

California

Laura

3 de diciembre de 2016

El conjunto de datos California

El conjunto de datos *California* contiene datos sobre viviendas de California y pretende estimar el precio de una nueva vivienda.

Las variables con las que se pretende estimar el precio de la vivienda son las siguientes:

- Longitud (*longitude*)
- Latitud (*latitude*)
- La edad media de las casas (*HousingMedianAge*)
- El número de habitaciones (*TotalRooms*)
- El número de dormitorios (*TotalBedrooms*)
- El número de habitantes (*Population*)
- El número de unidades familiares en el edificio (*Households*)
- La media de ingresos (*MedianIncome*)
- El valor medio de la casa (*MedianHouseValue*). El valor de esta variable es la que pretendemos obtener.

Hipótesis previas

Sin mirar el contenido del conjunto de datos se plantean las siguientes hipótesis:

1. El número de habitaciones incrementa el precio de forma lineal.
2. A mayor edad media más disminuye el precio.
3. A mayor población más incrementa el precio de la vivienda.
4. La distancia del centro disminuye el precio.
5. La media de ingresos hace aumenta el precio.
6. El número de unidades familiares en la vivienda hace disminuir el precio.

El conjunto de datos

La comprobación de las anteriores hipótesis la podemos realizar mirando como se comportan cada una de las variables con respecto a la salida.

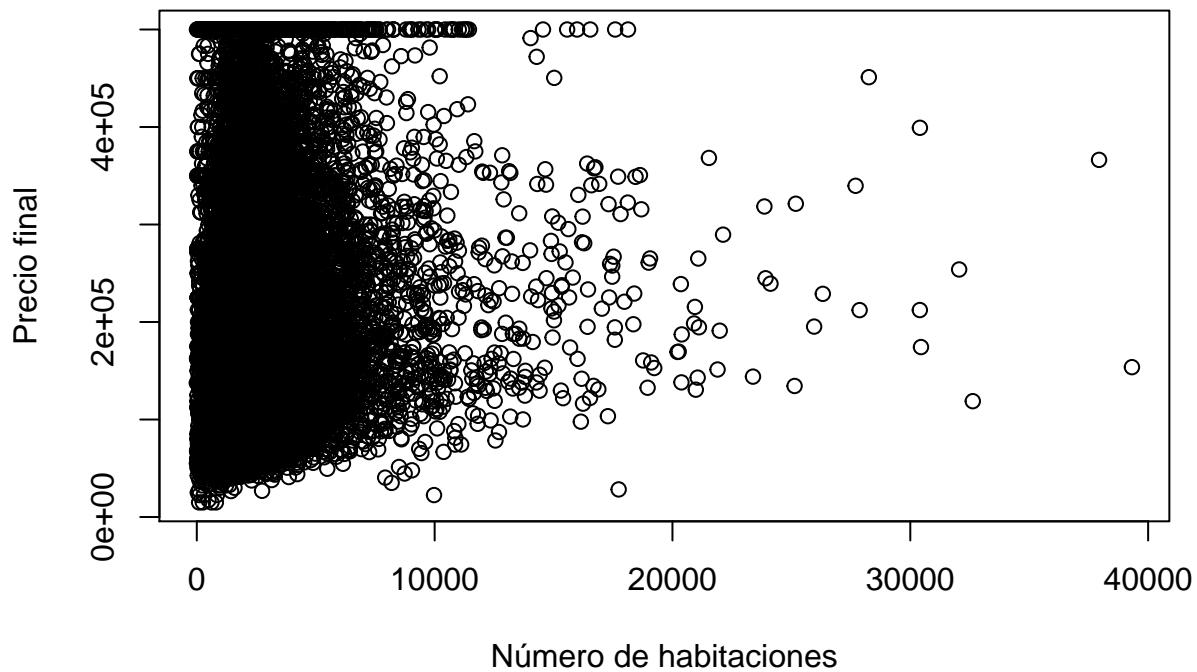
Para ello primero cargamos la base de datos

```
california_original <- read.csv("california.dat", header = FALSE,
  comment.char = "@")
names(california_original) <- c("Longitude", "Latitude", "HousingMedianAge",
  "TotalRooms", "TotalBedrooms", "Population", "Households",
  "MedianIncome", "MedianHouseValue")
```

Y mostramos las comparaciones con la salida de las variables anteriores

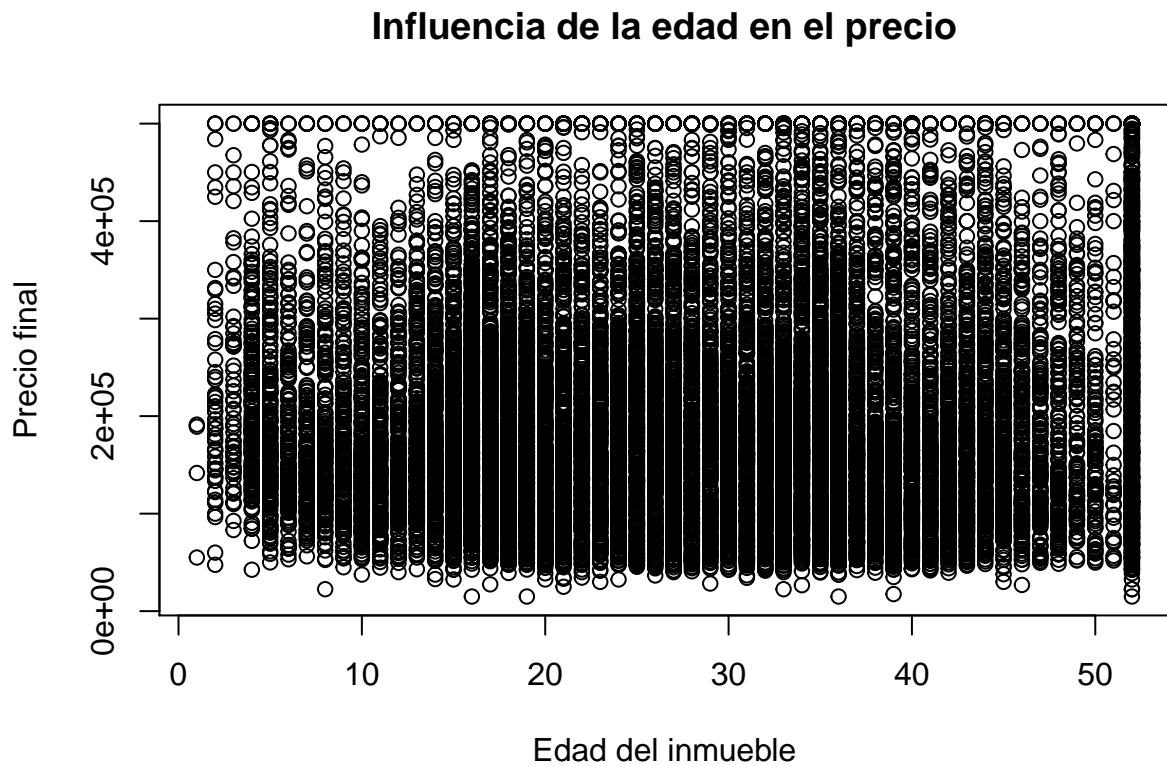
```
plot(california_original$TotalRooms, california_original$MedianHouseValue,
      main = "Influencia del número de habitaciones en el precio",
      xlab = "Número de habitaciones", ylab = "Precio final")
```

Influencia del número de habitaciones en el precio



Aparentemente no existe una relación lineal entre el número de habitaciones y el precio.

