

# Evaluación Continua: Regresión

*Juan Ramón Gómez Berzosa*

26/11/2018

A continuación importaré las librerías que vamos a utilizar a lo largo del trabajo.

```
library(ISLR)
library(MASS)
require(kknn)
```

```
## Loading required package: kknn
```

El conjunto que vamos a utilizar es el de California, a continuación procederemos a importarlo.

```
california <- read.csv("california.dat", comment.char="@")
dim(california)
```

```
## [1] 20639      9
names(california) <- c("Longitude", "Latitude", "HousingMedianAge",
"TotalRooms", "TotalBedrooms", "Population", "Households", "MedianIncome", "MedianHouseValue")
```

California es un dataset de 9 variables y 20639 entradas referente a las casas de California, sus variables son las siguientes:

```
colnames(california)
```

```
## [1] "Longitude"          "Latitude"           "HousingMedianAge"
## [4] "TotalRooms"          "TotalBedrooms"       "Population"
## [7] "Households"          "MedianIncome"        "MedianHouseValue"
```

Sus atributos son los siguientes (información obtenida de: California Housing Dataset)

1. Longitude: Medida de como de lejos al oeste está esta casa, un valor más alto indica que está más al oeste.
2. Latitude: Medida de como de lejos al norte está esta casa, un valor más alto indica que está más al norte.
3. HousingMedianAge: Edad media de una casa dentro de un bloque, cuanto más bajo es la edad más nueva será la casa.
4. TotalRooms: Número total de habitaciones dentro de un bloque.
5. TotalBedrooms: Número total de dormitorios dentro de un bloque.
6. Population: Número total de personas que residen en un bloque.
7. Households: Número total de hogares por bloque. Un grupo de personas viviendo en una casa constituyen un hogar.
8. MedianIncome: Ingresos medios por hogar dentro de un bloque de casas (decenas de miles de dólares).
9. MedianHouseValue: Valor medio de la vivienda por bloque (dólares).

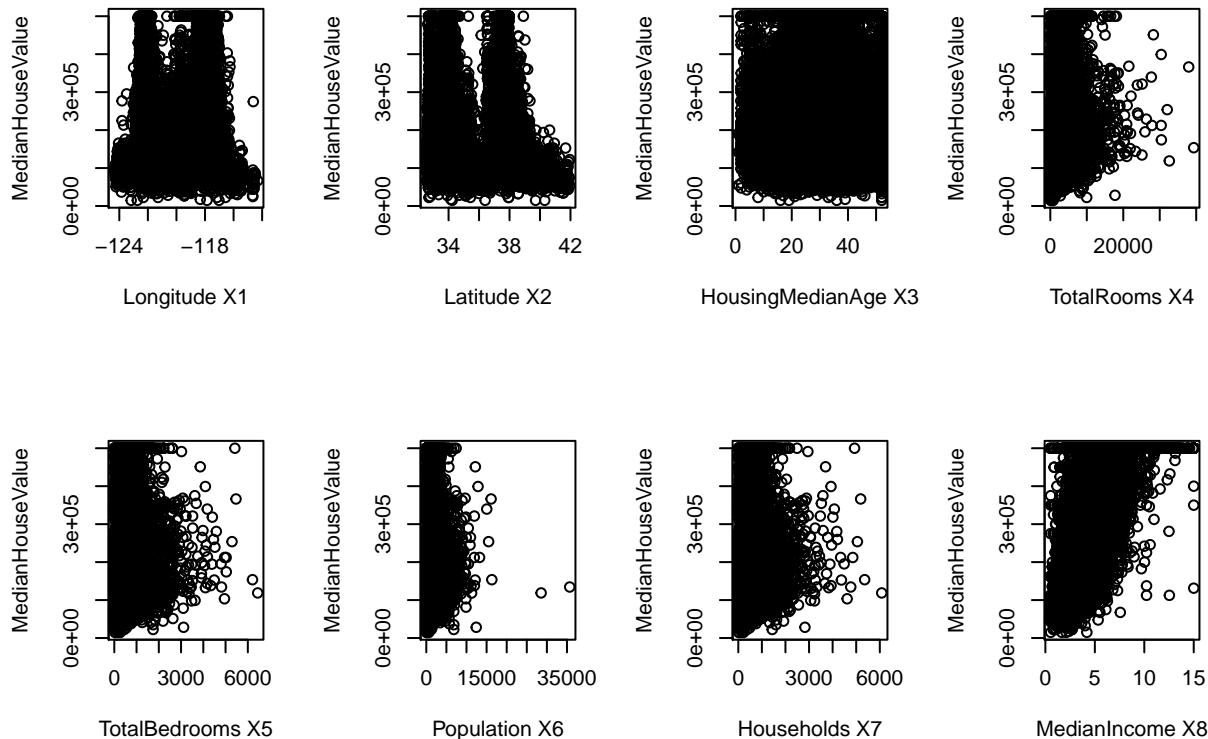
```
## Modelo Lineal Simple
```

A continuación vamos a añadir el objeto al entorno de trabajo para poder utilizarlo con más comodidad sin tener que estar continuamente haciendo accesos por el operador “\$” al dataset california.

```
attach(california)
```

Vamos a comparar las variables respecto a la de salida: "MedianHouseValue".

