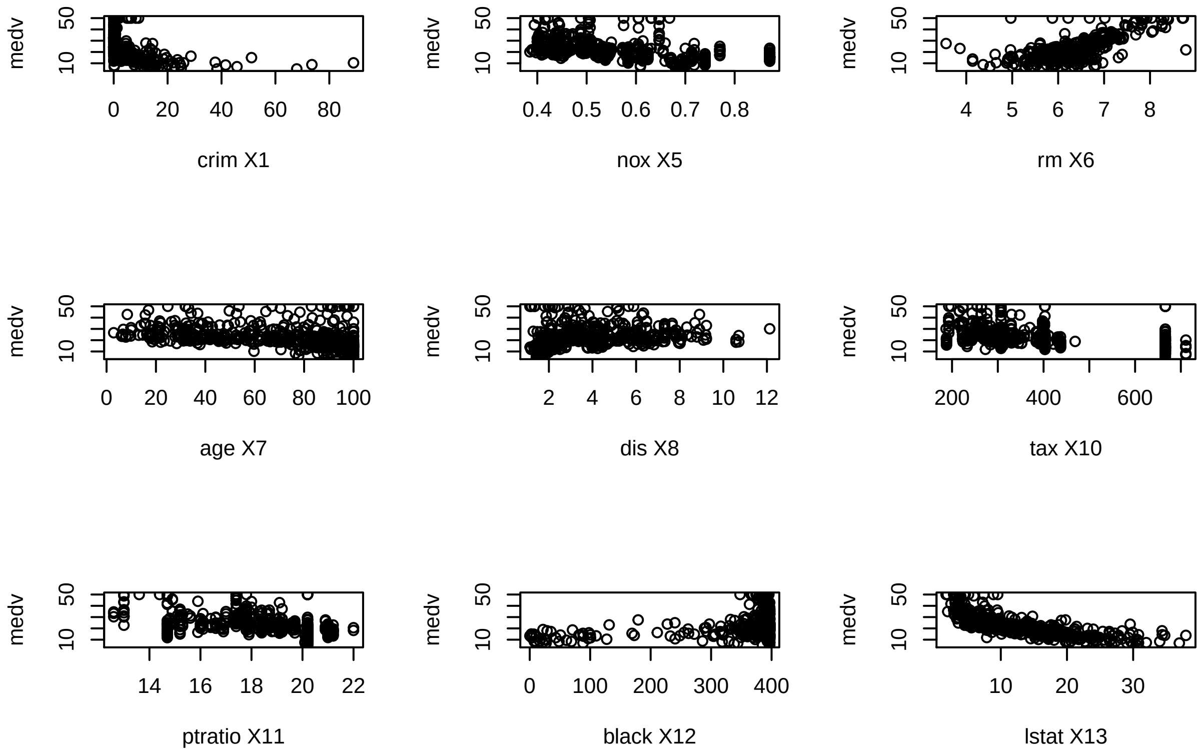
```r
temp <- Boston
plotY <- function (x,y) {
  plot(temp[,y]~temp[,x], xlab=paste(names(temp)[x]," X",x,sep=""),
       ylab=names(temp)[y])
}
par(mfrow=c(3,4))
x <- sapply(1:(dim(temp)[2]-1), plotY, dim(temp)[2])
```




```r
par(mfrow=c(1,1))
```




```r
par(mfrow=c(3,3))
x <- sapply(c(1, 5, 6, 7, 8, 10, 11, 12, 13), plotY, dim(temp)[2])
```




```r
par(mfrow=c(1,1))
```

```r
# Probamos primero con un modelo linear simple
fit1=lm(medv~lstat,data=Boston)
fit1
```

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

```r
# Probamos con otro modelo.
fit2=lm(medv~rm,data=Boston)
fit2
```

```
##
## Call:
## lm(formula = medv ~ rm, data = Boston)
##
## Coefficients:
## (Intercept)           rm
##     -34.671        9.102
```

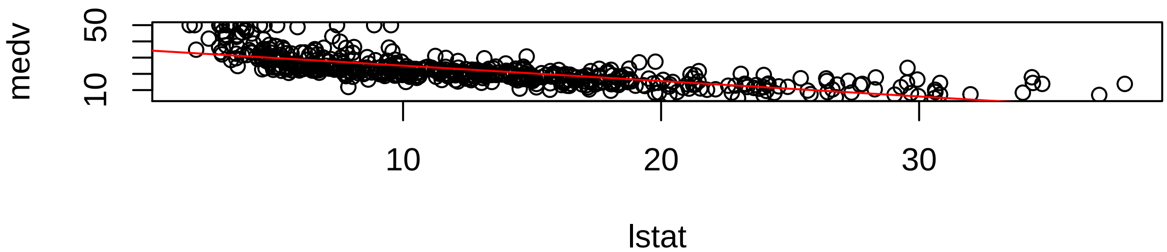
```r
# Ahora miramos la información más detallada de cada uno de los modelos.
summary(fit1)
```