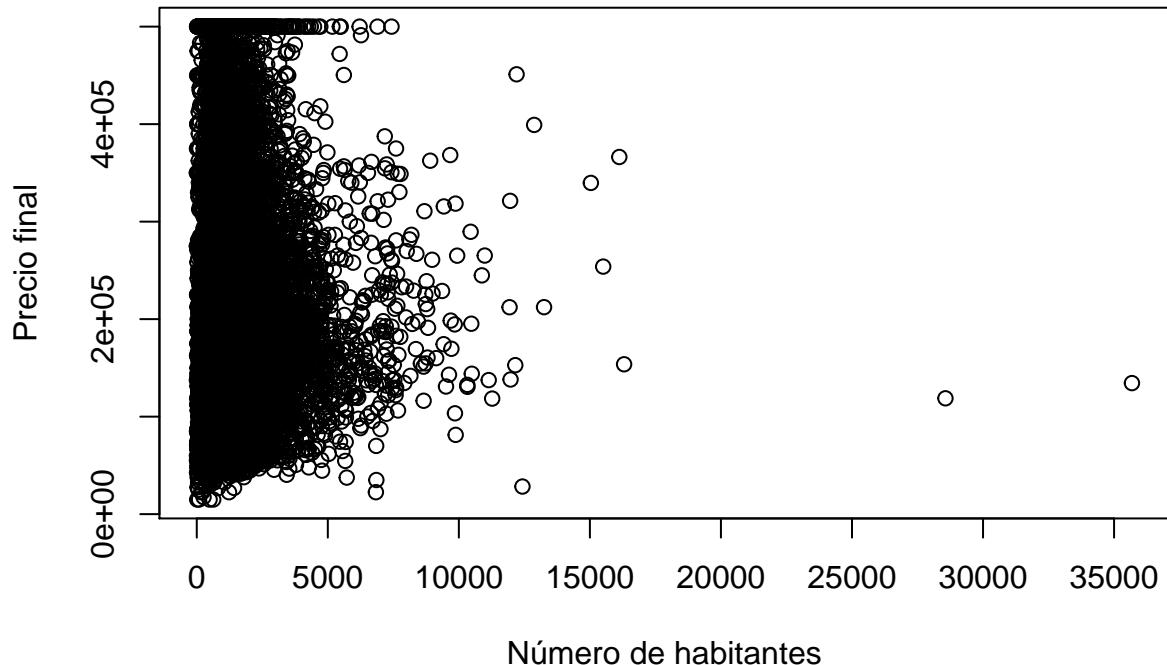
```
plot(california_original$HousingMedianAge, california_original$MedianHouseValue,  
     main = "Influencia de la edad en el precio", xlab = "Edad del inmueble",  
     ylab = "Precio final")
```



Esta variable tampoco tiene una relación lineal como habíamos deducido.

```
plot(california_original$Population, california_original$MedianHouseValue,  
      main = "Influencia del número de habitantes en el precio",  
      xlab = "Número de habitantes", ylab = "Precio final")
```

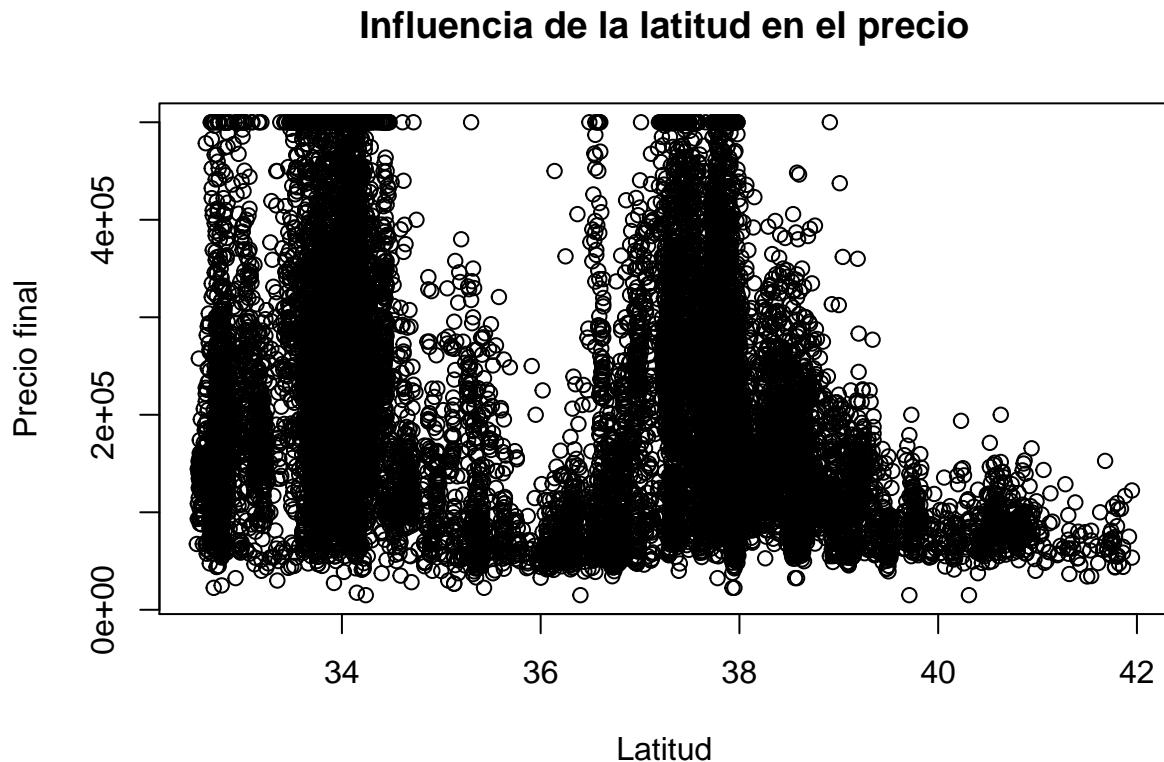
Influencia del número de habitantes en el precio



Esta variable tampoco tiene una relación lineal con la salida, pero tiene una forma similar al gráfico del número de habitaciones.