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



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

Como podemos observar, hay algunas variables que podemos ver que van a ser malas a priori como son "Population", "Total Bedrooms", "TotalRooms" y "House Holds" y podemos ver que la que mejor pinta tiene es "MedianIncome". Vamos a descartar las 4 que hemos dicho que apreciamos que son las peores y probaremos con el resto de variables, aunque seguramente la que mejor modelo nos de sea la variable "Median Income".

```
# Obtenemos el modelo para cada variable  
fit1=lm(MedianHouseValue~Longitude,data=california)  
fit1  
  
##  
## Call:  
## lm(formula = MedianHouseValue ~ Longitude, data = california)  
##  
## Coefficients:  
## (Intercept)    Longitude  
##           -109598            -2647
```

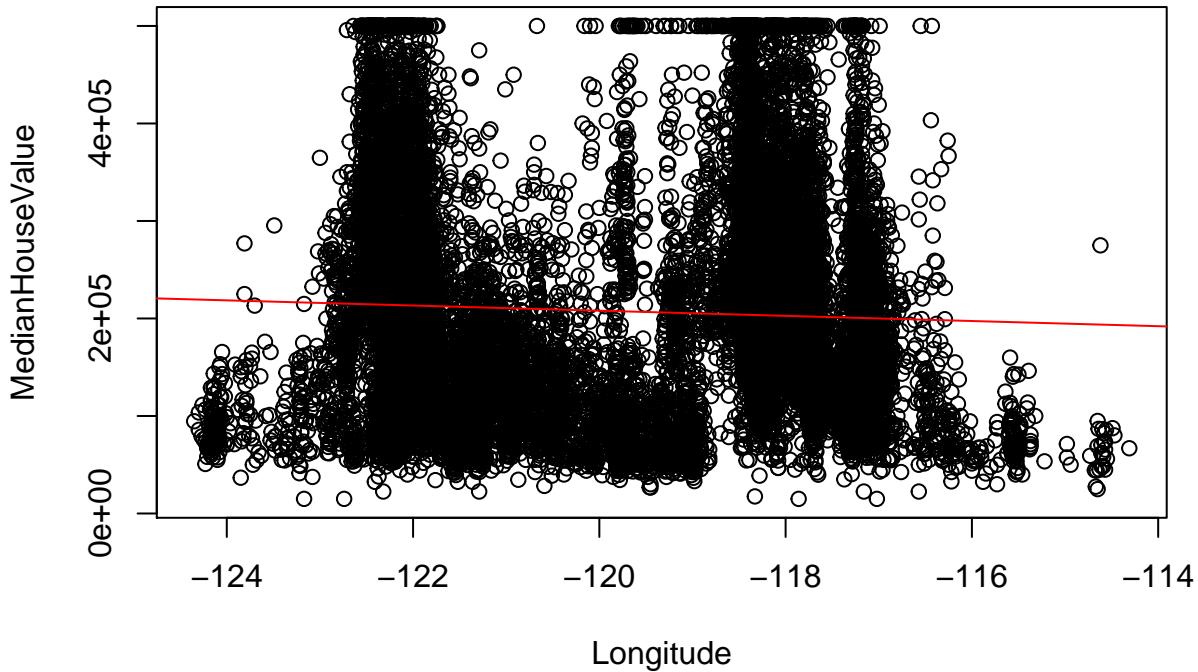
```

#Visualizamos los resultados
summary(fit1)

## 
## Call:
## lm(formula = MedianHouseValue ~ Longitude, data = california)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -201387 -86464 -26344  56599 301453 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -109597.6    47897.1  -2.288  0.0221 *  
## Longitude     -2646.6      400.5  -6.608 3.99e-11 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 115300 on 20637 degrees of freedom 
## Multiple R-squared:  0.002111, Adjusted R-squared:  0.002063 
## F-statistic: 43.66 on 1 and 20637 DF, p-value: 3.993e-11 

par(mfrow=c(1,1))
plot(MedianHouseValue~Longitude,california)
abline(fit1,col="red")

```



```

confint(fit1)

##           2.5 %    97.5 %
## (Intercept) -203479.739 -15715.506
## Longitude     -3431.669 -1861.558

# Obtenemos el modelo para cada variable
fit2=lm(MedianHouseValue~Latitude,data=california)

