

Regresión Lineal R - Laboratorio 1

Juanjo Sierra

28 de noviembre de 2018

Regresión Lineal con R - Laboratorio 1

En primer lugar hay que leer el dataset que se va a utilizar, ubicado en la carpeta ‘Datos’: `california.dat`. El dataset está en formato KEEL.

```
california = read.csv("../Datos/california.dat", header = FALSE, comment.char = "@")  
  
##      V1     V2 V3   V4   V5   V6   V7     V8     V9  
## 1 -117.03 32.78 17 5481 1618 2957 1537 2.5707 171300  
## 2 -118.23 33.80 26 239  135  165 112 1.3333 187500  
## 3 -122.46 37.71 39 2076 482 1738 445 3.1958 232100  
## 4 -122.06 37.94 19 4005 972 1896 893 2.5268 235700  
## 5 -122.87 38.68 32 4073 718 2053 629 3.7352 228000  
## 6 -122.47 37.66 18 4172 806 3226 790 5.7535 297900
```

Se asignan los nombres de las variables adecuadamente.

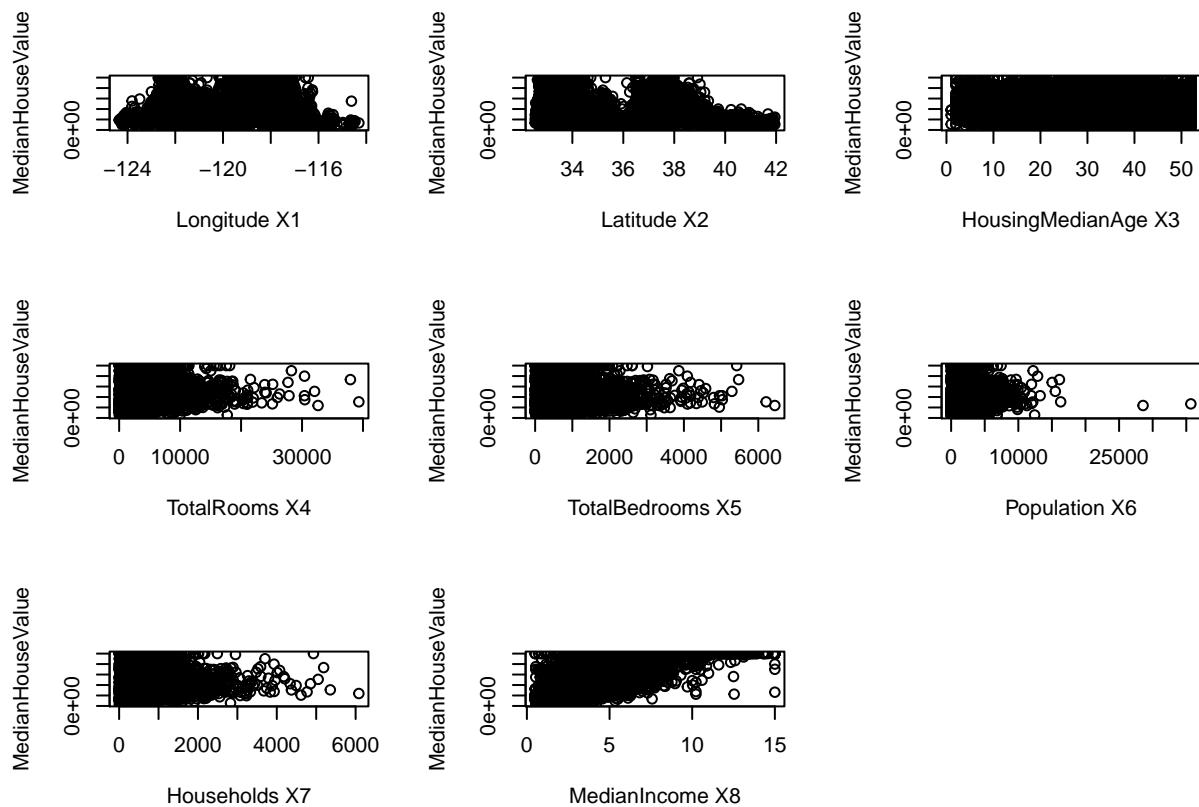
```
names(california) = c("Longitude", "Latitude", "HousingMedianAge",  
"TotalRooms", "TotalBedrooms", "Population", "Households",  
"MedianIncome", "MedianHouseValue")
```

```
head(california)
```

```
##    Longitude Latitude HousingMedianAge TotalRooms TotalBedrooms Population  
## 1    -117.03     32.78              17      5481       1618     2957  
## 2    -118.23     33.80              26      239        135      165  
## 3    -122.46     37.71              39     2076       482     1738  
## 4    -122.06     37.94              19     4005       972     1896  
## 5    -122.87     38.68              32     4073       718     2053  
## 6    -122.47     37.66              18     4172       806     3226  
##    Households MedianIncome MedianHouseValue  
## 1         1537     2.5707          171300  
## 2          112     1.3333          187500  
## 3          445     3.1958          232100  
## 4          893     2.5268          235700  
## 5          629     3.7352          228000  
## 6          790     5.7535          297900
```

