

# California Housing Prices

*Diego Alejandro Mernies*

*Primer cuatrimestre 2019*

## Introducción

Realizaremos una análisis de los datos de las viviendas que se encuentran en un distrito determinado de **California** y algunas estadísticas basadas en el censo de **1990**.

### Objetivo.

Predecir el precio de la casa de la época con un modelo de regresión lineal. El problema es de aprendizaje supervisado.

En primer lugar cargamos las librerías requeridas. Si no las tiene en su sistema, puede instalarlas con `install.packages("librería")`.

```
library(readr)
library(dplyr)
library(corrplot)
library(ggplot2)
library(scales)
library(rms)
```

## Definición de constantes

A continuación se define las **constantes** que se utilizarán en el proyecto.

```
# URL donde reside el dataset a utilizar
dataurl <- "https://raw.githubusercontent.com/dmerniestic1987/tp_ciencia_datos_california_housing/master/california_housing.csv"
# Ubicación local en donde se guardará el dataset para su procesamiento
datadir <- "~/workspace/R/data"
```

## Carga de datos

### Set de datos

El set de datos es un archivo .csv (comma separated value) de exactamente 10 columnas y 20641 filas de las cuales la primera contiene los nombres. El archivo de input original se llama **housing.csv** y se tomó de [California Housing Price] (<https://www.kaggle.com/camnugent/california-housing-prices>), pero fue subido a un repositorio GIT para simplificar la descarga de los datos y controlar las versiones. El repositorio GIT se puede explorar ingresando a: [https://github.com/dmerniestic1987/tp\\_ciencia\\_datos\\_california\\_housing](https://github.com/dmerniestic1987/tp_ciencia_datos_california_housing).

Los datos pertenecen a las casas que se encuentran en un distrito de California y algunas estadísticas basadas en los datos del censo de 1990. Las variables son:

Variable	Descripción
longitude	Qué tan lejos al oeste este está una casa. Un valor más alto está más al oeste.
latitude	Qué tan lejos al norte está una casa. Un valor más alto está más al norte.
housing_median_age	Edad media de una casa dentro de un bloque de casas. Un número más bajo es un edificio más nuevo.
total_rooms	Número total de ambientes dentro de un bloque de casas.
total_bedrooms	Número total de habitaciones dentro de un bloque de casas.
population	Número total de personas que residen dentro de un bloque de casas.
households	Número total de hogares, un grupo de personas que residen dentro de una unidad de hogar, por un bloque.
median_income	Ingreso promedio para hogares dentro de un bloque de casas (medido en decenas de miles de dólares estadounidenses)
median_house_value	Valor medio de la vivienda para hogares dentro de un bloque (medido en dólares estadounidenses)
ocean_proximity	Ubicación de la casa con relación al océano o mar.

La variable a predecir es **median\_house\_value**. ## Descarga del set de datos Descargamos los datos.

```
datafile <- paste(datadir, "housing.csv", sep = "/")
```

En primer lugar se verifica si es necesario crear un directorio para almacenar el archivo.

```
if (dir.exists(datadir)) {
  print(paste("El directorio ", datadir, " ya existe."))
} else {
  print(paste("Creando directorio de datos", datadir, "."))
  dir.create(datadir)
}
```

```
## [1] "El directorio ~/workspace/R/data ya existe."
```

El segundo lugar se descarga la última versión del archivo para poder utilizar la información actualizada.

```
if (file.exists(datafile)) {
  print(paste("El archivo ", datafile, " ya existe, lo elimino."))
  file.remove(datafile)
}

download.file(dataurl, datafile, method="auto")
```

## Lectura de los datos

Leemos el archivo recientemente descargado convirtiendo los espacios vacíos en N/A. Las columnas sin información no necesitan ser eliminadas dado que el resto de la información del futbolista puede ser útil.

```
dfhousing <- read.csv(datafile)
```

## Control de datos

Verificamos que los nombres de las columnas sean correctos.

```
colnames(dfhousing)
```

```
## [1] "longitude"          "latitude"           "housing_median_age"  
## [4] "total_rooms"         "total_bedrooms"      "population"  
## [7] "households"          "median_income"       "median_house_value"  
## [10] "ocean_proximity"
```

Verificamos la dimensión y los tipos de datos de las columnas.

```
dim(dfhousing)
```

```
## [1] 20640    10
```

```
str(dfhousing)
```

```
## 'data.frame': 20640 obs. of 10 variables:  
## $ longitude : num -122 -122 -122 -122 -122 ...  
## $ latitude : num 37.9 37.9 37.9 37.9 37.9 ...  
## $ housing_median_age: num 41 21 52 52 52 52 52 52 42 52 ...  
## $ total_rooms : num 880 7099 1467 1274 1627 ...  
## $ total_bedrooms : num 129 1106 190 235 280 ...  
## $ population : num 322 2401 496 558 565 ...  
## $ households : num 126 1138 177 219 259 ...  
## $ median_income : num 8.33 8.3 7.26 5.64 3.85 ...  
## $ median_house_value: num 452600 358500 352100 341300 342200 ...  
## $ ocean_proximity : Factor w/ 5 levels "<1H OCEAN","INLAND",...: 4 4 4 4 4 4 4 4 4 4 ...
```

Verificamos el contenido de los primeros y últimos registros del archivo.

```
head(dfhousing, give.attr=FALSE)
```

```
##   longitude latitude housing_median_age total_rooms total_bedrooms  
## 1    -122.23     37.88                 41        880        129  
## 2    -122.22     37.86                 21       7099       1106  
## 3    -122.24     37.85                 52       1467        190  
## 4    -122.25     37.85                 52       1274        235  
## 5    -122.25     37.85                 52       1627        280  
## 6    -122.25     37.85                 52        919        213  
##   population households median_income median_house_value ocean_proximity  
## 1        322        126     8.3252        452600    NEAR BAY  
## 2        2401       1138     8.3014        358500    NEAR BAY  
## 3        496        177     7.2574        352100    NEAR BAY  
## 4        558        219     5.6431        341300    NEAR BAY  
## 5        565        259     3.8462        342200    NEAR BAY  
## 6        413        193     4.0368        269700    NEAR BAY
```

```
tail(dfhousing, give.attr=FALSE)
```

```
##   longitude latitude housing_median_age total_rooms total_bedrooms  
## 20635    -121.56     39.27                 28        2332        395  
## 20636    -121.09     39.48                 25        1665        374  
## 20637    -121.21     39.49                 18        697         150  
## 20638    -121.22     39.43                 17        2254        485  
## 20639    -121.32     39.43                 18        1860        409  
## 20640    -121.24     39.37                 16        2785        616
```

```

##      population households median_income median_house_value
## 20635        1041       344     3.7125          116800
## 20636         845       330     1.5603           78100
## 20637         356       114     2.5568           77100
## 20638        1007       433     1.7000          92300
## 20639         741       349     1.8672          84700
## 20640        1387       530     2.3886          89400
##      ocean_proximity
## 20635          INLAND
## 20636          INLAND
## 20637          INLAND
## 20638          INLAND
## 20639          INLAND
## 20640          INLAND

```

Obtenemos un resumen de las variables para verificar los datos.

```
summary(dfhousing)
```