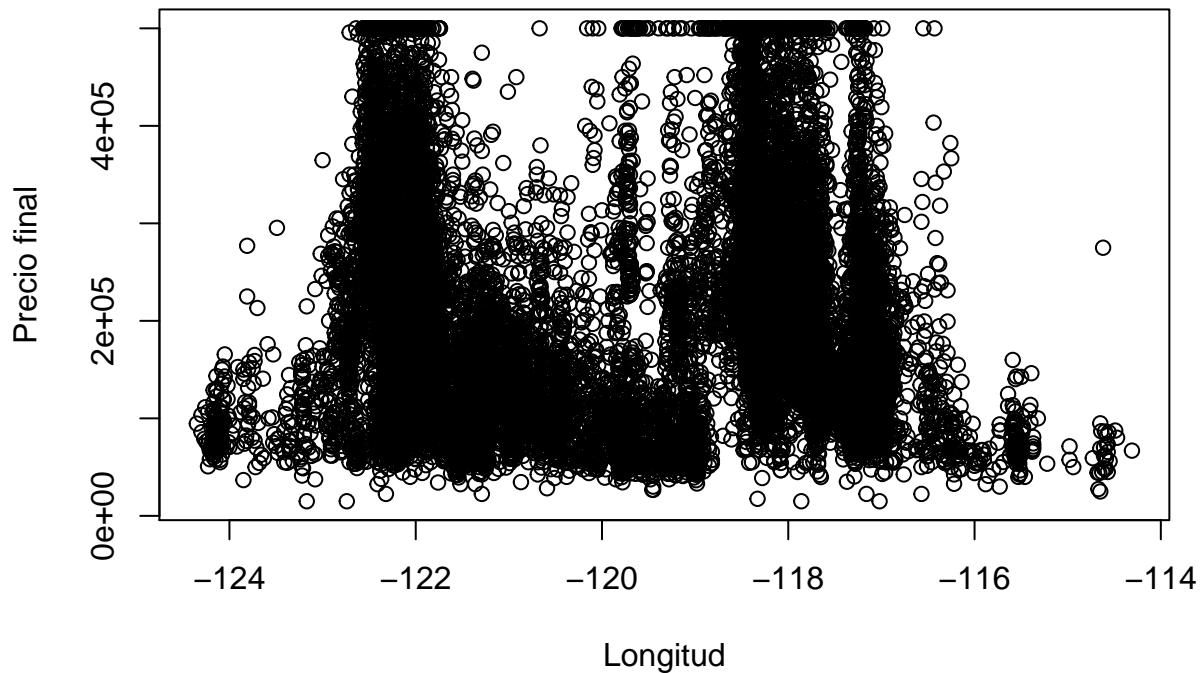
Para calcular la distancia al centro como hemos planteado en la hipótesis 4 tenemos que calcular un centro. Pero antes de eso miremos como se comportan las variables longitud y latitud por separado con respecto de la salida.

```
plot(california_original$Latitude, california_original$MedianHouseValue,  
      main = "Influencia de la latitud en el precio", xlab = "Latitud",  
      ylab = "Precio final")
```



```
plot(california_original$Longitude, california_original$MedianHouseValue,  
      main = "Influencia de la longitud en el precio", xlab = "Longitud",  
      ylab = "Precio final")
```

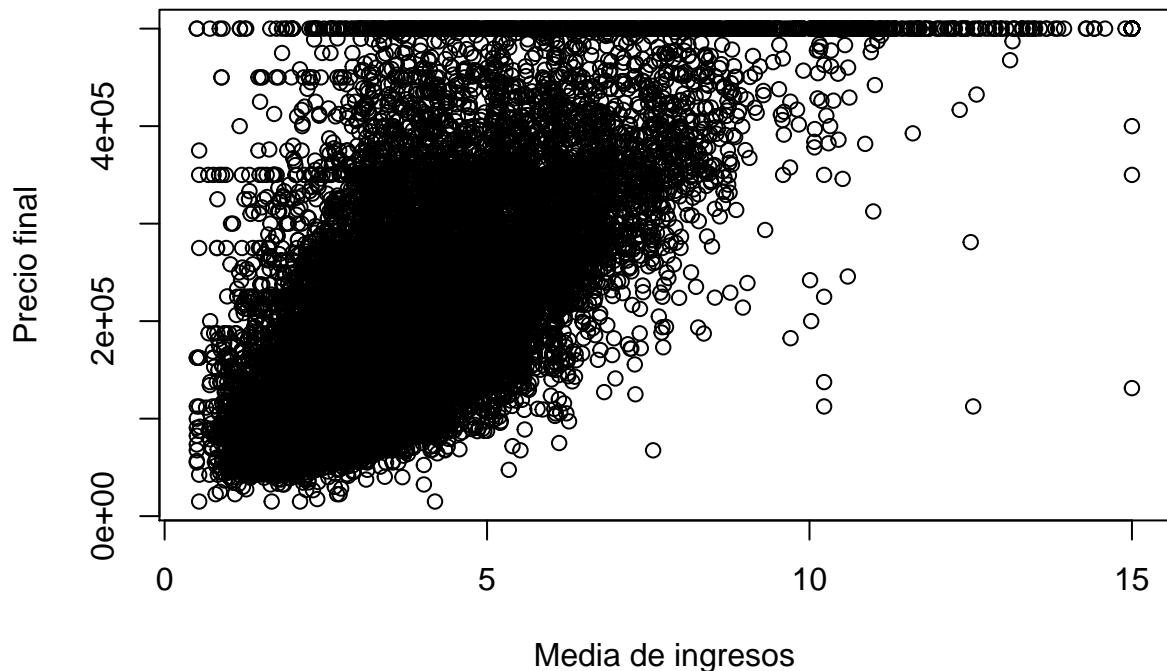
Influencia de la longitud en el precio



Claramente la relación latitud/longitud con el precio final no es lineal, pero el hecho de que existan varios picos me hace pensar que puede haber varias ciudades con lo que calcular la media para establecerlo como centro no es una buena opción.

```
plot(california_original$MedianIncome, california_original$MedianHouseValue,  
     main = "Influencia de la media de los ingresos en el precio",  
     xlab = "Media de ingresos", ylab = "Precio final")
```

Influencia de la media de los ingresos en el precio

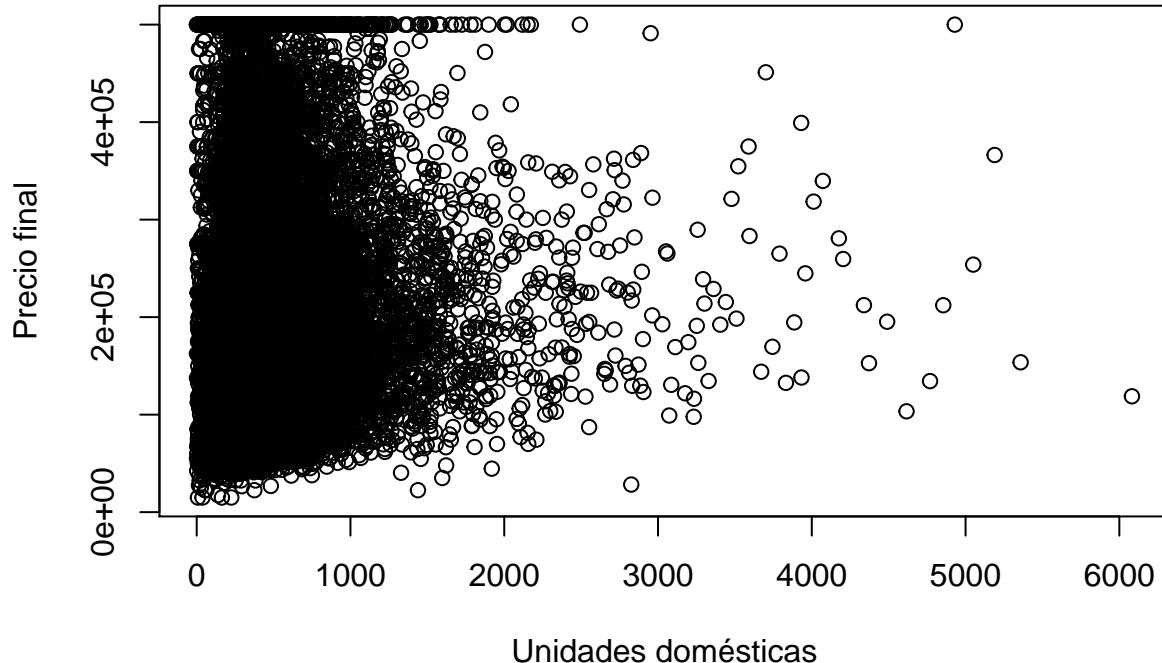


Puede ser una relación lineal con una dispersión muy alta, debido probablemente a la existencia de varias ciudades en el conjunto de datos.