```

```

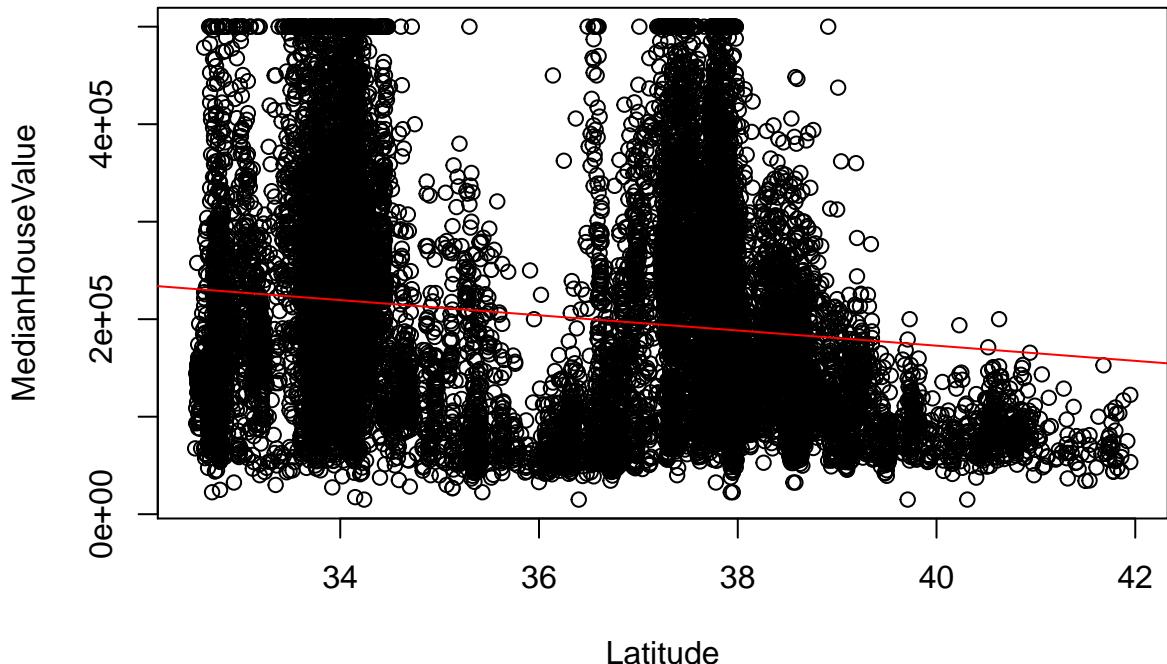
fit2

##
## Call:
## lm(formula = MedianHouseValue ~ Latitude, data = california)
##
## Coefficients:
## (Intercept)      Latitude
##        484433         -7790
#Visualizamos los resultados
summary(fit2)

##
## Call:
## lm(formula = MedianHouseValue ~ Latitude, data = california)
##
## Residuals:
##    Min     1Q   Median     3Q    Max 
## -207120 -84026 -30049  57108 318679
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 484433.0  13284.6   36.47 <2e-16 ***
## Latitude     -7790.1    372.2  -20.93 <2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 114200 on 20637 degrees of freedom
## Multiple R-squared:  0.02079,   Adjusted R-squared:  0.02074 
## F-statistic: 438.1 on 1 and 20637 DF,  p-value: < 2.2e-16

par(mfrow=c(1,1))
plot(MedianHouseValue~Latitude,california)
abline(fit2,col="red")

```



```

confint(fit2)

##              2.5 %    97.5 %
## (Intercept) 458394.116 510471.931
## Latitude     -8519.526  -7060.598
# Obtenemos el modelo para cada variable
fit3=lm(MedianHouseValue~HousingMedianAge,data=california)
fit3

##
## Call:
## lm(formula = MedianHouseValue ~ HousingMedianAge, data = california)
##
## Coefficients:
##             (Intercept)  HousingMedianAge
##                 179123.6                  968.4
#Visualizamos los resultados
summary(fit3)

##
## Call:
## lm(formula = MedianHouseValue ~ HousingMedianAge, data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -214479  -85039  -25833   58351  318941 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 179123.57    1985.54   90.21 <2e-16 ***
## HousingMedianAge 968.36      63.47   15.26 <2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

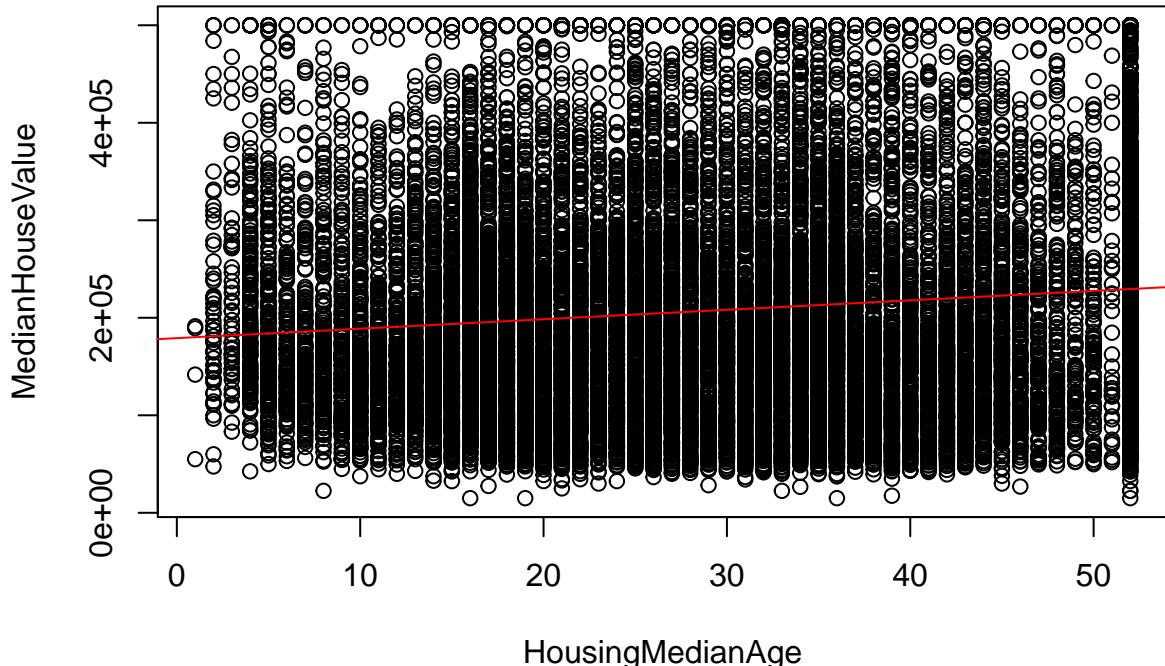
```

```

## 
## Residual standard error: 114800 on 20637 degrees of freedom
## Multiple R-squared:  0.01115,   Adjusted R-squared:  0.01111 
## F-statistic: 232.8 on 1 and 20637 DF,  p-value: < 2.2e-16

par(mfrow=c(1,1))
plot(MedianHouseValue~HousingMedianAge,california)
abline(fit3,col="red")