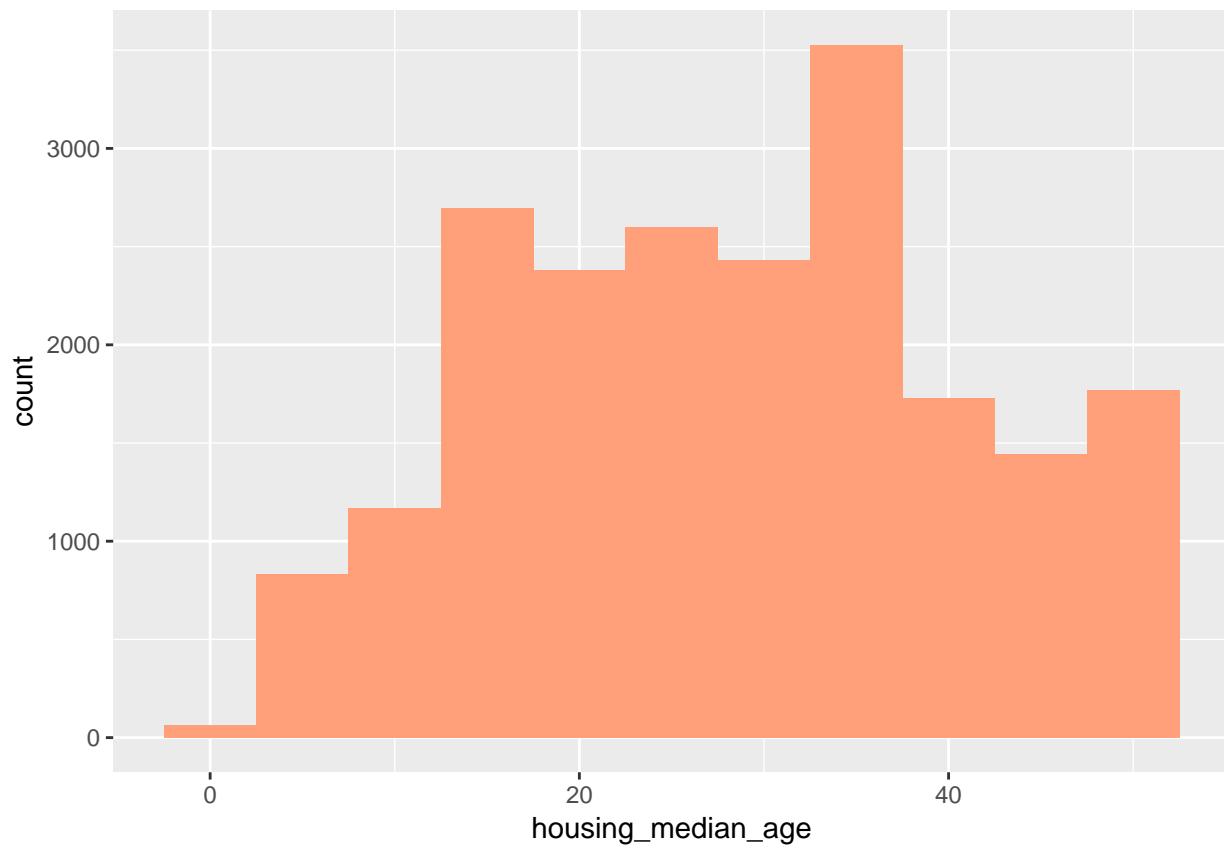
```

##      longitude      latitude   housing_median_age total_rooms
## Min.    :-124.3   Min.    :32.54      Min.    : 1.00      Min.    :  2
## 1st Qu.:-121.8   1st Qu.:33.93      1st Qu.:18.00      1st Qu.: 1448
## Median :-118.5   Median :34.26      Median :29.00      Median : 2127
## Mean   :-119.6   Mean   :35.63      Mean   :28.64      Mean   : 2636
## 3rd Qu.:-118.0   3rd Qu.:37.71      3rd Qu.:37.00      3rd Qu.: 3148
## Max.   :-114.3   Max.   :41.95      Max.   :52.00      Max.   :39320
##
##      total_bedrooms population households median_income
## Min.    : 1.0    Min.    :  3    Min.    : 1.0    Min.    : 0.4999
## 1st Qu.: 296.0   1st Qu.: 787   1st Qu.: 280.0   1st Qu.: 2.5634
## Median : 435.0   Median : 1166   Median : 409.0   Median : 3.5348
## Mean   : 537.9   Mean   : 1425   Mean   : 499.5   Mean   : 3.8707
## 3rd Qu.: 647.0   3rd Qu.: 1725   3rd Qu.: 605.0   3rd Qu.: 4.7432
## Max.   :6445.0   Max.   :35682   Max.   :6082.0   Max.   :15.0001
## NA's   :207
##      median_house_value ocean_proximity
## Min.    : 14999    <1H OCEAN :9136
## 1st Qu.:119600    INLAND    :6551
## Median :179700    ISLAND    :  5
## Mean   :206856    NEAR BAY  :2290
## 3rd Qu.:264725    NEAR OCEAN:2658
## Max.   :500001
## 
```

Del resumen estadísticos se detectó:  
\* Es necesario limpiar los NA'S de la columna total\_bedrooms.  
\* Existen sólo 5 casas que están en una Isla.  
\* Los valores máximos de total\_rooms, total\_bedrooms, population y households son muy altos en comparación a la media.

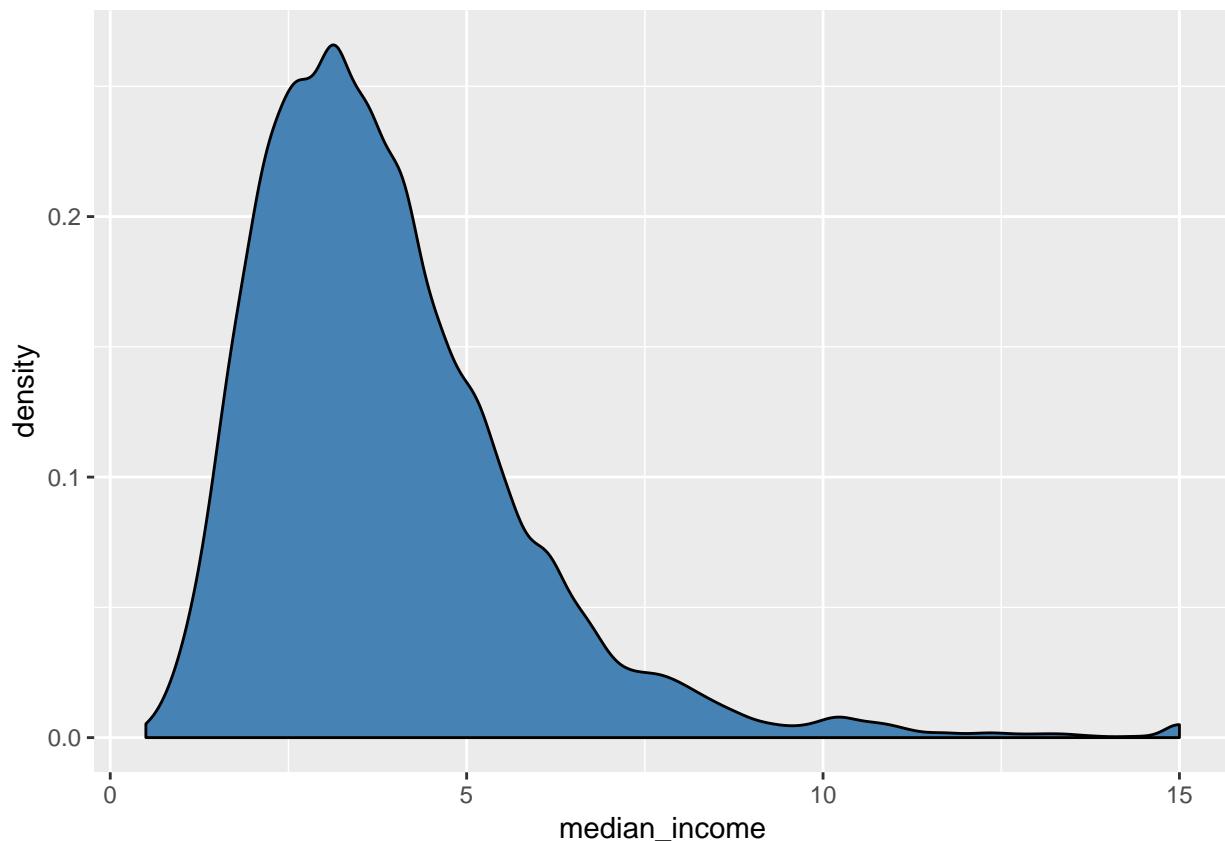
## Visualización de datos

```
ggplot(dfhousing) +
  geom_histogram(aes(x=housing_median_age), binwidth=5, fill="lightsalmon", bins=30)
```



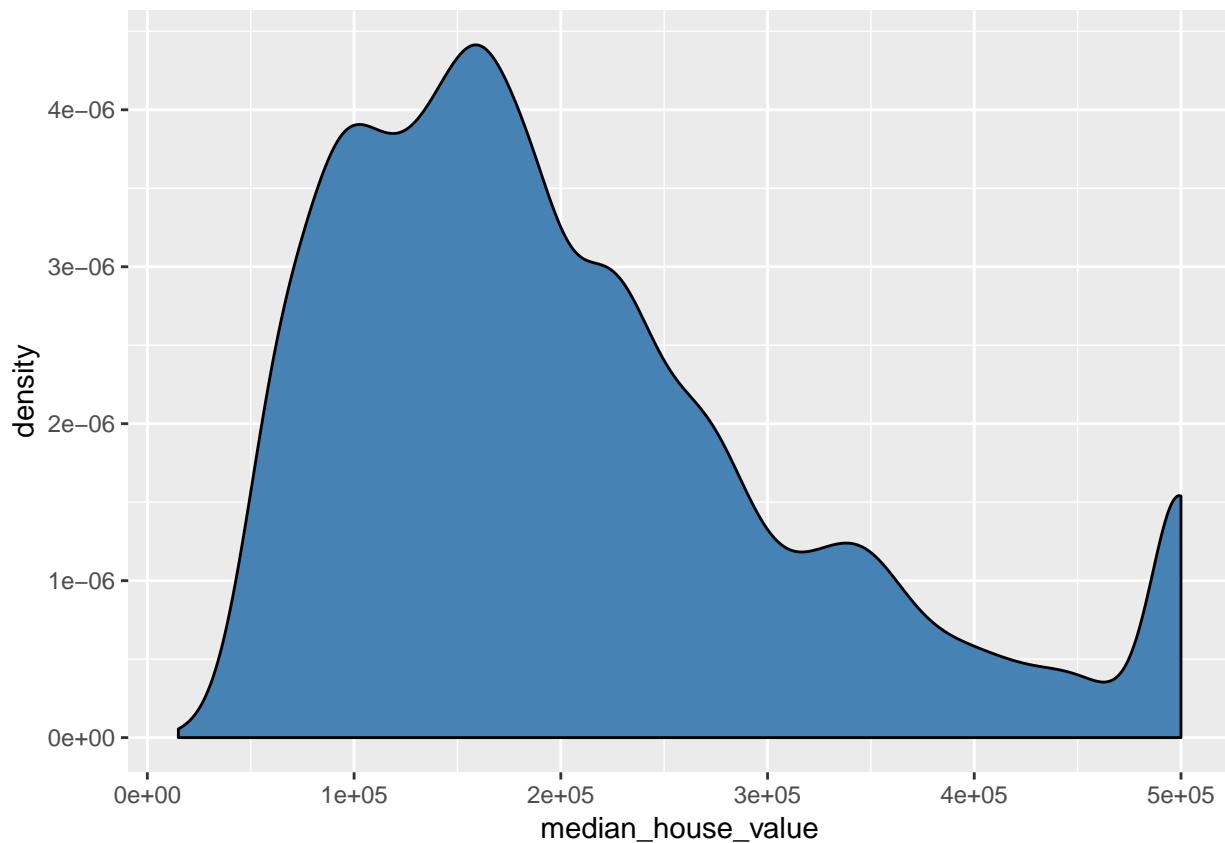
Observamos que en el comunto de datos las casas tenían entre 1 y 52 años de antigüedad.

```
ggplot(dfhousing) +  
  geom_density(aes(x=median_income), fill="steel blue")
```