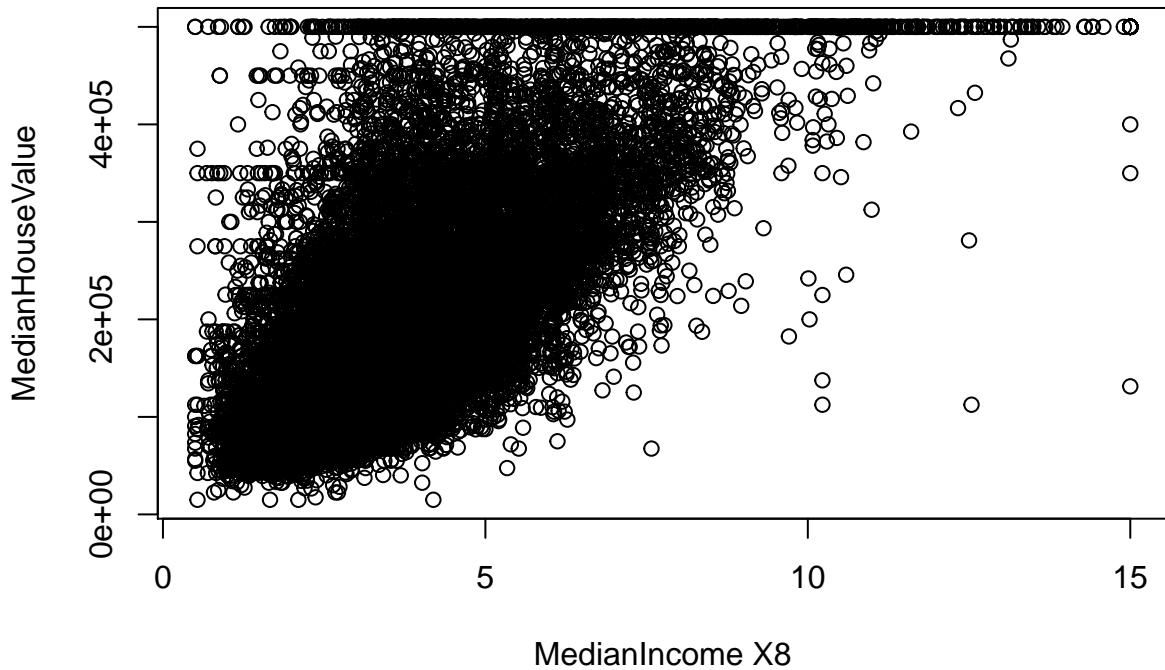
Vamos a visualizar todas las variables entre sí y respecto a la salida.

```
temp = california  
plotY <- function (x,y) {  
  plot(temp[,y]~temp[,x], xlab=paste(names(temp)[x], " X", x, sep=""),  
  ylab=names(temp)[y])  
}  
par(mfrow=c(3,3)) # Fijar ventana para gráficas  
x = sapply(1:(dim(temp)[2]-1), plotY, dim(temp)[2])  
par(mfrow=c(1,1)) # Cambiar el tipo de ventana a 1,1 otra vez
```