Ya solo queda comprobar que al aumentar el número de unidades familiares disminuye el precio.

```
plot(california_original$Households, california_original$MedianHouseValue,
     main = "Influencia del número de unidades domésticas en el precio",
     xlab = "Unidades domésticas", ylab = "Precio final")
```

Influencia del número de unidades domésticas en el precio



No sigue una distribución lineal pero por la forma que tiene la hipótesis de que a mayor número de unidades domésticas disminuye también es falsa.

Conclusiones sobre las hipótesis previas.

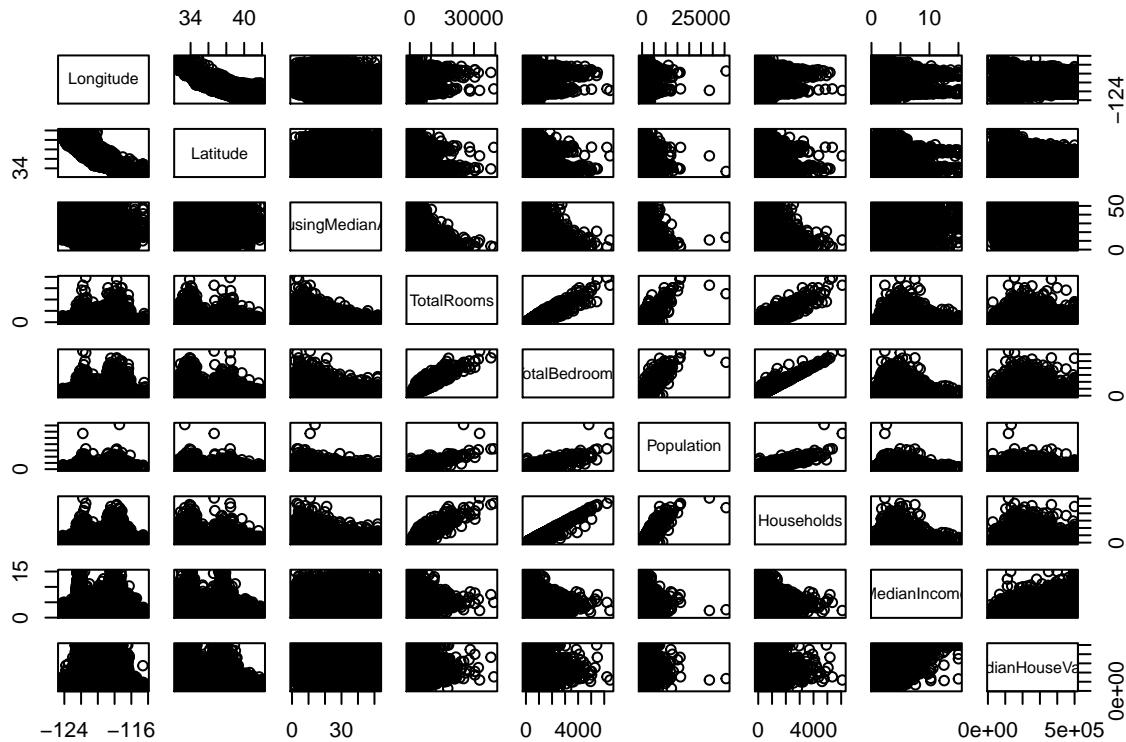
Tras ver la relación entre las variables de las hipótesis previas y la salida esperada, podemos llegar a las siguientes conclusiones:

- El conjunto de datos recoge varias ciudades.
- Al recoger varias ciudades las variables pueden presentar una tendencia lineal pero tener una amplia dispersión, relacionada con la ciudad en la que esté. ** Se debería clusterizar el conjunto de datos por ciudades. Pero el ejercicio es de modelos lineales por lo que no nos vamos a meter en la elaboración de clusters. Eso sí se espera que el modelo de regresión lineal sea bastante malo y que el modelo KNN presente mejores resultados.
- La única variable estudiada hasta ahora que parece independiente de la ciudad es la media de ingresos.

Comparación de todas las variables con la salida

Ahora compararemos todas las variables con todas para tener una idea de las interacciones que pueden existir entre ellas.

```
attach(california_original)
pairs(california_original)
```



Las variables *TotalRooms*, *TotalBedrooms*, *Population* y *Households* tienen una relación lineal entre ellas.

Construcción del modelo lineal

Separación del conjunto de train

Para comenzar a elaborar los modelos predictivos necesitamos separar los datos en train y test. En esta primera parte del ejercicio vamos a coger el 60% del conjunto para el entrenamiento y el 40% restante para validación.

```
train <- california_original[1:(dim(california_original)[1] *
  0.6), ]
test <- california_original[(dim(california_original)[1] * 0.6):dim(california_original)[1], ]

detach(california_original)
attach(train)
```

Modelo lineal simple

Una vez conocemos como se comportan las variables podemos plantear un modelo lineal que aproxime una solución a nuestro problema.

Podemos plantearlo como un modelo lineal de una sola variable, aunque sabemos que no va a ser un ajuste muy bueno pero de esta forma tendremos un parámetro de R^2 ajustada base para comparar más tarde.

La variable que se usará para este modelo es *MedianIncome* porque presentaba una tendencia lineal, más independiente que las demás estudiadas en el apartado anterior.

```
model1_singleLineal <- lm(MedianHouseValue ~ MedianIncome)
summary(model1_singleLineal)
```

Call:

```
lm(formula = MedianHouseValue ~ MedianIncome)
```

Residuals:

Min	1Q	Median	3Q	Max
-456316	-56084	-16655	36800	425413

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	45253	1716	26.37	<2e-16 ***
MedianIncome	41758	400	104.39	<2e-16 ***

Signif. codes:

```
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 83350 on 12382 degrees of freedom

Multiple R-squared: 0.4681, Adjusted R-squared: 0.4681

F-statistic: 1.09e+04 on 1 and 12382 DF, p-value: < 2.2e-16

El valor del parámetro R^2 es de **0.4681**, es bastante malo pero nos sirve como base para ir aumentando el número de variables que intervienen en el modelo de regresión lineal múltiple.

Modelo lineal múltiple

Las variables que vamos a añadir al modelo anterior serán *HouseHold*, *Population*, *TotalRooms*, *TotalBedRooms* una a una comprobando siempre el coeficiente Cuadrado para compararlo.