```



```

confint(fit3)

##                   2.5 %      97.5 %
## (Intercept) 175231.7612 183015.378
## HousingMedianAge     843.9577    1092.769

# Obtenemos el modelo para cada variable
fit4=lm(MedianHouseValue~MedianIncome,data=california)
fit4

```

```

## 
## Call:
## lm(formula = MedianHouseValue ~ MedianIncome, data = california)
## 
## Coefficients:
##   (Intercept) MedianIncome
##       45083        41794

#Visualizamos los resultados
summary(fit4)

```

```

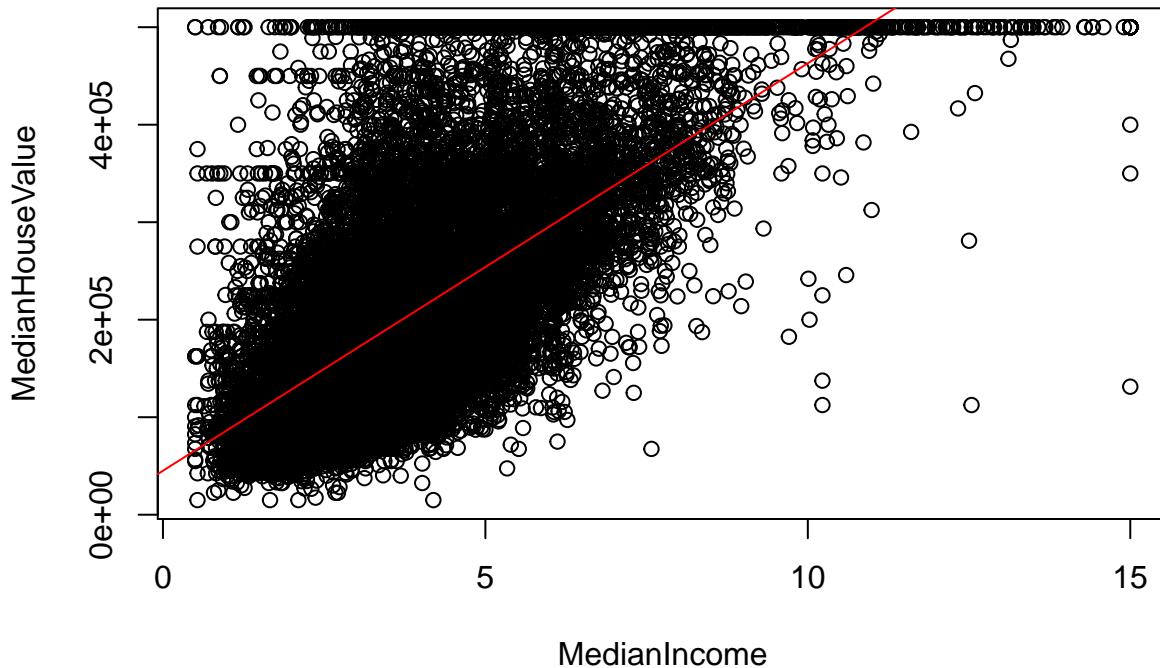
## 
## Call:
## lm(formula = MedianHouseValue ~ MedianIncome, data = california)
## 
## Residuals:

```

```

##      Min      1Q   Median      3Q     Max
## -540700 -55951 -16978  36979 434025
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 45083.4    1322.9   34.08 <2e-16 ***
## MedianIncome 41794.2     306.8  136.22 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83740 on 20637 degrees of freedom
## Multiple R-squared:  0.4734, Adjusted R-squared:  0.4734
## F-statistic: 1.856e+04 on 1 and 20637 DF, p-value: < 2.2e-16
par(mfrow=c(1,1))
plot(MedianHouseValue~MedianIncome,california)
abline(fit4,col="red")

```



```
confint(fit4)
```

```

##           2.5 % 97.5 %
## (Intercept) 42490.34 47676.46
## MedianIncome 41192.79 42395.56

```

Como hemos apreciado, la variable que nos da el mejor modelo es “Median Income” según el valor del residuo cuadrático (tanto normal como ajustado). Además, tiene un p-value bastante bajo, lo cual nos afirma que ambos términos son significativos y se pueden utilizar como predictores para nuestro modelo lineal.