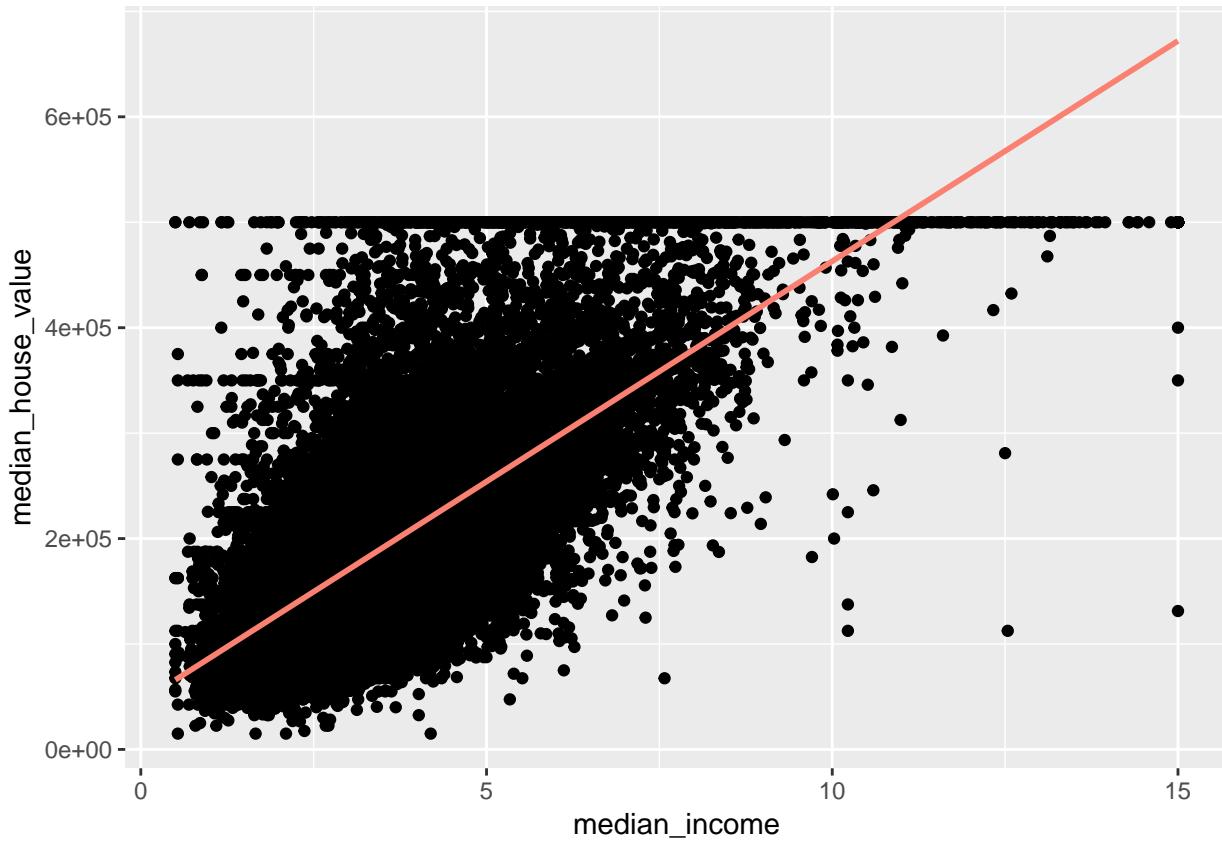
En este gráfico se observa que hay pocas personas que tiene altos ingresos en la población analizada.

```
ggplot(dfhousing) +  
  geom_density(aes(x=median_house_value), fill="steel blue")
```



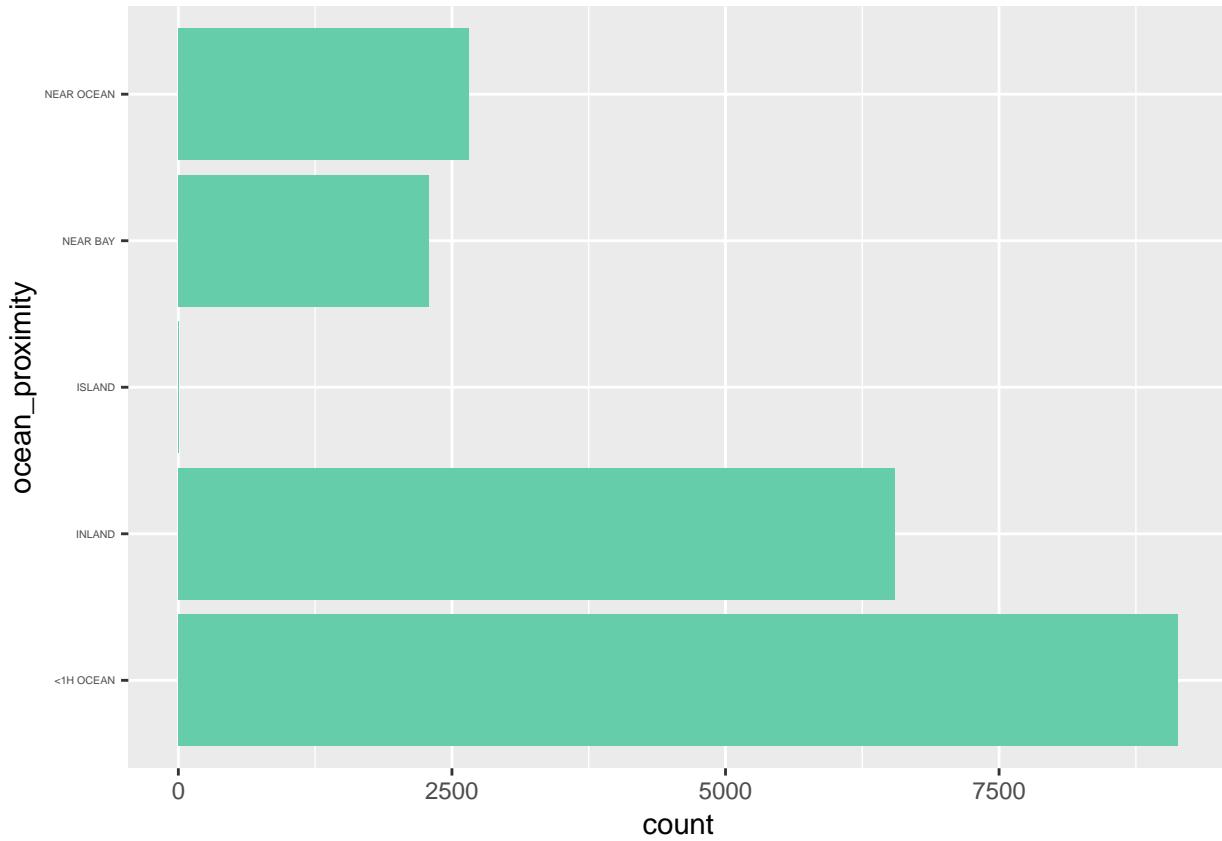
En el gráfico anterior observamos que la mayoría de las casas tienen un precio inferior a los 30.000 dólares estadounidenses, siendo la menor cantidad comprendido entre los 40.000 y 46.000 dólares aproximadamente.

```
ggplot(dfhousing) +  
  geom_point(aes(x = median_income, y = median_house_value)) +  
  stat_smooth(aes(x = median_income, y = median_house_value), method = "lm", color = "salmon", se = F)
```



En base a este gráfico podemos observar que en promedio las casas no superan los U\$S 500.000 y esto podría afectar al modelo porque hay un límite claramente marcado, pero desconocemos si ese límite fue puesto a propósito por lo que los eliminaremos para nuestro análisis.

```
ggplot(dfhousing) +
  geom_bar(aes(x=ocean_proximity), fill="mediumaquamarine") +
  coord_flip() +
  theme(axis.text.y=element_text(size=rel(0.4)))
```



## Limpieza de datos

Se eliminan los NA de **total\_bedrooms**

```
dfhousing$total_bedrooms[is.na(dfhousing$total_bedrooms)] = median(dfhousing$total_bedrooms, na.rm=TRUE)
```

Filtramos los datos para eliminar los valores máximos de **median\_house\_value** para eliminar las casa que tienen un precio mayor o igual a U\$s 500.000

```
dfhousing <- dfhousing[dfhousing$median_house_value < 500000,]
```

Para corregir los altos valores máximos se crean dos nuevas columnas: *mean\_bedrooms*: El cuociente entre habitaciones por hogares. *mean\_rooms*: El cuociente entre ambientes por hogares. Posteriormente eliminamos las columnas **total\_bedrooms** y **total\_rooms**

```
dfhousing$mean_bedrooms = dfhousing$total_bedrooms / dfhousing$households
dfhousing$mean_rooms = dfhousing$total_rooms / dfhousing$households
```

```
#Eliminamos las columnas total_bedrooms y total_rooms para usar el nuevo índice
drops = c('total_bedrooms', 'total_rooms', 'longitude', 'latitude')
dfhousing = dfhousing[ , !(names(dfhousing) %in% drops) ]
```