La única variable que podría resultar interesante parece ser MedianIncome. Podemos ampliar el gráfico para verla con más claridad.

```
plotY(8, dim(temp)[2])
```



Modelos lineales múltiples

Podemos probar a realizar un modelo lineal utilizando únicamente esta variable. Para ello vamos a utilizar la función `lm`.

```
fit1 = lm(MedianHouseValue~MedianIncome, data=california)
summary(fit1)

##
## Call:
## lm(formula = MedianHouseValue ~ MedianIncome, data = califor...
##
## Residuals:
##     Min      1Q  Median      3Q      Max
## -540697 -55950 -16979   36978  434023
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 45085.6    1322.9   34.08 <2e-16 ***
## MedianIncome 41793.8     306.8   136.22 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83740 on 20638 degrees of freedom
## Multiple R-squared:  0.4734, Adjusted R-squared:  0.4734
## F-statistic: 1.856e+04 on 1 and 20638 DF,  p-value: < 2.2e-16
```

El p-value del F-statistic nos sirve para calcular la confianza con la que afirmar que las variables mantienen una relación lineal ($1 - \text{p-value} * 100 = \% \text{ confianza}$). Ya que el p-value del F-statistic es muy pequeño la confianza es alta, con un modelo realizado con la variable `MedianIncome` el error de 0.4734 (nos fijamos en el Adjusted R-squared) es casi asegurado.

Podemos también probar a añadir una nueva variable al modelo, generando así un **modelo lineal múltiple**. Como ninguna variable parece especialmente prometedora, vamos a probar con `TotalBedrooms`, que al menos semánticamente parece relevante.

```
fit2 = lm(MedianHouseValue~MedianIncome + TotalBedrooms, data=california)
summary(fit2)
```

```
##
## Call:
## lm(formula = MedianHouseValue ~ MedianIncome + TotalBedrooms,
##      data = califor...
##
## Residuals:
##     Min      1Q  Median      3Q      Max
## -533221 -55954 -16538   36370  441499
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36702.58    1518.09   24.18 <2e-16 ***
## MedianIncome 41821.46     305.90   136.72 <2e-16 ***
## TotalBedrooms 15.39      1.38   11.15 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```

## Residual standard error: 83490 on 20637 degrees of freedom
## Multiple R-squared:  0.4766, Adjusted R-squared:  0.4766
## F-statistic:  9396 on 2 and 20637 DF,  p-value: < 2.2e-16

```

La variable TotalBedrooms de por sí no aporta mucho al modelo, ya que añadiéndola simplemente aumenta un 0.3% el R-squared, y sin embargo estamos aumentando su complejidad. Probemos a añadir una tercera variable, TotalRooms, que sigue una distribución aparentemente similar.

```

fit3 = lm(MedianHouseValue~MedianIncome + TotalBedrooms + TotalRooms, data=california)
summary(fit3)

```

```

##
## Call:
## lm(formula = MedianHouseValue ~ MedianIncome + TotalBedrooms +
##     TotalRooms, data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -613772  -52207  -15948   34782  463576
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9678.8791 1646.3690  5.879 4.19e-09 ***
## MedianIncome 49128.4737  357.5063 137.420 < 2e-16 ***
## TotalBedrooms 164.7084    4.2966  38.335 < 2e-16 ***
## TotalRooms    -30.9510    0.8464 -36.569 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80910 on 20636 degrees of freedom
## Multiple R-squared:  0.5085, Adjusted R-squared:  0.5084
## F-statistic:  7115 on 3 and 20636 DF,  p-value: < 2.2e-16

```

Como el R-squared ha aumentado y el p-value sigue siendo lo suficientemente pequeño como para garantizar una alta confianza, podemos seguir aumentando variables para el modelo. Vamos a probar Households que también parece seguir una distribución parecida.

```

fit4 = lm(MedianHouseValue~MedianIncome + TotalBedrooms + TotalRooms + Households, data=california)
summary(fit4)