```
model1_multipleLineal <- lm(MedianHouseValue ~ MedianIncome +
Households)
summary(model1_multipleLineal)
```

Call:

```
lm(formula = MedianHouseValue ~ MedianIncome + Households)
```

Residuals:

Min	1Q	Median	3Q	Max
-448292	-56101	-16384	36217	423085

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
--	----------	------------	---------	----------

```
(Intercept) 37665.451   1964.970   19.17 < 2e-16 ***
MedianIncome 41700.483    399.116   104.48 < 2e-16 ***
Households      15.678     1.992    7.87 3.85e-15 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 83150 on 12381 degrees of freedom
Multiple R-squared: 0.4707, Adjusted R-squared: 0.4707
F-statistic: 5506 on 2 and 12381 DF, p-value: < 2.2e-16

Al añadir la variable Household la variación ha sido mínima en el modelo, ha pasado de 0.4681 a 0.4703
Como las variables _Population, TotalRooms, TotalBedRooms_ tenían todas una gráfica similar a _Households_.

```
model2_multipleLineal <- lm(MedianHouseValue ~ MedianIncome +
Households + Population + TotalRooms + TotalBedrooms)
summary(model2_multipleLineal)
```

Call:

```
lm(formula = MedianHouseValue ~ MedianIncome + Households + Population +
TotalRooms + TotalBedrooms)
```

Residuals:

Min	1Q	Median	3Q	Max
-500335	-51209	-12871	35338	433276

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	21247.997	2154.704	9.861	< 2e-16 ***
MedianIncome	47209.385	464.275	101.684	< 2e-16 ***
Households	167.159	10.254	16.302	< 2e-16 ***
Population	-40.921	1.645	-24.872	< 2e-16 ***
TotalRooms	-24.314	1.109	-21.922	< 2e-16 ***
TotalBedrooms	78.310	9.546	8.203	2.57e-16 ***

Signif. codes:

```
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 78590 on 12378 degrees of freedom
Multiple R-squared: 0.5273, Adjusted R-squared: 0.5271
F-statistic: 2761 on 5 and 12378 DF, p-value: < 2.2e-16

Añadir estas variables ha conseguido una mejora de casi un 0.5% del modelo. Puesto que sabemos que las variables anteriores probablemente dependen de la longitud y la latitud la incorporaremos dichas variables al modelo.

```
model3_multipleLineal <- lm(MedianHouseValue ~ MedianIncome +
Households + Population + TotalRooms + TotalBedrooms + Longitude +
Latitude)
summary(model3_multipleLineal)
```

Call:

```
lm(formula = MedianHouseValue ~ MedianIncome + Households + Population +
TotalRooms + TotalBedrooms + Longitude + Latitude)
```

Residuals:

Min	1Q	Median	3Q	Max
-409638	-43806	-11661	30201	487402

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-3.973e+06	7.934e+04	-50.08	< 2e-16 ***
MedianIncome	3.910e+04	4.421e+02	88.44	< 2e-16 ***
Households	6.852e+01	9.424e+00	7.27	3.8e-13 ***
Population	-4.344e+01	1.493e+00	-29.10	< 2e-16 ***
TotalRooms	-9.538e+00	1.036e+00	-9.21	< 2e-16 ***
TotalBedrooms	9.976e+01	8.634e+00	11.55	< 2e-16 ***
Longitude	-4.765e+04	8.977e+02	-53.08	< 2e-16 ***
Latitude	-4.686e+04	8.509e+02	-55.08	< 2e-16 ***

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 70390 on 12376 degrees of freedom
Multiple R-squared: 0.6209, Adjusted R-squared: 0.6206
F-statistic: 2895 on 7 and 12376 DF, p-value: < 2.2e-16

Modelos con interacciones y polinomiales

Añadir las variables longitud y latitud al modelo ha supuesto una notable mejora al modelo, pero vamos a añadir una nueva variable que sea una interacción de la longitud y la latitud a ver si ayuda a mejorar el modelo.

```
model4_multipleLineal <- lm(MedianHouseValue ~ MedianIncome +
  Households + Population + TotalRooms + TotalBedrooms + Longitude +
  Latitude + (Longitude * Latitude))
summary(model4_multipleLineal)
```

Call:

```
lm(formula = MedianHouseValue ~ MedianIncome + Households + Population +
  TotalRooms + TotalBedrooms + Longitude + Latitude + (Longitude *
  Latitude))
```

Residuals:

Min	1Q	Median	3Q	Max
-408607	-43581	-11818	30270	483829

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-7.284e+06	9.745e+05	-7.474	8.30e-14
MedianIncome	3.903e+04	4.424e+02	88.230	< 2e-16
Households	6.970e+01	9.427e+00	7.394	1.52e-13
Population	-4.386e+01	1.497e+00	-29.294	< 2e-16
TotalRooms	-9.386e+00	1.036e+00	-9.059	< 2e-16
TotalBedrooms	9.902e+01	8.633e+00	11.470	< 2e-16
Longitude	-7.521e+04	8.135e+03	-9.246	< 2e-16
Latitude	4.640e+04	2.737e+04	1.695	0.090102

```
Longitude:Latitude 7.756e+02 2.275e+02 3.409 0.000655
```

```
(Intercept) ***  
MedianIncome ***  
Households ***  
Population ***  
TotalRooms ***  
TotalBedrooms ***  
Longitude ***  
Latitude .  
Longitude:Latitude ***  
---  
Signif. codes:  
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 70360 on 12375 degrees of freedom  
Multiple R-squared: 0.6212, Adjusted R-squared: 0.621  
F-statistic: 2537 on 8 and 12375 DF, p-value: < 2.2e-16
```

La mejora del modelo es mínima y además aparece la variable latitud como la menos significativa del conjunto. Dicha variable aunque la quitemos seguirá apareciendo porque tenemos el término Longitud*Latitud que necesita de los términos de las dos componentes por separado, por ello se desarrollará un modelo por la vía polinomial.

Dado que las variables longitud y latitud visualizado con la gráfica adquieran una forma con varias cimas, podemos probar a utilizar el modelo anterior con polinomios sobre las variables latitud y longitud.

Vamos a poner un polinomio de grado 10 en cada una de las variables para comprobar en qué momento deja de ser interesante el miembro del polinomio.

```
model5_multipleLineal <- lm(MedianHouseValue ~ MedianIncome +  
Households + Population + TotalRooms + TotalBedrooms + Longitude +  
Latitude + poly(Longitude, 10) + poly(Latitude, 10))  
summary(model5_multipleLineal)
```

Call:

```
lm(formula = MedianHouseValue ~ MedianIncome + Households + Population +  
TotalRooms + TotalBedrooms + Longitude + Latitude + poly(Longitude,  
10) + poly(Latitude, 10))
```

Residuals:

Min	1Q	Median	3Q	Max
-370497	-41654	-9868	27234	469734

Coefficients: (2 not defined because of singularities)

	Estimate	Std. Error	t value
(Intercept)	-3.898e+06	1.064e+05	-36.626
MedianIncome	3.605e+04	4.354e+02	82.806
Households	4.972e+01	9.357e+00	5.314
Population	-4.464e+01	1.449e+00	-30.802
TotalRooms	-2.295e+00	1.017e+00	-2.256
TotalBedrooms	8.583e+01	8.544e+00	10.046
Longitude	-4.680e+04	1.214e+03	-38.547
Latitude	-4.579e+04	1.135e+03	-40.333
poly(Longitude, 10)1	NA	NA	NA