Controlamos la nueva estructura de la tabla

```
head(dfhousing)
```

```
##   housing_median_age population households median_income
## 1                 41        322       126      8.3252
## 2                 21       2401      1138      8.3014
```

```

## 3          52      496      177      7.2574
## 4          52      558      219      5.6431
## 5          52      565      259      3.8462
## 6          52      413      193      4.0368
##   median_house_value ocean_proximity mean_bedrooms mean_rooms
## 1          452600    NEAR BAY     1.0238095  6.984127
## 2          358500    NEAR BAY     0.9718805  6.238137
## 3          352100    NEAR BAY     1.0734463  8.288136
## 4          341300    NEAR BAY     1.0730594  5.817352
## 5          342200    NEAR BAY     1.0810811  6.281853
## 6          269700    NEAR BAY     1.1036269  4.761658

#Creamos un nuevo dataframe auxiliar sin ocean_proximity que es categórico, ni median_house_value
#que el dato que se intentará predecir. Luego se escalan los valores para trabajarlos con gráficos más
#También eliminamos las columnas latitud y longitud ya que no las usamos.
drops = c('ocean_proximity', 'median_house_value', 'latitude', 'longitude')
dfhousing_aux = dfhousing[, !(names(dfhousing) %in% drops)]
dfscaledhousing_aux = scale(dfhousing_aux)

#Creamos un nuevo dataframe que contenga solo la proximidad al mar. Luego limpiamos el resto de las col
dropsCategories = c('ocean_proximity', 'median_house_value')
dfcat_aux = dfhousing[, !(names(dfhousing) %in% dropsCategories)]

#Combinamos los dataframes y generamos un nuevo que contenga la combinación con los datos escalados y l
dfhousing_clean = cbind(DataSet1=dfcat_aux, DataSet2=dfscaledhousing_aux, median_house_value=dfhousing$median_house_value)
dropClean = c('DataSet1.median_house_value')
dfhousing_clean = dfhousing_clean[, !(names(dfhousing_clean) %in% dropClean)]

newNames = c('ocean_proximity', 'housing_median_age', 'population', 'households', 'median_income', 'mean_income')
colnames(dfhousing_clean) <- newNames

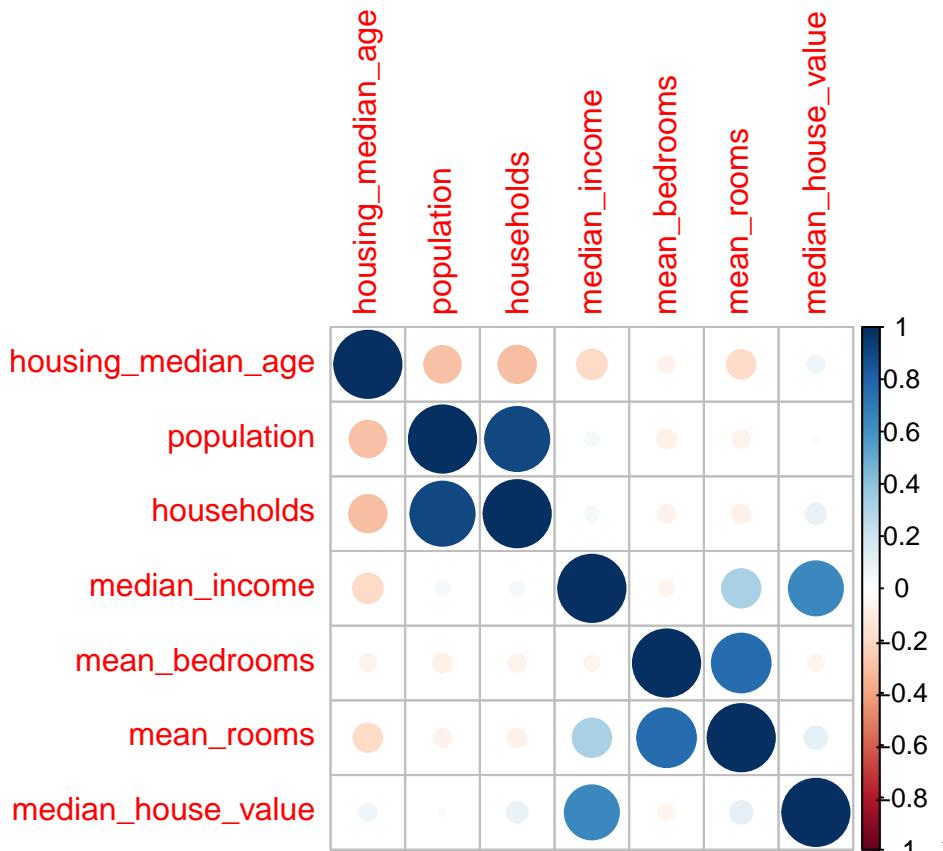
```

## Visualización de correlaciones

```

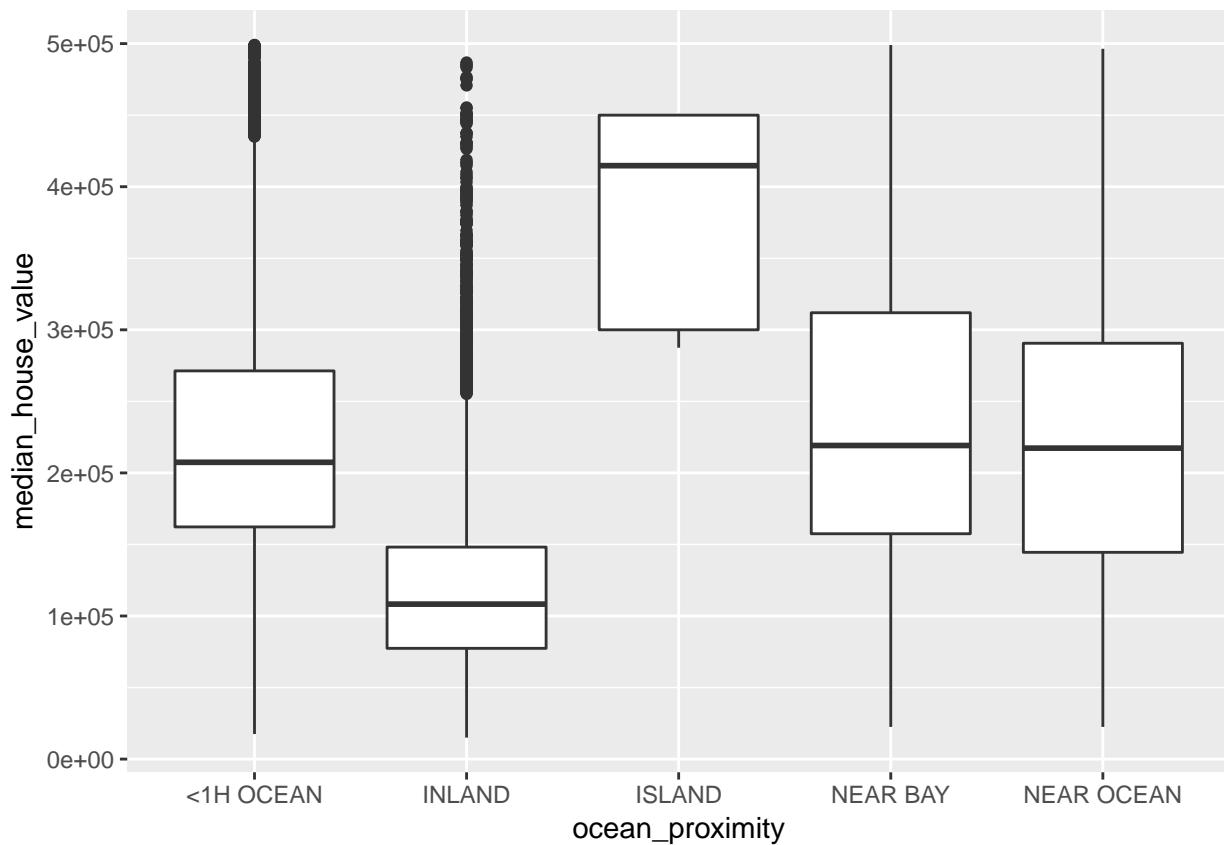
dfhousing_clean %>%
  select_if(is.numeric) %>%
  cor() %>%
  corrplot()

```



En base al gráfico de correlación se puede determinó que existe una fuerte relación entre la cantidad de la población y la cantidad de hogares. También que el precio de las casas está relacionado con el ingreso medio por lo que pueden haber barrios más caros.

```
ggplot(dfhousing_clean) +
  geom_boxplot(aes(x=ocean_proximity, y=median_house_value))
```



En este gráfico podemos observar que la proximidad al mar y al océano impactan fuertemente en el precio de la propiedad.