```

```

##
## Call:
## lm(formula = MedianHouseValue ~ MedianIncome + TotalBedrooms +
##     TotalRooms + Households, data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -612232  -52296  -15873   34765  464450
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9408.4542 1647.4486  5.711 1.14e-08 ***
## MedianIncome 49049.0816  358.0272 136.998 < 2e-16 ***
## TotalBedrooms 140.8491    7.7029 18.285 < 2e-16 ***
## TotalRooms    -31.1095    0.8472 -36.722 < 2e-16 ***
## Households    27.6838    7.4192  3.731 0.000191 ***

```

```

## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80880 on 20635 degrees of freedom
## Multiple R-squared: 0.5088, Adjusted R-squared: 0.5087
## F-statistic: 5343 on 4 and 20635 DF, p-value: < 2.2e-16

```

Llegados a este punto, esta variable aumenta complejidad pero el modelo no mejora prácticamente nada, por lo que no merecería la pena añadirla. El modelo que utiliza MedianIncome, TotalBedrooms y TotalRooms es el mejor dentro de los que tienen poca complejidad que hemos podido encontrar. Vamos a hacerlo ahora a la inversa: comenzamos con el modelo que tiene todas las variables y vamos eliminando las menos prometedoras.

```

fit5 = lm(MedianHouseValue~., data=california)
summary(fit5)

```

```

##
## Call:
## lm(formula = MedianHouseValue ~ ., data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -563013  -43592  -11327   30307  803996
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.594e+06 6.254e+04 -57.468 < 2e-16 ***
## Longitude    -4.282e+04 7.130e+02 -60.061 < 2e-16 ***
## Latitude     -4.258e+04 6.733e+02 -63.240 < 2e-16 ***
## HousingMedianAge 1.156e+03 4.317e+01 26.787 < 2e-16 ***
## TotalRooms   -8.182e+00 7.881e-01 -10.381 < 2e-16 ***
## TotalBedrooms 1.134e+02 6.902e+00 16.432 < 2e-16 ***
## Population   -3.854e+01 1.079e+00 -35.716 < 2e-16 ***
## Households    4.831e+01 7.515e+00  6.429 1.32e-10 ***
## MedianIncome  4.025e+04 3.351e+02 120.123 < 2e-16 ***
##
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69530 on 20631 degrees of freedom
## Multiple R-squared: 0.6371, Adjusted R-squared: 0.637
## F-statistic: 4528 on 8 and 20631 DF, p-value: < 2.2e-16

```

Se observa un incremento considerable en el valor del R-squared del modelo, aunque también se ha incrementado su complejidad al añadir todas las variables. Se puede eliminar aquella que tiene el p-value más alto (es la menos prometedora) para ver si puede mejorar aún más.

```

fit6 = lm(MedianHouseValue~.-Households, data=california)
summary(fit6)

```

```

##
## Call:
## lm(formula = MedianHouseValue ~ . - Households, data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -566279  -43499  -11344   30482  744792
##
```

```

## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.673e+06 6.139e+04 -59.83 <2e-16 ***
## Longitude   -4.368e+04 7.010e+02 -62.31 <2e-16 ***
## Latitude    -4.326e+04 6.654e+02 -65.02 <2e-16 ***
## HousingMedianAge 1.165e+03 4.319e+01 26.99 <2e-16 ***
## TotalRooms   -8.439e+00 7.879e-01 -10.71 <2e-16 ***
## TotalBedrooms 1.497e+02 3.967e+00 37.74 <2e-16 ***
## Population   -3.514e+01 9.416e-01 -37.32 <2e-16 ***
## MedianIncome  4.042e+04 3.343e+02 120.91 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69600 on 20632 degrees of freedom
## Multiple R-squared:  0.6364, Adjusted R-squared:  0.6363
## F-statistic:  5158 on 7 and 20632 DF, p-value: < 2.2e-16