La segunda mejor variable es “Latitude”, aunque como podemos apreciar ninguna variable del dataset es buena por sí sola para ajustar un buen modelo lineal a estos datos.

Ahora vamos a predecir los datos utilizando el modelo de regresión lineal que hemos aprendido usando la variable “Median Income” y vamos a calcular el RMSE (raíz de la suma del error cuadrático medio) con respecto al conjunto inicial. Usaremos el conjunto inicial a modo de test.

```

#predict(fit4,data.frame(MedianIncome=c(5,10,15)))
#yprime=predict(fit1,data.frame(lstat=Boston$lstat))
yprime=predict(fit4,california)
sqrt(sum(abs(MedianHouseValue-yprime)^2)/length(yprime))

## [1] 83735.5

```

## Regresión Lineal Múltiple

Para aplicar la regresión lineal múltiple voy a empezar desde arriba, es decir, partiendo de todas las variables e ir quitando poco a poco las que peor p-value tengan para intentar maximizar el valor del R-squared.

Como ya hemos graficado anteriormente todas la salida una a una contra todas las variables y hemos aplicado regresión lineal múltiple, ya sabemos cuáles son a priori las mejores variables. Empezaremos viendo el valor de r-squared con todas las variables e iremos quitando algunas a ver si podemos maximizarlo.

```

#con el punto indico que me haga la combinación lineal con todas las variables.
fit.l1=lm(MedianHouseValue~.,data=california)
summary(fit.l1)

```

```

##
## Call:
## lm(formula = MedianHouseValue ~ ., data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -563016  -43593  -11324   30320  804281
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -3.594e+06 6.254e+04 -57.462 < 2e-16 ***
## Longitude              -4.282e+04 7.130e+02 -60.056 < 2e-16 ***
## Latitude               -4.258e+04 6.733e+02 -63.239 < 2e-16 ***
## HousingMedianAge     1.156e+03 4.317e+01  26.782 < 2e-16 ***
## TotalRooms             -8.196e+00 7.882e-01 -10.397 < 2e-16 ***
## TotalBedrooms          1.134e+02 6.902e+00  16.434 < 2e-16 ***
## Population             -3.855e+01 1.079e+00 -35.728 < 2e-16 ***
## Households              4.843e+01 7.516e+00   6.444 1.19e-10 ***
## MedianIncome            4.025e+04 3.351e+02 120.127 < 2e-16 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69530 on 20630 degrees of freedom
## Multiple R-squared:  0.6371, Adjusted R-squared:  0.637
## F-statistic:  4528 on 8 and 20630 DF,  p-value: < 2.2e-16

```

Como podemos observar hemos obtenido un valor de R-squared más alto que cuando aplicamos regresión lineal simple con la variable MedianIncome. Ahora vamos a intentar quitar alguna variable, en este caso quitaremos en primer lugar la variable "Households" ya que es la que tiene el p-value más bajo:

```

#con el punto indico que me haga la combinación lineal con todas las variables.
fit.l2=lm(MedianHouseValue~.-Households,data=california)
summary(fit.l2)

```

```

##

```

```

## Call:
## lm(formula = MedianHouseValue ~ . - Households, data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5666289 -43500  -11341   30484  744913
##
## 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 26.98 <2e-16 ***
## TotalRooms    -8.453e+00 7.880e-01 -10.73 <2e-16 ***
## TotalBedrooms 1.498e+02 3.969e+00 37.76 <2e-16 ***
## Population    -3.515e+01 9.416e-01 -37.33 <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 20631 degrees of freedom
## Multiple R-squared:  0.6364, Adjusted R-squared:  0.6363
## F-statistic:  5158 on 7 and 20631 DF, p-value: < 2.2e-16

```

Como podemos observar el valor de R-squared es más bajo, como tenemos todas las demás con el mismo p-value intentaremos hacer diferentes combinaciones para ver si lo podemos mejorar, para ello nos basaremos en que variables tenían mejor aspecto en los gráficos:

```

#con el punto indico que me haga la combinación lineal con todas las variables.
fit.13=lm(MedianHouseValue~.-Households-Population,data=california)
summary(fit.13)

```

```

##
## Call:
## lm(formula = MedianHouseValue ~ . - Households - Population,
##      data = california)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -585007 -46276  -13637   31565  490581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.497e+06 6.324e+04 -55.30 <2e-16 ***
## Longitude    -4.099e+04 7.205e+02 -56.90 <2e-16 ***
## Latitude     -3.960e+04 6.799e+02 -58.24 <2e-16 ***
## HousingMedianAge 1.206e+03 4.461e+01 27.03 <2e-16 ***
## TotalRooms    -1.704e+01 7.787e-01 -21.88 <2e-16 ***
## TotalBedrooms 1.091e+02 3.942e+00 27.67 <2e-16 ***
## MedianIncome   4.261e+04 3.401e+02 125.28 <2e-16 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71910 on 20632 degrees of freedom
## Multiple R-squared:  0.6118, Adjusted R-squared:  0.6117