## Construcción de modelos

### Creación de datos de entrenamiento y pruebas

Creamos dos datasets: Uno de entrenamiento y otro de pruebas utilizando la proporción 80 para entrenamiento y 20 para test.

```
set.seed(77222)
seleccion <- runif(dim(dfhousing_clean)[1])

dftrain <- select(dfhousing_clean, population, median_house_value, median_income, mean_rooms, households)
dftest <- select(dfhousing_clean, population, median_house_value, median_income, mean_rooms, households)
```

### Modelo 1.

El primer modelo utiliza los predictores: **median\_income**, **mean\_rooms** y **population** para intentar predecir el valor de las propiedades plasmados en **median\_house\_value**.

```
lineModel <- lm(median_house_value~median_income+mean_rooms+population, data=dftrain)
summary(lineModel)
```

```
## 
## Call:
```

```

## lm(formula = median_house_value ~ median_income + mean_rooms +
##     population, data = dftrain)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -535377 -50017 -13915  36035 503241
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 192561.6    584.3 329.577 < 2e-16 ***
## median_income 67281.2    618.4 108.796 < 2e-16 ***
## mean_rooms   -10406.6    607.8 -17.121 < 2e-16 ***
## population    -2557.8    581.1  -4.402 1.08e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73360 on 15763 degrees of freedom
## Multiple R-squared:  0.4363, Adjusted R-squared:  0.4362
## F-statistic:  4066 on 3 and 15763 DF,  p-value: < 2.2e-16
vif(lineModel)

## median_income      mean_rooms      population
##           1.117881          1.121178          1.009090

```

El coeficiente **R-squared** de 0.4846 del modelo nos dice que el modelo explica el 48.46% de la varianza total de la variable en la regresión.

El coeficiente **Adjusted R-squared** nos dice que el 48.45% de la variable dependiente (median\_house\_value) es explicado por las variables independientes.

Los factores de inflación de la varianza tomados tienen valores aceptables ya que son menores a 10.

```

predModel <- predict(lineModel, newdata = dftrain)
summary(predModel)

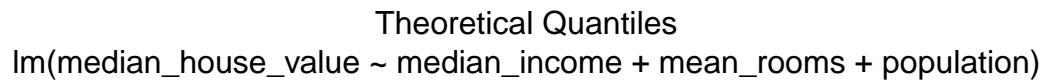
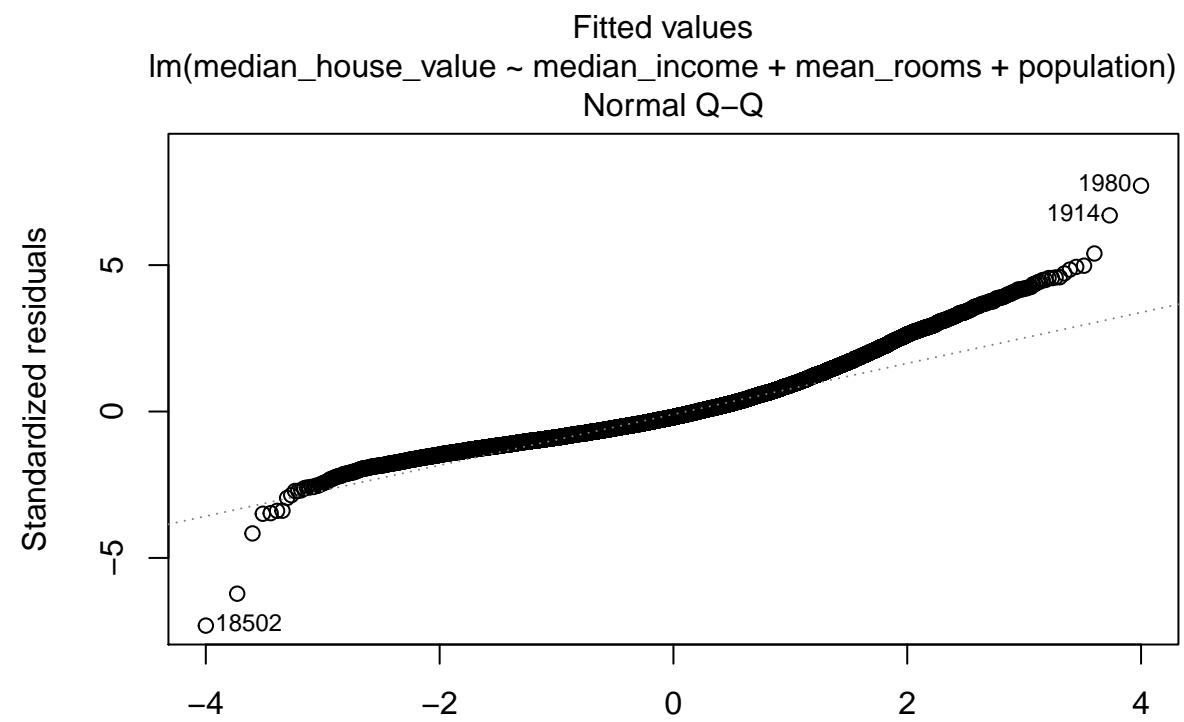
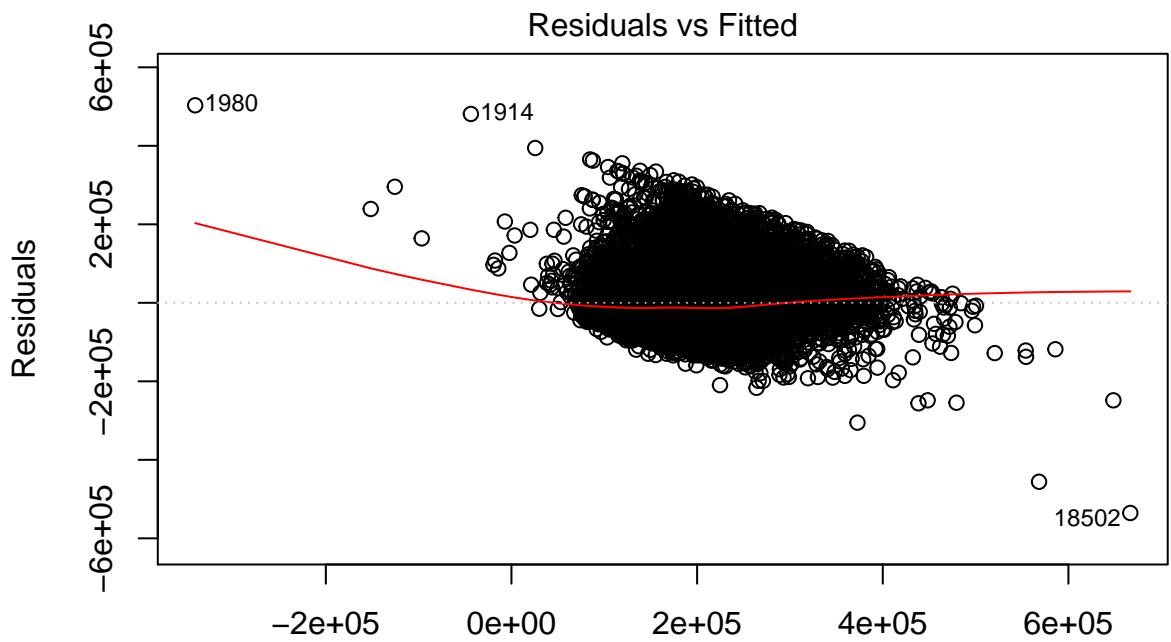
##      Min. 1st Qu. Median  Mean 3rd Qu.  Max.
## -340741 145121 183300 192263 229394 666677

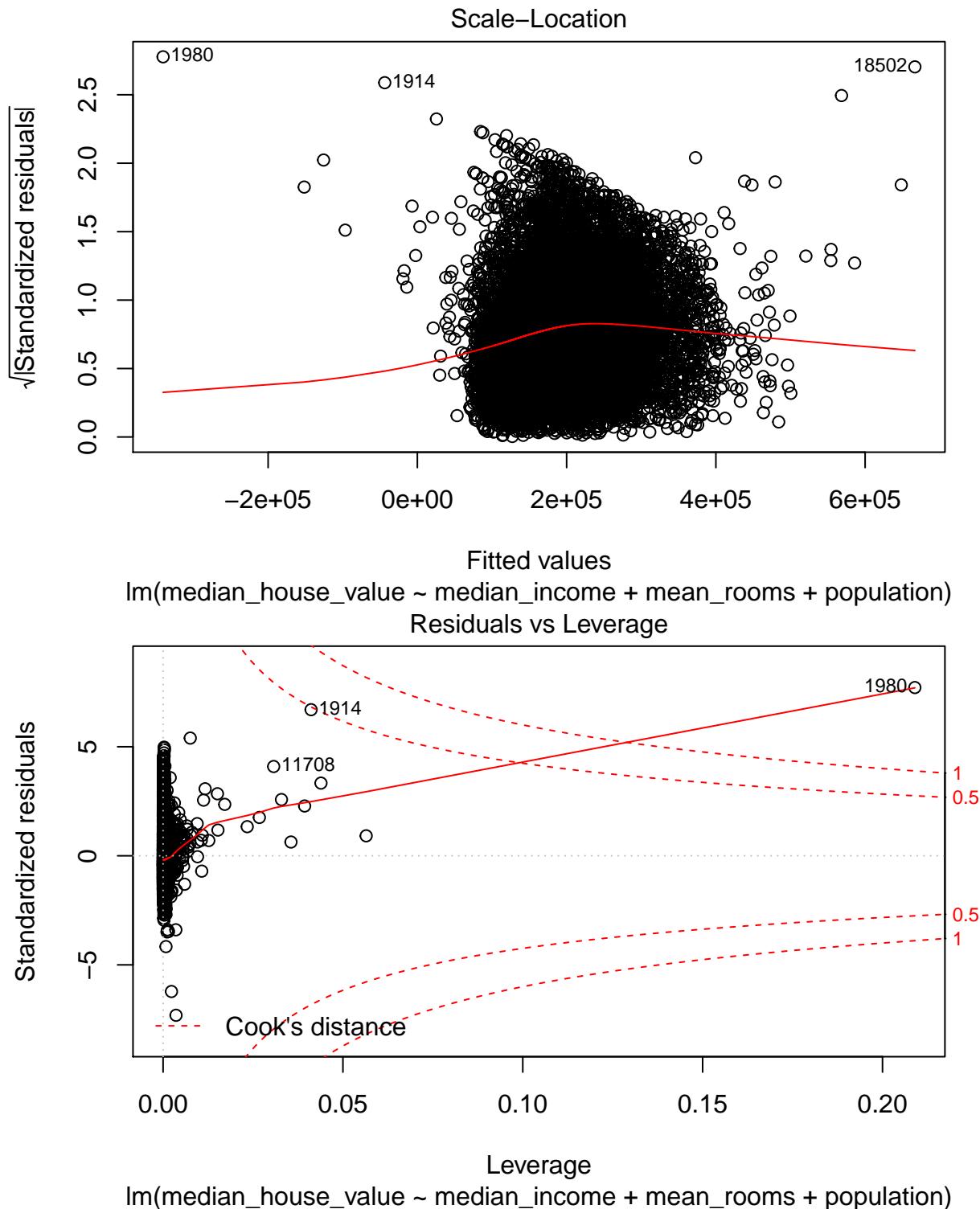
mean(predModel)

## [1] 192263.2
sd(predModel)

## [1] 64532.45
plot(lineModel)