```

El acierto del modelo disminuye muy poco (menos de un 1%) y le estamos restando complejidad, así que podemos seguir probando a eliminar variables. Como todas tienen un p-value similar ahora mismo, nos guiaremos por las gráficas, y eliminamos la variable HousingMedianAge que no parece nada lineal.

```
fit7 = lm(MedianHouseValue ~ . - Households - HousingMedianAge, data=california)
summary(fit7)
```

```

##
## Call:
## lm(formula = MedianHouseValue ~ . - Households - HousingMedianAge,
##      data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -535455 -43549 -11901   30095  792056
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.109e+06 6.026e+04 -68.19 <2e-16 ***
## Longitude   -4.915e+04 6.829e+02 -71.97 <2e-16 ***
## Latitude    -4.813e+04 6.516e+02 -73.87 <2e-16 ***
## TotalRooms   -9.774e+00 8.001e-01 -12.22 <2e-16 ***
## TotalBedrooms 1.466e+02 4.035e+00 36.33 <2e-16 ***
## Population   -3.577e+01 9.577e-01 -37.35 <2e-16 ***
## MedianIncome  3.928e+04 3.374e+02 116.41 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 70810 on 20633 degrees of freedom
## Multiple R-squared:  0.6235, Adjusted R-squared:  0.6234
## F-statistic:  5696 on 6 and 20633 DF, p-value: < 2.2e-16

```

Ha descendido un 1% el R-squared pero para eliminar complejidad es un justo precio a pagar. Podemos seguir así eliminando variables, probando ahora con Population.

```
fit8 = lm(MedianHouseValue ~ . - Households - HousingMedianAge - Population, data=california)
summary(fit8)
```

```
##
```

```

## Call:
## lm(formula = MedianHouseValue ~ . - Households - HousingMedianAge -
##     Population, data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -553439  -46021  -14613   30331  504198
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.945e+06 6.209e+04 -63.53 <2e-16 ***
## Longitude    -4.660e+04 7.020e+02 -66.38 <2e-16 ***
## Latitude     -4.457e+04 6.660e+02 -66.92 <2e-16 ***
## TotalRooms   -1.857e+01 7.900e-01 -23.51 <2e-16 ***
## TotalBedrooms 1.050e+02 4.007e+00  26.20 <2e-16 ***
## MedianIncome  4.146e+04 3.433e+02 120.76 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73170 on 20634 degrees of freedom
## Multiple R-squared:  0.5981, Adjusted R-squared:  0.598
## F-statistic:  6141 on 5 and 20634 DF,  p-value: < 2.2e-16

```

Eliminando esta variable el R-squared desciende por debajo de un 0.6, y aquí consideramos que es mejor mantener el modelo anterior y primar en este caso la precisión frente a la complejidad.

Interacciones

Vamos a realizar algunas interacciones entre las variables más prometedoras que hemos encontrado realizando los modelos lineales. MedianIncome parecía ser la más interesante, realizaremos el producto entre ella y TotalBedrooms.

```

fit9 = lm(MedianHouseValue~MedianIncome*TotalBedrooms, data=california)
summary(fit9)

```

```

##
## Call:
## lm(formula = MedianHouseValue ~ MedianIncome * TotalBedrooms,
##     data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -533987  -55922  -16523   36366  441738
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36393.393   2142.425 16.987 < 2e-16 ***
## MedianIncome        41896.940    479.347 87.404 < 2e-16 ***
## TotalBedrooms         16.043      3.495  4.591 4.44e-06 ***
## MedianIncome:TotalBedrooms -0.162      0.792 -0.205   0.838
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83490 on 20636 degrees of freedom

```

```

## Multiple R-squared:  0.4766, Adjusted R-squared:  0.4765
## F-statistic:  6264 on 3 and 20636 DF,  p-value: < 2.2e-16

```

Vemos que el p-value de TotalBedrooms crece un poco y que el de la variable resultante es demasiado alto. Además el R-squared obtenido no es nada reseñable. Cambiamos y probamos la variable HousingMedianAge.

```

fit10 = lm(MedianHouseValue~MedianIncome*HousingMedianAge, data=california)
summary(fit10)

```

```

##
## Call:
## lm(formula = MedianHouseValue ~ MedianIncome * HousingMedianAge,
##      data = california)
##
## Residuals:
##    Min     1Q   Median     3Q    Max
## -625683 -53564 -15167  36644 442699
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3190.48    3259.72   0.979   0.328
## MedianIncome               39894.56    711.28  56.089  < 2e-16 ***
## HousingMedianAge           1293.55    99.61  12.987  < 2e-16 ***
## MedianIncome:HousingMedianAge 112.95    22.27   5.071 3.99e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80810 on 20636 degrees of freedom
## Multiple R-squared:  0.5097, Adjusted R-squared:  0.5097
## F-statistic:  7152 on 3 and 20636 DF,  p-value: < 2.2e-16

