What makes a house valuable?

A reproducible analysis for the Boston housing data

Outstanding student 1, Awesome student 2 and Great student 3

31/11/16

This short report shows a simple and non-exhaustive analysis for the price of the houses in the `Boston` dataset. The purpose is to quantify, by means of a multiple linear model, the effect of 14 variables in the price of a house in the suburbs of Boston.

We start by importing the data into `R` and considering a multiple linear regression of `medv` (median house value) in the rest of variables:

```
> # Import data
> library(MASS)
> data(Boston)
```

```
> mod <- lm(medv ~ ., data = Boston)
> summary(mod)
```

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

Residuals:
    Min      1Q  Median      3Q     Max
-15.595  -2.730  -0.518   1.777  26.199

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
zn           4.642e-02  1.373e-02   3.382 0.000778 ***
indus        2.056e-02  6.150e-02   0.334 0.738288
chas         2.687e+00  8.616e-01   3.118 0.001925 **
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
age          6.922e-04  1.321e-02   0.052 0.958229
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
black        9.312e-03  2.686e-03   3.467 0.000573 ***
lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared:  0.7406,    Adjusted R-squared:  0.7338
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

The variables `indus` and `age` are non-significant in this model. Also, although the adjusted R-squared is high, there seems to be a*clear non-linearity:

```
> plot(mod, 1)
```

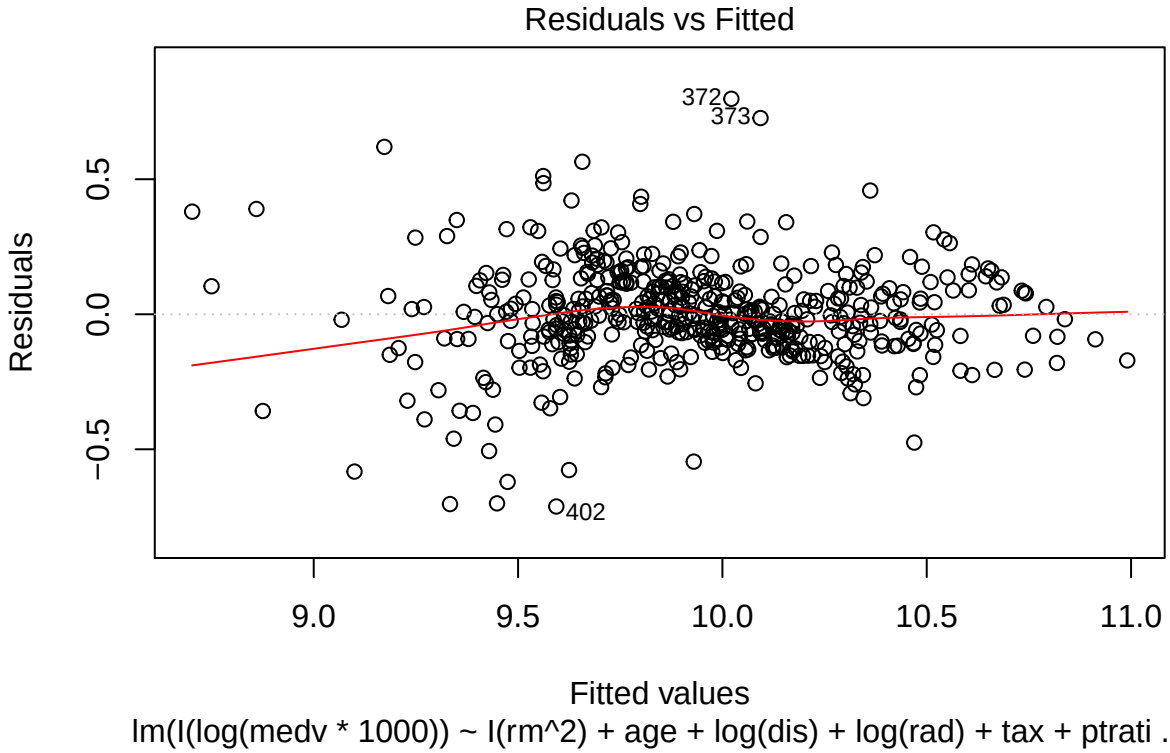
In order to bypass the non-linearity, we are going to consider the non-linear transformations given in Harrison and Rubinfeld (1978) for both the response and the predictors:

```
> modTransf <- lm(I(log(medv * 1000)) ~ I(rm^2) + age + log(dis) +
+                   log(rad) + tax + ptratio + I(black / 1000) +
+                   I(log(lstat / 100)) + crim + zn + indus + chas +
+                   I((10*nox)^2), data = Boston)
> summary(modTransf)


Call:
lm(formula = I(log(medv * 1000)) ~ I(rm^2) + age + log(dis) +
    log(rad) + tax + ptratio + I(black/1000) + I(log(lstat/100)) +
    crim + zn + indus + chas + I((10 * nox)^2), data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-0.71176 -0.09169 -0.00566  0.09895  0.79780

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       9.756e+00  1.496e-01  65.221  < 2e-16 ***
I(rm^2)           6.328e-03  1.312e-03   4.823 1.89e-06 ***
age               9.074e-05  5.263e-04   0.172 0.863179
log(dis)         -1.913e-01  3.339e-02  -5.727 1.78e-08 ***
log(rad)          9.571e-02  1.913e-02   5.002 7.91e-07 ***
tax              -4.203e-04  1.227e-04  -3.426 0.000664 ***
ptratio          -3.112e-02  5.013e-03  -6.208 1.14e-09 ***
```

```
I(black/1000)      3.637e-01  1.031e-01   3.527 0.000460 ***
I(log(lstat/100)) -3.712e-01  2.501e-02 -14.841  < 2e-16 ***
crim              -1.186e-02  1.245e-03  -9.532  < 2e-16 ***
zn                 8.016e-05  5.056e-04   0.159 0.874105
indus              2.395e-04  2.364e-03   0.101 0.919318
chas               9.140e-02  3.320e-02   2.753 0.006129 **
I((10 * nox)^2)   -6.380e-03  1.131e-03  -5.639 2.88e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1825 on 492 degrees of freedom
Multiple R-squared:  0.8059,    Adjusted R-squared:  0.8008
F-statistic: 157.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

The adjusted R-squared is now higher and, what is more important, the non-linearity now is more subtle (it is still not linear but closer than before):