```

```

## F-statistic:  5420 on 6 and 20632 DF,  p-value: < 2.2e-16
#con el punto indico que me haga la combinación lineal con todas las variables.
fit.14=lm(MedianHouseValue~.-Households-Population-Longitude,data=california)
summary(fit.14)

##
## Call:
## lm(formula = MedianHouseValue ~ . - Households - Population -
##     Longitude, data = california)
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -643116 -49505 -14247  34477  468122
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 64996.3959   9623.2990   6.754 1.48e-11 ***
## Latitude    -3360.0829   255.8983  -13.131 < 2e-16 ***
## HousingMedianAge 1936.1645   45.9451   42.141 < 2e-16 ***
## TotalRooms     -24.8837    0.8243  -30.188 < 2e-16 ***
## TotalBedrooms   152.9074    4.1577   36.777 < 2e-16 ***
## MedianIncome   48951.7757   345.5511  141.663 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 77340 on 20633 degrees of freedom
## Multiple R-squared:  0.5509, Adjusted R-squared:  0.5508
## F-statistic:  5063 on 5 and 20633 DF,  p-value: < 2.2e-16
#con el punto indico que me haga la combinación lineal con todas las variables.
fit.15=lm(MedianHouseValue~.-Households-Longitude,data=california)
summary(fit.15)

##
## Call:
## lm(formula = MedianHouseValue ~ . - Households - Longitude, data = california)
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -630772 -47828 -12169  34352  604720
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.128e+05  9.587e+03   11.77  <2e-16 ***
## Latitude    -4.424e+03  2.538e+02  -17.43  <2e-16 ***
## HousingMedianAge 1.943e+03  4.507e+01   43.10  <2e-16 ***
## TotalRooms   -1.820e+01  8.419e-01  -21.61  <2e-16 ***
## TotalBedrooms  1.891e+02  4.271e+00   44.27  <2e-16 ***
## Population   -2.911e+01  1.021e+00  -28.52  <2e-16 ***
## MedianIncome   4.749e+04  3.428e+02  138.53  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75860 on 20632 degrees of freedom
## Multiple R-squared:  0.568, Adjusted R-squared:  0.5678

```

```

## F-statistic:  4521 on 6 and 20632 DF,  p-value: < 2.2e-16
#con el punto indico que me haga la combinación lineal con todas las variables.
fit.16=lm(MedianHouseValue~.-Population,data=california)
summary(fit.16)

##
## Call:
## lm(formula = MedianHouseValue ~ . - Population, data = california)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -587511 -45759 -13174  31325 473437 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -3.662e+06  6.441e+04  -56.84   <2e-16 ***
## Longitude    -4.292e+04  7.347e+02  -58.41   <2e-16 ***
## Latitude     -4.139e+04  6.929e+02  -59.73   <2e-16 ***
## HousingMedianAge 1.215e+03  4.445e+01   27.32   <2e-16 ***
## TotalRooms    -1.605e+01  7.800e-01  -20.58   <2e-16 ***
## TotalBedrooms 1.783e+02  6.861e+00   25.99   <2e-16 ***
## Households    -8.312e+01  6.751e+00  -12.31   <2e-16 ***
## MedianIncome   4.254e+04  3.389e+02   125.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71650 on 20631 degrees of freedom
## Multiple R-squared:  0.6147, Adjusted R-squared:  0.6145 
## F-statistic:  4701 on 7 and 20631 DF,  p-value: < 2.2e-16

```

Como podemos observar, usemos la combinación que usemos no vamos a obtener un valor de r-squared más alto que usando la combinación de todas las variables, así que nos quedaremos con el modelo inicial, el que usaba todas las variables.

A continuación vamos a predecir los valores utilizando dicho modelo y vamos a comprobar el valor de RMSE que encontramos:

```

#predict(fit4,data.frame(MedianIncome=c(5,10,15)))
#yprime=predict(fit1,data.frame(lstat=Boston$lstat))
yprime=predict(fit.11,california)
sqrt(sum(abs(MedianHouseValue-yprime)^2)/length(yprime))

## [1] 69513.3

```

Como hemos podido apreciar el valor del RMSE es más bajo que el obtenido utilizando la regresión lineal simple.

## KNN

Ahora vamos a pasar a utilizar el algoritmo KNN para regresión. En primer lugar vamos el modelo para el conjunto de datos de california, usando la variable de salida “MedianHouseValue” con el resto.