```





## Modelo 2.

El segundo modelo utiliza los predictores: **median\_income**, **mean\_rooms**, **population** y **housing\_median\_age** para intentar predecir el valor de las propiedades plasmados en **median\_house\_value**.

```

lineModel2 <- lm( median_house_value~median_income+mean_rooms+population+housing_median_age, data=dftrain)
summary(lineModel2)

##
## Call:
## lm(formula = median_house_value ~ median_income + mean_rooms +
##     population + housing_median_age, data = dftrain)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -586838 -48977 -12660   35151  432713
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 192466.4    567.2 339.30 < 2e-16 ***
## median_income 69892.3    606.3 115.28 < 2e-16 ***
## mean_rooms    -7618.7    596.9 -12.76 < 2e-16 ***
## population     3044.8    592.4   5.14 2.78e-07 ***
## housing_median_age 18921.0    610.2  31.01 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71230 on 15762 degrees of freedom
## Multiple R-squared:  0.4687, Adjusted R-squared:  0.4685
## F-statistic:  3476 on 4 and 15762 DF,  p-value: < 2.2e-16
vif(lineModel2)

##
##          median_income          mean_rooms          population
##             1.139868            1.147205            1.112583
## housing_median_age
##                 1.163611

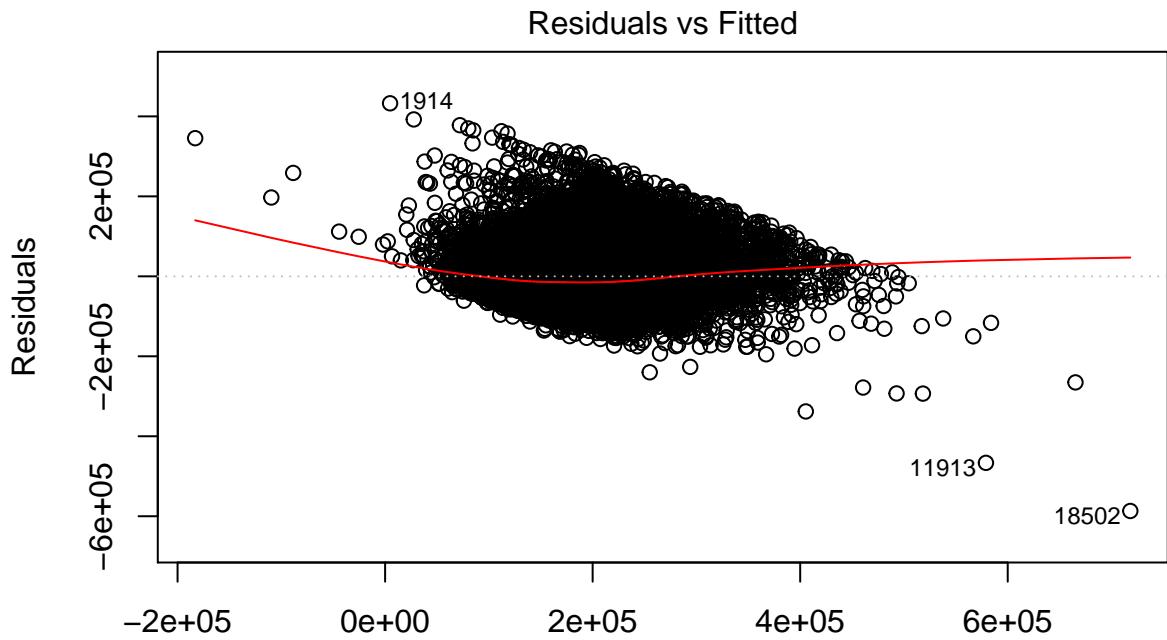
```

El coeficiente **R-squared** de 0.5155 del modelo nos dice que el modelo explica el 51.55% de la varianza total de la variable en la regresión, supera al modelo anterior en casi 3%.

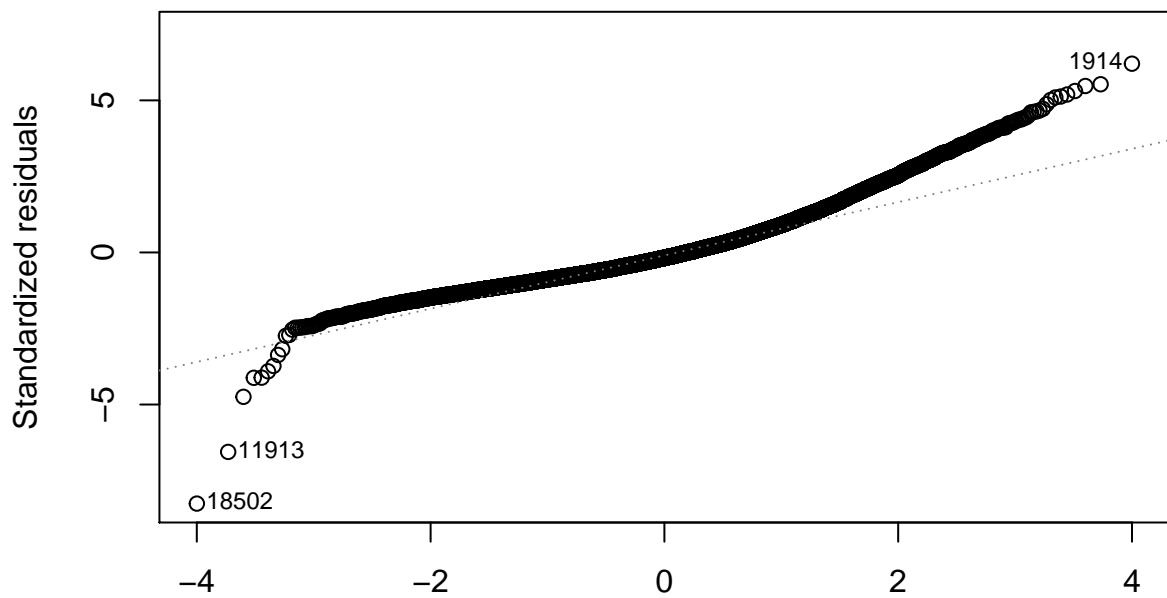
El coeficiente **Adjusted R-squared** nos dice que el 51.54% de la variable dependiente (median\_house\_value) es explicado por las variables independientes, también es superior al modelo anterior.