```
> plot(modTransf, 1)
```



However, `modTransf` has more non-significant variables. Let's see if we can improve over the previous model by removing some of the non-significant variables. To that aim, we look for the best model in terms of the Bayesian Information Criterion (BIC) by `stepwise`:

```
> modTransfBIC <- stepwise(modTransf, trace = 0)
```

```
Direction:  backward/forward
Criterion:  BIC
```

```
> summary(modTransfBIC)
```

```
Call:
```

```
lm(formula = I(log(medv * 1000)) ~ I(rm^2) + log(dis) + log(rad) +
    tax + ptratio + I(black/1000) + I(log(lstat/100)) + crim +
    chas + I((10 * nox)^2), data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-0.71182 -0.09288 -0.00590  0.09763  0.79880

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        9.7677775  0.1386224  70.463  < 2e-16 ***
I(rm^2)            0.0063831  0.0012498   5.107 4.67e-07 ***
log(dis)          -0.1929697  0.0262514  -7.351 8.20e-13 ***
log(rad)           0.0947128  0.0181870   5.208 2.81e-07 ***
tax               -0.0004115  0.0001062  -3.874 0.000122 ***
ptratio           -0.0312259  0.0046959  -6.650 7.79e-11 ***
I(black/1000)      0.3643185  0.1025799   3.552 0.000420 ***
I(log(lstat/100)) -0.3696816  0.0225919 -16.363  < 2e-16 ***
crim              -0.0118642  0.0012204  -9.722  < 2e-16 ***
chas               0.0920105  0.0328785   2.799 0.005334 **
I((10 * nox)^2)   -0.0063382  0.0010951  -5.788 1.27e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1819 on 495 degrees of freedom
Multiple R-squared:  0.8059,    Adjusted R-squared:  0.8019
F-statistic: 205.5 on 10 and 495 DF,  p-value: < 2.2e-16
```

The resulting model has a slightly higher adjusted R-squared than `modTransf` with all the variables significant.

We explore the most significant variables to see if the model can be reduced drastically in complexity.