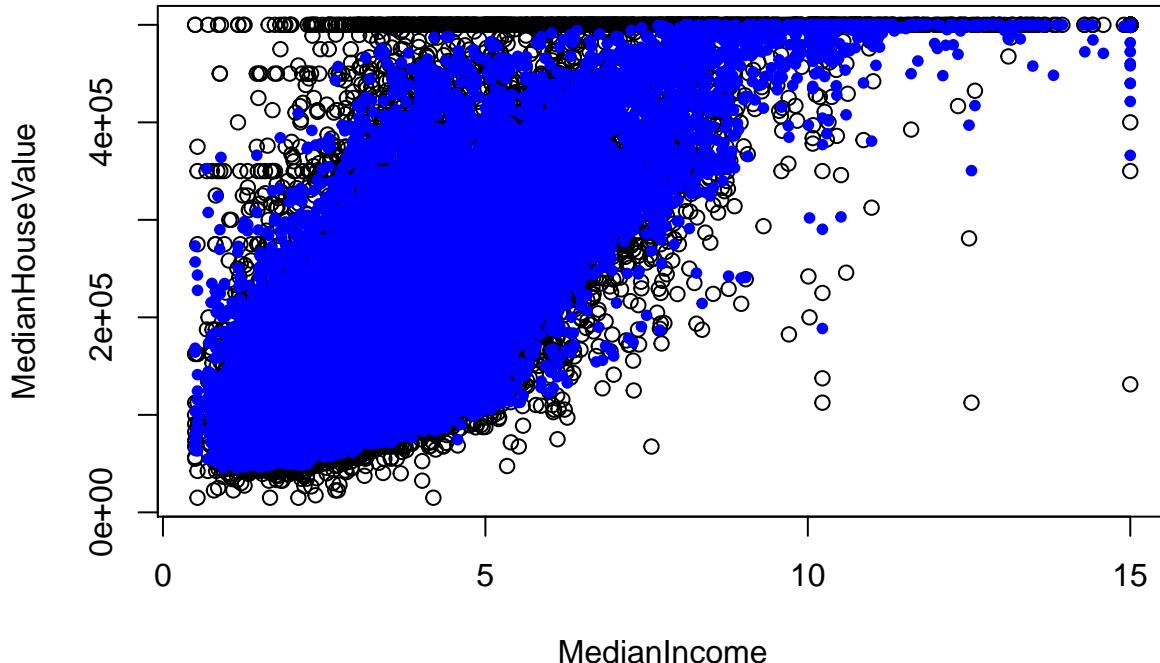
```

# Por defecto k = 7, distance = 2, kernel = "optimal"
# y scale=TRUE
fitknn1 <- kknn(MedianHouseValue ~ ., california, california)

```

A continuación vamos a visualizar los datos obtenidos, como sabemos que la mejor variable era “MedianIncome” vamos a visualizar los datos respecto a esta.

```
plot(MedianHouseValue~MedianIncome)
points(MedianIncome,fitknn1$fitted.values,col="blue",pch=20)
```



Ahora vamos a crear una predicción de los datos del conjunto con el modelo que hemos aprendido y posteriormente vamos a calcular de forma manual la raíz de ECM (RMSE).

```
yprime = fitknn1$fitted.values

sqrt(sum((california$MedianHouseValue-yprime)^2)/length(yprime))

## [1] 39132.79
```

El mejor modelo con la regresión lineal múltiple que habíamos obtenido utilizando la variable MedianHouseValue con todas las demás nos daba un error de 69513.3, aproximadamente casi el doble del que hemos obtenido utilizando Knn con la mejor variable.

Ahora vamos a intentar minimizar el error utilizando la información obtenida en regresión lineal acerca de las variables.

```
fitknn2 <- kknn(MedianHouseValue ~ MedianIncome*Latitude+I(MedianIncome^2) + HousingMedianAge, californi
yprime = fitknn2$fitted.values
sqrt(sum((california$MedianHouseValue-yprime)^2)/length(yprime)) #RMSE

## [1] 51457.83

fitknn3 <- kknn(MedianHouseValue ~ .+ MedianIncome*Latitude+I(MedianIncome^2) + HousingMedianAge, califo
yprime = fitknn3$fitted.values
sqrt(sum((california$MedianHouseValue-yprime)^2)/length(yprime)) #RMSE

## [1] 38954.25

fitknn4 <- kknn(MedianHouseValue ~ .+ MedianIncome*Latitude+I(MedianIncome^2) + HousingMedianAge - Popula
yprime = fitknn4$fitted.values
sqrt(sum((california$MedianHouseValue-yprime)^2)/length(yprime))
```

```

## [1] 42699.17
fitknn5 <- kknn(MedianHouseValue ~ . + MedianIncome*Latitude+I(MedianIncome^2) - Population - TotalBedrooms
yprime = fitknn5$fitted.values
sqrt(sum((california$MedianHouseValue-yprime)^2)/length(yprime))

## [1] 43124.39

Hemos obtenido un modelo que minimiza un poco el error que habíamos obtenido anteriormente, el fitknn3.
Vamos a intentar probar sin tener en cuenta lo aprendido por regresión lineal.

fitknn6 <- kknn(MedianHouseValue ~ . - Latitude, california, california)
yprime = fitknn6$fitted.values
sqrt(sum((california$MedianHouseValue-yprime)^2)/length(yprime))

## [1] 43485.3

fitknn7 <- kknn(MedianHouseValue ~ . - TotalBedrooms - HousingMedianAge, california, california)
yprime = fitknn7$fitted.values
sqrt(sum((california$MedianHouseValue-yprime)^2)/length(yprime))

## [1] 38938.17

fitknn8 <- kknn(MedianHouseValue ~ . - HousingMedianAge - Households , california, california)
yprime = fitknn8$fitted.values
sqrt(sum((california$MedianHouseValue-yprime)^2)/length(yprime))

## [1] 38824.73