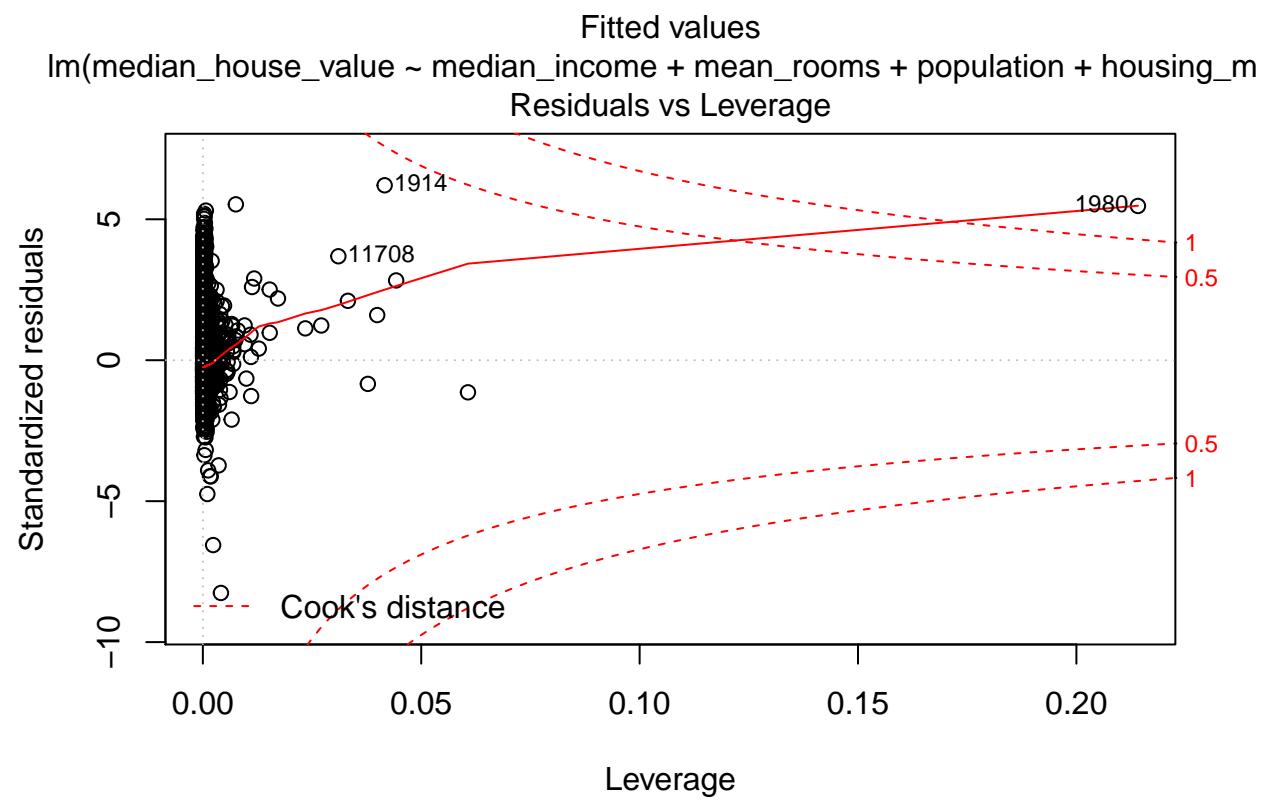
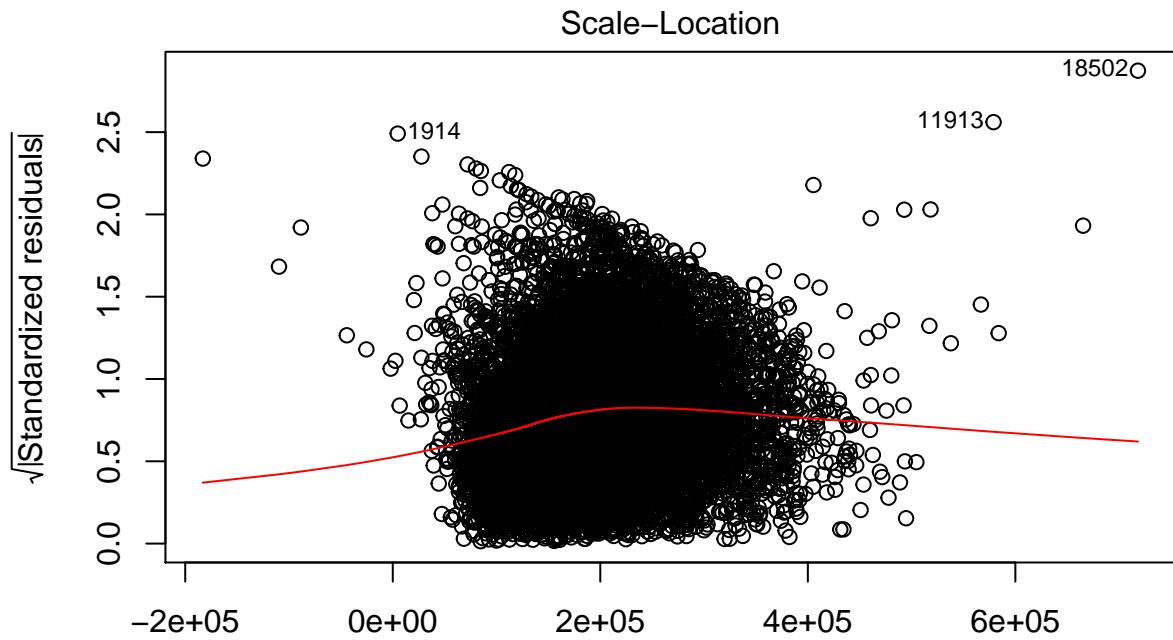
Los factores de inflación de la varianza tomados tienen valores aceptables ya que son menores a 10.

```
plot(lineModel2)
```



Normal Q-Q





```
predModel2 <- predict(lineModel2, newdata = dftrain)
summary(predModel2)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	-183025	144131	183991	192263	232235	718138

```
mean(predModel2)
```

```
## [1] 192263.2
```

```
sd(predModel2)
```

```
## [1] 66886.83
```

### Modelo 3.

El último modelo utiliza los predictores: **mean\_bedrooms**, **mean\_rooms**, **median\_income** y **housing\_median\_age** para intentar predecir el valor de las propiedades plasmados en **median\_house\_value**.

```
lineModel3 <- lm(median_house_value ~ mean_bedrooms + mean_rooms + housing_median_age + median_income + ocean_proximity, data = dftrain)
summary(lineModel3)
```

```
##
```

```
## Call:
```

```
## lm(formula = median_house_value ~ mean_bedrooms + mean_rooms +
##      housing_median_age + median_income + ocean_proximity, data = dftrain)
```

```
##
```

```
## Residuals:
```

```
##      Min      1Q  Median      3Q     Max
## -526817 -40905 -10254  28656 372066
```

```
##
```

```
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t )
## (Intercept)	210826.1	793.3	265.765	< 2e-16 ***
## mean_bedrooms	6223.9	879.5	7.077	1.53e-12 ***
## mean_rooms	-5934.0	980.3	-6.054	1.45e-09 ***
## housing_median_age	9016.8	553.0	16.306	< 2e-16 ***
## median_income	60565.4	671.2	90.235	< 2e-16 ***
## ocean_proximityINLAND	-65262.1	1302.4	-50.108	< 2e-16 ***
## ocean_proximityISLAND	184647.9	32029.5	5.765	8.32e-09 ***
## ocean_proximityNEAR BAY	10359.8	1784.1	5.807	6.49e-09 ***
## ocean_proximityNEAR OCEAN	16654.2	1639.5	10.158	< 2e-16 ***

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard error: 64020 on 15758 degrees of freedom
```

```
## Multiple R-squared:  0.5708, Adjusted R-squared:  0.5706
```

```
## F-statistic: 2620 on 8 and 15758 DF, p-value: < 2.2e-16
```

```
vif(lineModel3)
```

	mean_bedrooms	mean_rooms
##	3.239517	3.829250
##	housing_median_age	median_income
##	1.182851	1.729180
##	ocean_proximityINLAND	ocean_proximityISLAND
##	1.443591	1.000902
##	ocean_proximityNEAR BAY	ocean_proximityNEAR OCEAN
##	1.155797	1.131427

Los coeficientes **R-squared** y **Adjusted R-squared** son los más altos de los 3 modelos y los factores de inflación de la varianza tienen valores aceptables ya que son menores a 10. Finalmente el error residual

standard es el menor de los 3 modelos.