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

```
##
## 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
```

```r
par(mfrow=c(2,1))
plot(medv~lstat,data=Boston)
abline(fit1,col="red")
confint(fit1)
```

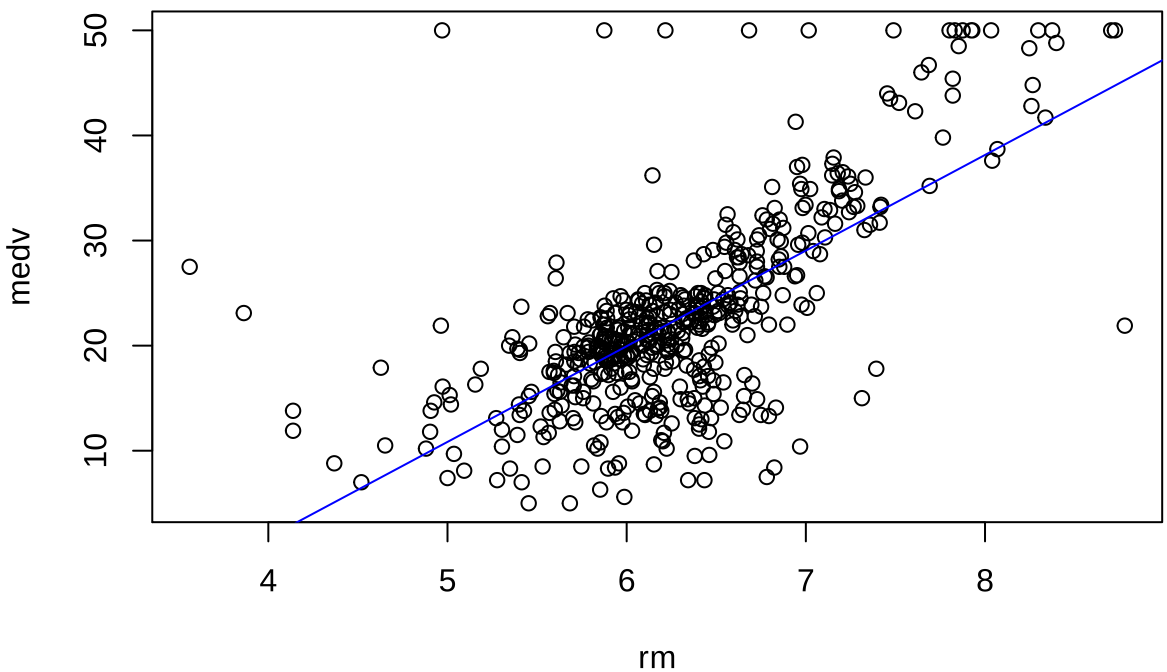
```
##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505
```




```r
# Hacemos los mismo para el modelo anterior.
summary(fit2)
```

```
##
## Call:
## lm(formula = medv ~ rm, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.346  -2.547   0.090   2.986  39.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -34.671      2.650  -13.08   <2e-16 ***
## rm             9.102      0.419   21.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.616 on 504 degrees of freedom
## Multiple R-squared:  0.4835, Adjusted R-squared:  0.4825
## F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
plot(medv~rm,data=Boston)
abline(fit2,col="blue")
```




```r
confint(fit2)
```

```
##                  2.5 %     97.5 %
## (Intercept) -39.876641 -29.464601
## rm            8.278855   9.925363
```

```r
par(mfrow=c(1,1))
```

```r
# Viendo que nuestro primer modelo tiene un mejor ajuste, nos centraremos en el modelo 'fit1'
# Por ello, vamos a calcular el error cuadrático medio (RMSE)
sqrt(sum(fit1$residuals^2)/length(fit1$residuals))
```

```
## [1] 6.203464
```

```r
predict(fit1,data.frame(lstat=c(5,10,15)))
```

```
##        1        2        3
## 29.80359 25.05335 20.30310
```

```r
# Ahora vamos a probar a añadir más variables a nuestro modelo lineal.
fit3=lm(medv~lstat+age,data=Boston)
summary(fit3)
```

```
##
## Call:
## lm(formula = medv ~ lstat + age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.981  -3.978  -1.283   1.968  23.158
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
## lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
## age          0.03454    0.01223   2.826  0.00491 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 6.173 on 503 degrees of freedom
## Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495 
## F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

Como se puede