```

poly(Longitude, 10)2 -1.041e+06 8.983e+04 -11.590
poly(Longitude, 10)3 -6.475e+04 9.591e+04 -0.675
poly(Longitude, 10)4 3.478e+05 7.953e+04 4.373
poly(Longitude, 10)5 1.434e+06 8.412e+04 17.044
poly(Longitude, 10)6 -3.435e+05 7.180e+04 -4.785
poly(Longitude, 10)7 -5.919e+05 7.536e+04 -7.855
poly(Longitude, 10)8 2.358e+05 7.630e+04 3.091
poly(Longitude, 10)9 -5.197e+04 6.970e+04 -0.746
poly(Longitude, 10)10 3.275e+05 7.388e+04 4.433
poly(Latitude, 10)1 NA NA NA
poly(Latitude, 10)2 1.104e+06 9.596e+04 11.500
poly(Latitude, 10)3 -1.101e+05 1.006e+05 -1.095
poly(Latitude, 10)4 -4.787e+05 8.614e+04 -5.558
poly(Latitude, 10)5 9.285e+05 7.841e+04 11.842
poly(Latitude, 10)6 -5.302e+05 7.011e+04 -7.562
poly(Latitude, 10)7 -3.339e+05 7.222e+04 -4.623
poly(Latitude, 10)8 5.032e+05 6.945e+04 7.245
poly(Latitude, 10)9 3.121e+05 7.221e+04 4.322
poly(Latitude, 10)10 -8.602e+05 6.900e+04 -12.466
Pr(>|t|)
(Intercept) < 2e-16 ***
MedianIncome < 2e-16 ***
Households 1.09e-07 ***
Population < 2e-16 ***
TotalRooms 0.0241 *
TotalBedrooms < 2e-16 ***
Longitude < 2e-16 ***
Latitude < 2e-16 ***
poly(Longitude, 10)1 NA
poly(Longitude, 10)2 < 2e-16 ***
poly(Longitude, 10)3 0.4996
poly(Longitude, 10)4 1.23e-05 ***
poly(Longitude, 10)5 < 2e-16 ***
poly(Longitude, 10)6 1.73e-06 ***
poly(Longitude, 10)7 4.32e-15 ***
poly(Longitude, 10)8 0.0020 **
poly(Longitude, 10)9 0.4559
poly(Longitude, 10)10 9.37e-06 ***
poly(Latitude, 10)1 NA
poly(Latitude, 10)2 < 2e-16 ***
poly(Latitude, 10)3 0.2736
poly(Latitude, 10)4 2.79e-08 ***
poly(Latitude, 10)5 < 2e-16 ***
poly(Latitude, 10)6 4.26e-14 ***
poly(Latitude, 10)7 3.81e-06 ***
poly(Latitude, 10)8 4.58e-13 ***
poly(Latitude, 10)9 1.56e-05 ***
poly(Latitude, 10)10 < 2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 67080 on 12358 degrees of freedom
Multiple R-squared: 0.6561, Adjusted R-squared: 0.6554

F-statistic: 943.2 on 25 and 12358 DF, p-value: < 2.2e-16

Un polinomio de grado 10 en las variables longitud y latitud tiene la gran mayoría de sus componentes como significativos para el ajuste de la función a los datos. Pero hay algunos miembros como el de grado 3 en la variable latitud y el de grado 9 y 3 para la variable longitud que aparecen como no significativas. Por esto y por simplicidad se va a regresar a la versión anterior del modelo.

Aún podemos probar a elaborar un modelo polinómico en el que la variable sea la interacción entre la latitud y la longitud. Y al igual que en el modelo anterior vamos a realizar la prueba poniendo un polinomio de grado muy alto para ver la significatividad que tendrían cada uno de los polinomios.

```
model6_multipleLineal <- lm(MedianHouseValue ~ MedianIncome +
  Households + Population + TotalRooms + TotalBedrooms + Longitude +
  Latitude + poly(Longitude * Latitude, 10))
summary(model6_multipleLineal)
```

Call:

```
lm(formula = MedianHouseValue ~ MedianIncome + Households + Population +
  TotalRooms + TotalBedrooms + Longitude + Latitude + poly(Longitude *
  Latitude, 10))
```

Residuals:

Min	1Q	Median	3Q	Max
-386854	-41623	-10916	28359	440608

Coefficients:

	Estimate	Std. Error
(Intercept)	-1.106e+08	6.530e+06
MedianIncome	3.636e+04	4.373e+02
Households	6.199e+01	9.130e+00
Population	-4.613e+01	1.456e+00
TotalRooms	-3.885e+00	1.017e+00
TotalBedrooms	8.401e+01	8.373e+00
Longitude	-4.990e+05	2.769e+04
Latitude	1.432e+06	9.037e+04
poly(Longitude * Latitude, 10)1	4.449e+08	2.720e+07
poly(Longitude * Latitude, 10)2	3.794e+06	2.413e+05
poly(Longitude * Latitude, 10)3	3.656e+04	7.495e+04
poly(Longitude * Latitude, 10)4	-1.023e+06	6.900e+04
poly(Longitude * Latitude, 10)5	-1.191e+06	7.257e+04
poly(Longitude * Latitude, 10)6	5.610e+04	6.804e+04
poly(Longitude * Latitude, 10)7	1.000e+06	7.116e+04
poly(Longitude * Latitude, 10)8	5.089e+05	6.938e+04
poly(Longitude * Latitude, 10)9	-2.910e+05	7.281e+04
poly(Longitude * Latitude, 10)10	-1.531e+05	6.935e+04
	t value	Pr(> t)
(Intercept)	-16.943	< 2e-16 ***
MedianIncome	83.148	< 2e-16 ***
Households	6.789	1.18e-11 ***
Population	-31.676	< 2e-16 ***
TotalRooms	-3.821	0.000133 ***
TotalBedrooms	10.034	< 2e-16 ***
Longitude	-18.019	< 2e-16 ***
Latitude	15.847	< 2e-16 ***
poly(Longitude * Latitude, 10)1	16.357	< 2e-16 ***

```

poly(Longitude * Latitude, 10)2  15.723 < 2e-16 ***
poly(Longitude * Latitude, 10)3   0.488 0.625677
poly(Longitude * Latitude, 10)4  -14.828 < 2e-16 ***
poly(Longitude * Latitude, 10)5  -16.409 < 2e-16 ***
poly(Longitude * Latitude, 10)6   0.825 0.409637
poly(Longitude * Latitude, 10)7   14.054 < 2e-16 ***
poly(Longitude * Latitude, 10)8    7.335 2.36e-13 ***
poly(Longitude * Latitude, 10)9  -3.997 6.46e-05 ***
poly(Longitude * Latitude, 10)10  -2.207 0.027298 *
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 67750 on 12366 degrees of freedom
Multiple R-squared: 0.6491, Adjusted R-squared: 0.6486
F-statistic: 1345 on 17 and 12366 DF, p-value: < 2.2e-16

El modelo mejora, poco pero mejora. Puesto que no todos los miembros del polinomio tienen un nivel de significación suficiente se probará con modelos donde el polinomio sea menor.