```

Esta variable mejora el modelo cuando se realiza el producto, por lo que es mejor que el modelo simple que se ha tomado como base. Comprobemos qué pasa si ahora añadimos la variable TotalBedrooms.

```

fit11 = lm(MedianHouseValue~MedianIncome*HousingMedianAge*TotalBedrooms, data=california)
summary(fit11)

```

```

##
## Call:
## lm(formula = MedianHouseValue ~ MedianIncome * HousingMedianAge *
##      TotalBedrooms, data = california)
##
## Residuals:
##    Min     1Q   Median     3Q    Max
## -512034 -51201 -14276  35190 454226
##
## Coefficients:
##                               Estimate Std. Error t value
## (Intercept)                7.506e+03  4.791e+03   1.567
## MedianIncome               3.799e+04  1.000e+03  37.970
## HousingMedianAge          6.208e+02  1.501e+02   4.136
## TotalBedrooms             -1.882e+01  6.635e+00  -2.836
## MedianIncome:HousingMedianAge 2.691e+01  3.219e+01   0.836
## MedianIncome:TotalBedrooms  2.016e-01  1.401e+00   0.144
## HousingMedianAge:TotalBedrooms 1.386e+00  2.660e-01   5.210
## MedianIncome:HousingMedianAge:TotalBedrooms 4.383e-01  6.137e-02   7.142

```

```

##                                     Pr(>|t|)
## (Intercept)                   0.11724
## MedianIncome                  < 2e-16 ***
## HousingMedianAge              3.55e-05 ***
## TotalBedrooms                 0.00458 **
## MedianIncome:HousingMedianAge 0.40326
## MedianIncome:TotalBedrooms    0.88556
## HousingMedianAge:TotalBedrooms 1.90e-07 ***
## MedianIncome:HousingMedianAge:TotalBedrooms 9.52e-13 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78030 on 20632 degrees of freedom
## Multiple R-squared: 0.5429, Adjusted R-squared: 0.5428
## F-statistic: 3501 on 7 and 20632 DF, p-value: < 2.2e-16

```

Ahora el resultado es mejor que al añadir solamente TotalBedrooms y MedianIncome. Y es que el producto de ambas tiene un p-value muy alto, pero la variable resultante de multiplicar las tres da un p-value bajo y el R-squared del modelo es sustancialmente mejor, por lo que aceptamos esta combinación como un mejor modelo.

No linealidad

A continuación vamos a probar la no linealidad de alguna variable en un modelo simple. Podemos probar con MedianIncome, que parece intuir una curva cuadrática.

```
fit12 = lm(MedianHouseValue~MedianIncome+I(MedianIncome^2), data=california)
summary(fit12)
```