```
predModel3 <- predict(lineModel3, newdata = dftrain)
summary(predModel3)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 15426 138997 191153 192263 240973 720278
```

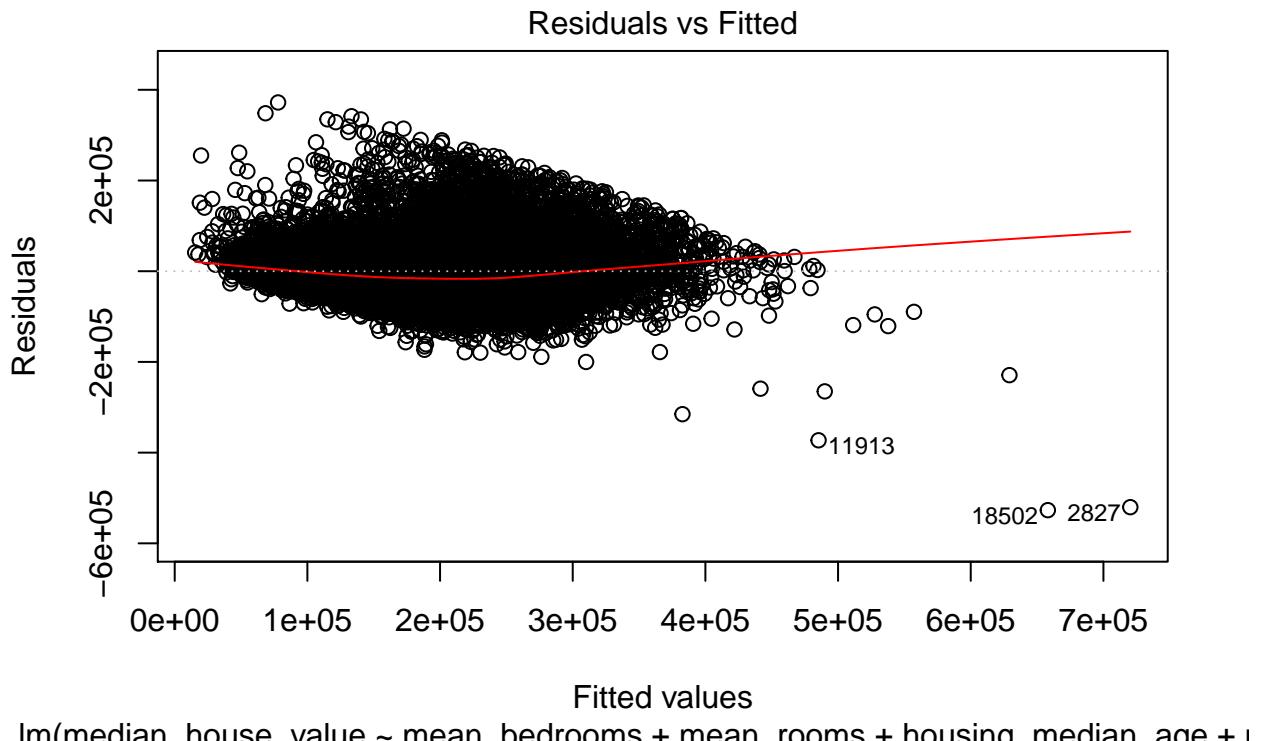
```
mean(predModel3)
```

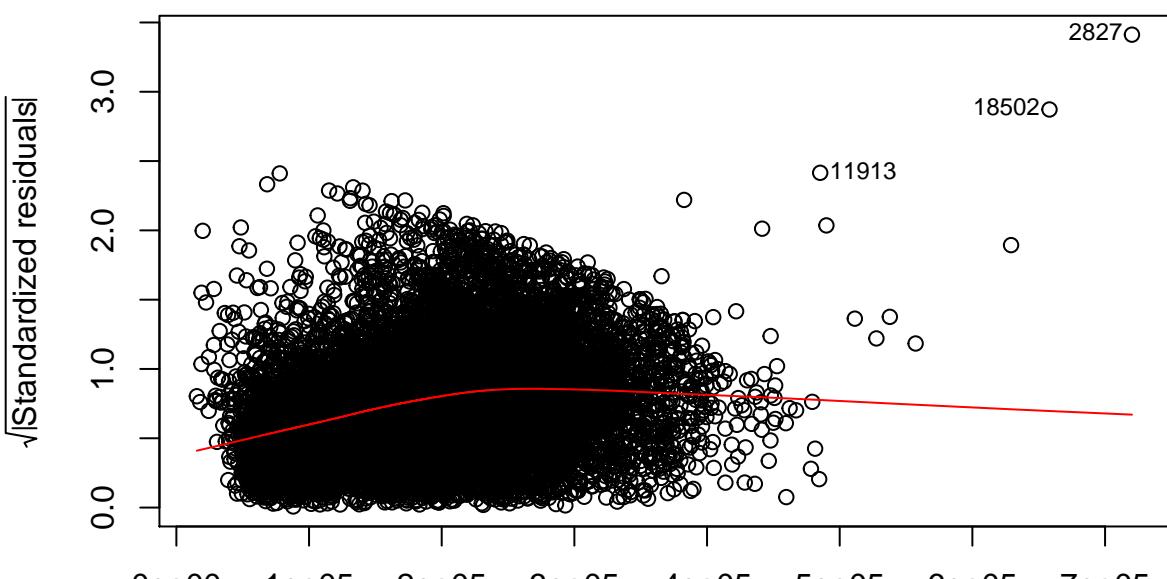
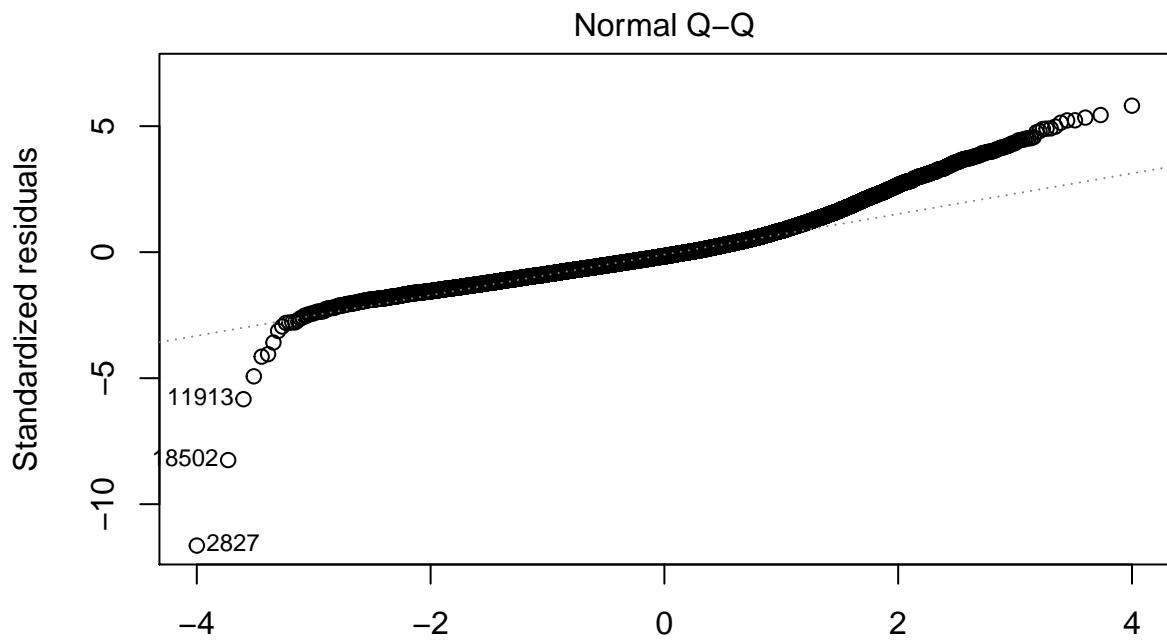
```
## [1] 192263.2
```

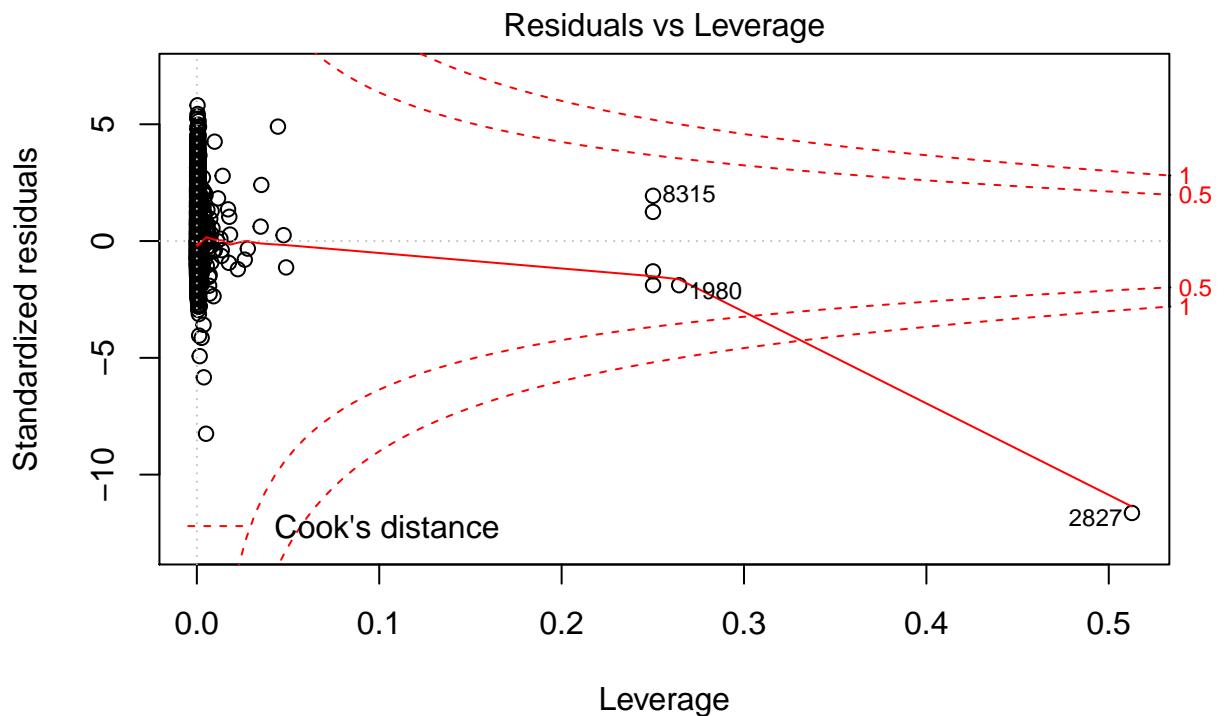
```
sd(predModel3)
```

```
## [1] 73816.73
```

```
plot(lineModel3)
```







lm(median\_house\_value ~ mean\_bedrooms + mean\_rooms + housing\_median\_age + i)

# Validación Se realiza la validación de los modelos con los datos de prueba. TODO: TENEMOS QUE PROBAR Y VALIDAR LOS DATOS. PARA ESTO HACEMOS UNA PREDICCIÓN CON TODO LOS DATOS Y LOS PREDECIDOS CON TEST Y COMPARAMOS. LUEGO HACEMOS EL SUM Y COMPARAMOS. UNA SEGUNDA OPCIÓN ES TOMAR 15 CASAS AL AZAR.

```

predTestModelo1 <- predict(lineModel, newdata=dftest)

dftest <- cbind(dftest, predSales=predTestModelo1)

mean(dftest$median_house_value)

## [1] 191210.8
sd(dftest$median_house_value)

## [1] 94679.68

```

## Conclusiones

Consideramos que la mejor opción es el modelo N° 3 debido a que los coeficientes  $R^2$  y  $R^2$  ajustado son los más altos de los tres modelos, además el error standard residual es el más bajo de los 3, sin embargo vimos que