```

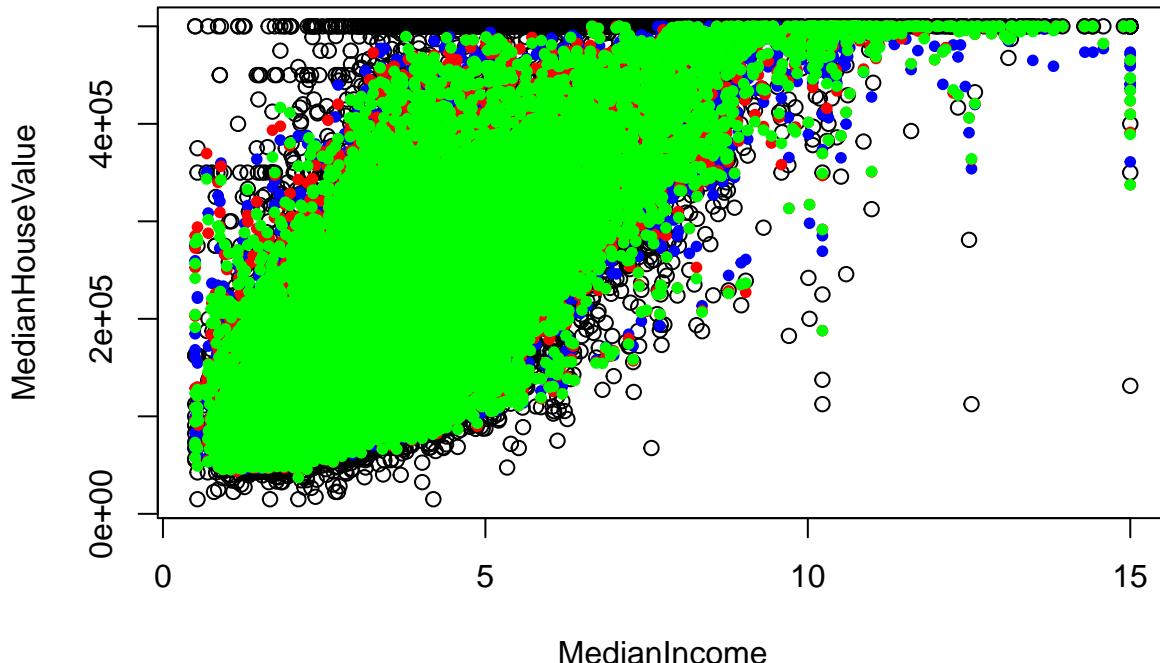
Hemos encontrado un modelo, el fitknn8, que nos da un RMSE más bajo, con un valor de 38824.73.

Ahora vamos a visualizar los datos.

```

plot(MedianHouseValue~MedianIncome)
points(MedianIncome,fitknn3$fitted.values,col="blue",pch=20)
points(MedianIncome,fitknn7$fitted.values,col="red",pch=20)
points(MedianIncome,fitknn8$fitted.values,col="green",pch=20)

```



Como podemos observar, el modelo 8 es el que mejor se ajustaría a nuestros datos del conjunto.

## K-fold Cross - Validation

A continuación ejecutaremos el algoritmo del k-fold cross-validation para obtener distintas medidas de error (error cuadrático medio MSE) sobre las mismas particiones para los distintos algoritmos. En primer lugar usaremos el modelo de regresión lineal con todas las variables.

```
nombre <- "california"

run_lm_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@")
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@")

  In <- length(names(x_tra)) - 1

  names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tst)[In+1] <- "Y"

  if (tt == "train") {
    test <- x_tra
  }
  else {
    test <- x_tst
  }
  fitMulti=lm(Y~.,x_tra)
  yprime=predict(fitMulti,test)
  sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}

lmMSEtrain = mean(sapply(1:5,run_lm_fold,nombre,"train"))
lmMSEtest = mean(sapply(1:5,run_lm_fold,nombre,"test"))

lmMSEtrain
```

```
## [1] 4826189710
```

```
lmMSEtest
```

```
## [1] 4844365688
```

A continuación usaremos el modelo Knn con todas las variables.

```
nombre <- "california"

run_knn_fold <- function(i, x, tt = "test") {
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@")
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@")

  In <- length(names(x_tra)) - 1
```

```

names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
names(x_tra)[In+1] <- "Y"
names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
names(x_tst)[In+1] <- "y"

if (tt == "train") {
  test <- x_tra
}
else {
  test <- x_tst
}
fitMulti=kknn(Y~.,x_tra,test)
yprime=fitMulti$fitted.values
sum(abs(test$Y-yprime)^2)/length(yprime)
}

knnMSEtrain = mean(sapply(1:5,run_knn_fold,nombre,"train"))
knnMSEtest = mean(sapply(1:5,run_knn_fold,nombre,"test"))

knnMSEtrain

## [1] 1560868807
knnMSEtest

## [1] 3845914481

```

De estos datos vemos que el modelo que menos error cuadrático medio para los datos del test genera es el realizado a partir de knn. Así que podemos decir que con Knn podremos ajustar un mejor modelo de regresión para el conjunto de datos california que utilizando el algoritmo de regresión lineal o regresión lineal simple.