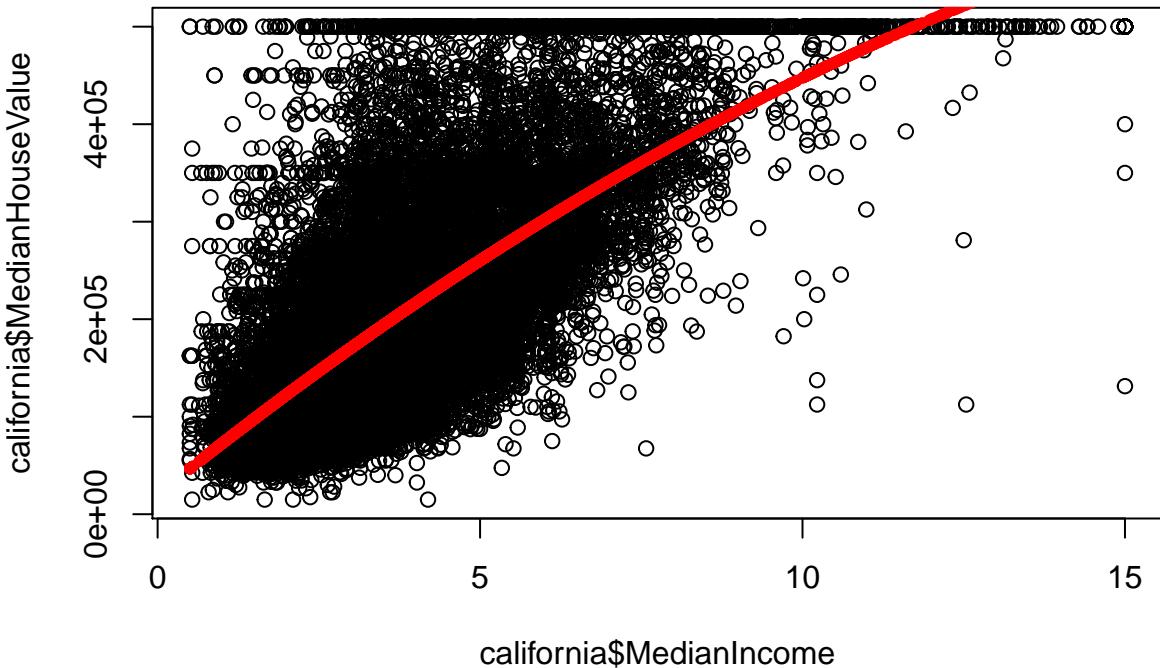
```

##
## Call:
## lm(formula = MedianHouseValue ~ MedianIncome + I(MedianIncome^2),
##      data = california)
##
## Residuals:
##   Min     1Q   Median     3Q    Max
## -450948 -55535 -16549  37400  453242
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20421.11    2257.64   9.045  <2e-16 ***
## MedianIncome 53211.27     902.09  58.987  <2e-16 ***
## I(MedianIncome^2) -1050.42      78.09 -13.451  <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83370 on 20637 degrees of freedom
## Multiple R-squared: 0.478, Adjusted R-squared: 0.478
## F-statistic: 9450 on 2 and 20637 DF, p-value: < 2.2e-16

```

El ajuste del modelo no muestra un gran cambio. Podemos comprobar cómo se ajusta el modelo a los valores de la variable.

```
plot(california$MedianHouseValue~california$MedianIncome)
points(california$MedianIncome,fitted(fit12),col="red",pch=20)
```



Podemos probar esta no linealidad de la variable MedianIncome dentro del mejor modelo que hemos encontrado hasta ahora, el que contenía las variables MedianIncome, Latitude, Longitude, TotalRooms, TotalBedrooms y Population.

```
fit13 = lm(MedianHouseValue ~ . + I(MedianIncome^2) - HousingMedianAge - Households,
            data=california)
summary(fit13)
```

```
##
## Call:
## lm(formula = MedianHouseValue ~ . + I(MedianIncome^2) - HousingMedianAge -
##     Households, data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -484224 -43889 -11564  30361  785227 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -4.042e+06  6.051e+04 -66.809 <2e-16 ***
## Longitude    -4.814e+04  6.892e+02 -69.840 <2e-16 ***
## Latitude     -4.709e+04  6.590e+02 -71.450 <2e-16 ***
## TotalRooms   -1.158e+01  8.196e-01 -14.134 <2e-16 ***
## TotalBedrooms 1.529e+02  4.077e+00  37.498 <2e-16 ***
## Population   -3.510e+01  9.580e-01 -36.636 <2e-16 ***
## MedianIncome  4.706e+04  8.672e+02  54.265 <2e-16 ***
## I(MedianIncome^2) -6.673e+02  6.851e+01 -9.739 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 70650 on 20632 degrees of freedom
## Multiple R-squared:  0.6253, Adjusted R-squared:  0.6251 
## F-statistic: 4918 on 7 and 20632 DF, p-value: < 2.2e-16
```

El nuevo modelo ha mejorado al anterior, pero es más complejo y la mejora es poco sustancial (menos de un 1%), por lo que no merece la pena. El mejor modelo encontrado, y que tiene un buen balance complejidad/acierto es el que tiene todas las variables menos Households y HousingMedianAge.