```

model7_multipleLineal <- lm(MedianHouseValue ~ MedianIncome +
  Households + Population + TotalRooms + TotalBedrooms + Longitude +
  Latitude + poly(Longitude * Latitude, 5))
summary(model7_multipleLineal)

```

Call:

```

lm(formula = MedianHouseValue ~ MedianIncome + Households + Population +
  TotalRooms + TotalBedrooms + Longitude + Latitude + poly(Longitude *
  Latitude, 5))

```

Residuals:

Min	1Q	Median	3Q	Max
-394571	-41866	-11277	27873	448092

Coefficients:

	Estimate	Std. Error
(Intercept)	-9.880e+07	6.212e+06
MedianIncome	3.722e+04	4.381e+02
Households	6.227e+01	9.221e+00
Population	-4.374e+01	1.463e+00
TotalRooms	-6.286e+00	1.016e+00
TotalBedrooms	8.925e+01	8.448e+00
Longitude	-4.503e+05	2.632e+04
Latitude	1.263e+06	8.606e+04
poly(Longitude * Latitude, 5)1	3.947e+08	2.589e+07
poly(Longitude * Latitude, 5)2	3.458e+06	2.299e+05
poly(Longitude * Latitude, 5)3	-3.217e+04	7.514e+04
poly(Longitude * Latitude, 5)4	-1.053e+06	6.968e+04
poly(Longitude * Latitude, 5)5	-1.100e+06	7.293e+04
	t value	Pr(> t)
(Intercept)	-15.906	< 2e-16 ***
MedianIncome	84.961	< 2e-16 ***
Households	6.752	1.52e-11 ***
Population	-29.903	< 2e-16 ***

```

TotalRooms           -6.186 6.38e-10 ***
TotalBedrooms        10.565 < 2e-16 ***
Longitude          -17.112 < 2e-16 ***
Latitude            14.680 < 2e-16 ***
poly(Longitude * Latitude, 5)1 15.245 < 2e-16 ***
poly(Longitude * Latitude, 5)2 15.038 < 2e-16 ***
poly(Longitude * Latitude, 5)3 -0.428   0.669
poly(Longitude * Latitude, 5)4 -15.114 < 2e-16 ***
poly(Longitude * Latitude, 5)5 -15.084 < 2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 68480 on 12371 degrees of freedom
Multiple R-squared: 0.6413, Adjusted R-squared: 0.6409
F-statistic: 1843 on 12 and 12371 DF, p-value: < 2.2e-16

Aún aparece el miembro de grado 3 como poco significativo por lo que vamos a reducir el grado del polinomio a 3, aprovechando que solamente se perdieron unas centésimas en el valor de R^2.

```

model8_multipleLineal <- lm(MedianHouseValue ~ MedianIncome +
    Households + Population + TotalRooms + TotalBedrooms + Longitude +
    Latitude + poly(Longitude * Latitude, 3))
summary(model8_multipleLineal)

```

Call:

```

lm(formula = MedianHouseValue ~ MedianIncome + Households + Population +
    TotalRooms + TotalBedrooms + Longitude + Latitude + poly(Longitude *
    Latitude, 3))

```

Residuals:

Min	1Q	Median	3Q	Max
-398348	-42464	-10823	29039	466499

Coefficients:

	Estimate	Std. Error
(Intercept)	-8.469e+07	6.098e+06
MedianIncome	3.889e+04	4.388e+02
Households	5.910e+01	9.391e+00
Population	-4.335e+01	1.490e+00
TotalRooms	-8.459e+00	1.030e+00
TotalBedrooms	1.036e+02	8.577e+00
Longitude	-3.905e+05	2.586e+04
Latitude	1.068e+06	8.438e+04
poly(Longitude * Latitude, 3)1	3.358e+08	2.540e+07
poly(Longitude * Latitude, 3)2	2.992e+06	2.277e+05
poly(Longitude * Latitude, 3)3	-1.053e+05	7.594e+04
	t value	Pr(> t)
(Intercept)	-13.888	< 2e-16 ***
MedianIncome	88.633	< 2e-16 ***
Households	6.293	3.23e-10 ***
Population	-29.097	< 2e-16 ***
TotalRooms	-8.211	2.42e-16 ***
TotalBedrooms	12.084	< 2e-16 ***

```

Longitude           -15.099 < 2e-16 ***
Latitude            12.653 < 2e-16 ***
poly(Longitude * Latitude, 3)1 13.221 < 2e-16 ***
poly(Longitude * Latitude, 3)2 13.141 < 2e-16 ***
poly(Longitude * Latitude, 3)3 -1.387   0.166
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 69770 on 12373 degrees of freedom
Multiple R-squared:  0.6276,    Adjusted R-squared:  0.6273
F-statistic: 2085 on 10 and 12373 DF,  p-value: < 2.2e-16

```

No ha sido una buena idea puesto que se ha perdido un 0.2 en el valor de R^2 , sin embargo este nuevo valor de R^2 es similar a la del modelo 3 que era mucho más simple y sencillo de explicar.

Por tanto es con este modelo 3 con el que vamos a realizar las pruebas con el conjunto de test.

Predicción de valores

Ahora que tenemos escogido el modelo lo probaremos sobre el conjunto de datos que hemos apartado como conjunto de test.

```

yprime <- predict(model4_multipleLineal, test)
sqrt(sum(abs(test$MedianHouseValue - yprime)^2)/length(yprime))

```

```
[1] 71265.99
```

Lo que nos devuelve la última llamada es el *root-mean-square-error(RMSE)* también conocido como la raíz del error cuadrático medio, similar a la desviación standar de la predicción.

El valor anterior es semejante a la décima parte del valor máximo que puede tomar el valor de las casas. Pero es el mejor modelo lineal que hemos obtenido.

Explicación del modelo lineal

En el modelo lineal escogido intervienen todas las variables registradas, es decir, en el precio final de una vivienda interviene la media de los ingresos de la zona, el número de unidades domésticas en el inmueble, el número de habitantes, el número total de habitaciones y de dormitorios así como la ciudad en la que está que se descompone en longitud y latitud.

Regresión usando KNN

La técnica de los k-vecinos más cercanos nos permite crear devolver un valor de predicción basáandonos en los datos ya existentes.

Paquetes necesarios

Para utilizar la técnica del vecino más cercano tenemos que utilizar funciones de la librería knn y MASS por lo que le indicamos a R que las necesitamos

```

require("MASS")
require("knn")

```

Creando el modelo

Ahora para crear el modelo realizamos la llamada a la función knn con la base de datos de California completa.