```
> mod3D <- lm(I(log(medv * 1000)) ~ I(log(lstat / 100)) + crim, data = Boston)
> summary(mod3D)


Call:
lm(formula = I(log(medv * 1000)) ~ I(log(lstat/100)) + crim,
    data = Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-0.75050 -0.13714 -0.01254  0.12003  0.88388

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        8.876988   0.041663 213.065   <2e-16 ***
I(log(lstat/100)) -0.495249   0.017291 -28.641   <2e-16 ***
crim              -0.011404   0.001208  -9.441   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2145 on 503 degrees of freedom
Multiple R-squared:  0.7258,    Adjusted R-squared:  0.7248
F-statistic: 665.9 on 2 and 503 DF,  p-value: < 2.2e-16
```

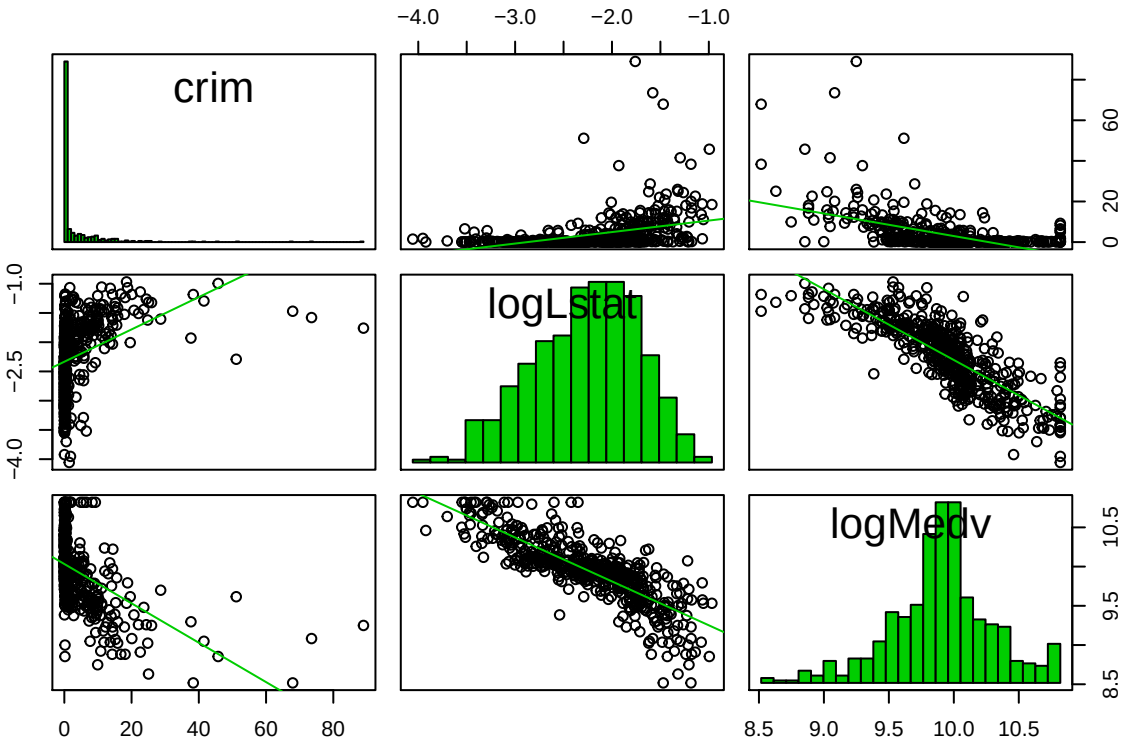
It turns out that **with only 2 variables, we explain the 72% of variability**. Compared with the 80% with 10 variables, it is an important improvement in terms of simplicity: the logarithm of `lstat` (percent of lower status of the population) and `crim` (crime rate) alone explain the 72% of the variability in the house prices. We add these variables to the dataset, so we can call `scatterplotMatrix` and `scatter3d` through `R Commander`,

```
> Boston$logMedv <- log(Boston$medv * 1000)
> Boston$logLstat <- log(Boston$lstat / 100)
```

and conclude with the visualization of:

1. the pair-by-pair relations of the response and the two predictors;
2. the full relation between the response and the two predictors.

```
> # 1
> scatterplotMatrix(~ crim + logLstat + logMedv, reg.line = lm, smooth = FALSE,
+                   spread = FALSE, span = 0.5, ellipse = FALSE,
+                   levels = c(.5, .9), id.n = 0, diagonal = 'histogram',
+                   data = Boston)
```




```
> # 2
> scatter3d(logMedv ~ crim + logLstat, data = Boston, fit = "linear",
+           residuals = TRUE, bg = "white", axis.scales = TRUE, grid = TRUE,
+           ellipsoid = FALSE)
```

You must enable Javascript to view this page properly.