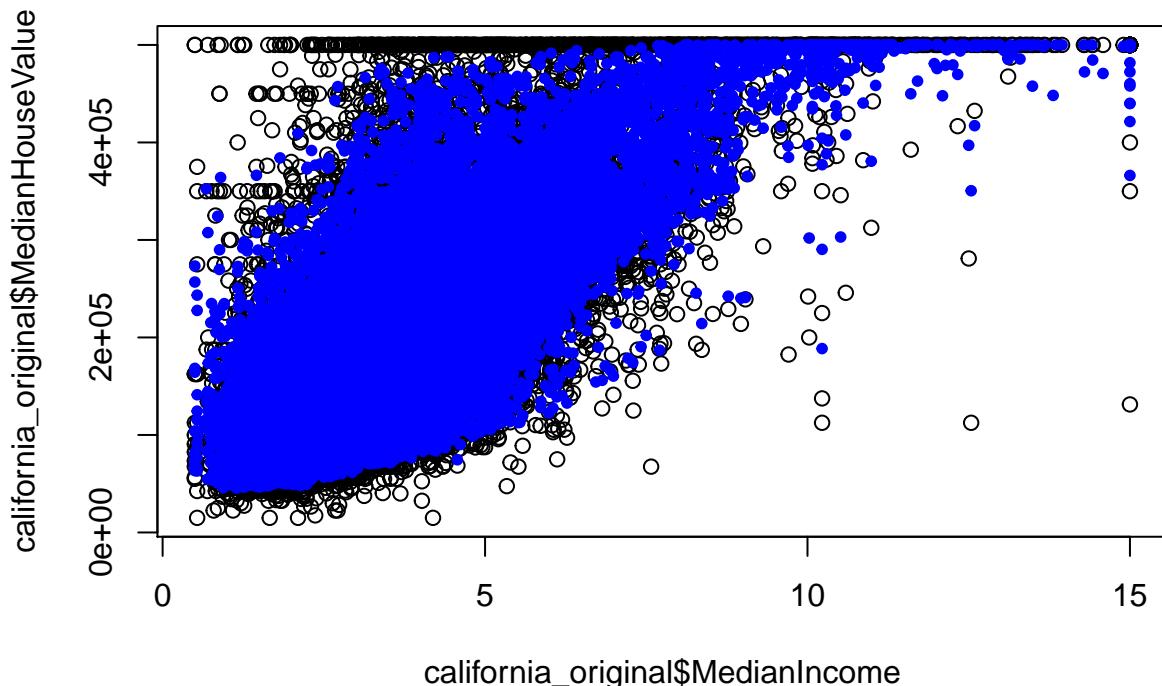
En este caso el número de vecinos más cercanos que se escogerán será de 7 y la distancia euclídea y se realizará el escalado de los datos para que tengan todos el mismo rango de valores.

```
fitknn1 <- knn(california_original$MedianHouseValue ~ ., california_original,  
california_original)
```

Este nuevo modelo tendrá en su componente “fitted.values” los valores obtenidos en la fase de test.

Antes de calcular su bondad vamos a visualizar los resultados.

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



No es un modelo perfecto pero es bastante bueno. Para compararlo con el mejor de los lineales vamos a calcular el RMSE

```
yprime_2 = fitknn1$fitted.values  
sqrt(sum((california_original$MedianHouseValue - yprime_2)^2)/length(yprime_2))  
  
[1] 39131.14
```

Como podemos ver el RMSE es significativamente menor que el obtenido por el considerado el mejor modelo de regression lineal múltiple, que tenía de ,valor RMSE 71274.62

Comprobemos que este error sigue bajando si le ponemos un modelo semejante al que desarrollamos con el

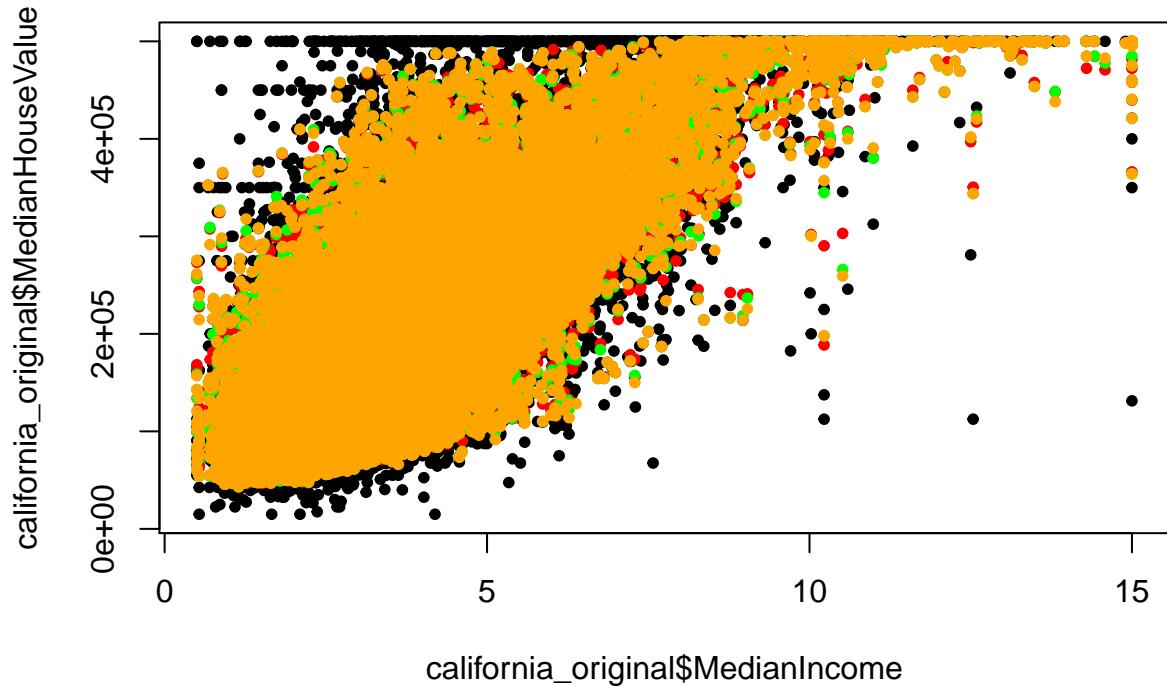
modelo lineal múltiple si añadimos polinomios de un grado superior, en este caso vamos a añadir un polinomo de grado 3 y de grado 5 para ver la evolución del error.

```
fitknn2 <- kknn(california_original$MedianHouseValue ~ . + poly(california_original$Latitude *  
    california_original$Longitude, 3), california_original, california_original)  
yprime_3 = fitknn2$fitted.values  
sqrt(sum((california_original$MedianHouseValue - yprime_3)^2)/length(yprime_3))  
  
[1] 38077.49  
  
fitknn3 <- kknn(california_original$MedianHouseValue ~ . + poly(california_original$Latitude *  
    california_original$Longitude, 5), california_original, california_original)  
yprime_4 = fitknn3$fitted.values  
sqrt(sum((california_original$MedianHouseValue - yprime_4)^2)/length(yprime_4))  
  
[1] 37789.83
```

Vemos en los resultados anteriores que al igual que los modelos homólogos del apartado anterior podemos ver una mejora del error. Pero mejor ilustrarlo con un gráfico:

```
plot(california_original$MedianHouseValue ~ california_original$MedianIncome,  
    pch = 20, main = "Comparación entre modelos Knn")  
  
points(california_original$MedianIncome, fitknn1$fitted.values,  
    col = "red", pch = 20)  
points(california_original$MedianIncome, fitknn2$fitted.values,  
    col = "green", pch = 20)  
points(california_original$MedianIncome, fitknn3$fitted.values,  
    col = "orange", pch = 20)
```

Comparación entre modelos Knn



Pero por simplicidad nos quedamos con el modelo que no tiene ningún tipo de polinomio en el modelo.