Práctica 2: Limpieza y análisis de datos

Autor: César Aguilera y Daniel Velasco

Diciembre 2020

Introducción

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

0. Carga del archivo

Se abre el archivo de datos y se examina el tipo de datos con los que R ha interpretado cada variable. Examinaremos también los valores resumen de cada tipo de variable.

```
houses = read.csv(file = './data/housing.csv')
```

```
head(houses)
```

	longitude <dbl>	latitude <dbl>	housing_median_age <dbl>	total_rooms <dbl>	total_bedrooms <dbl>	population <dbl>	hous
1	-122.23	37.88	41	880	129	322	
2	-122.22	37.86	21	7099	1106	2401	
3	-122.24	37.85	52	1467	190	496	
4	-122.25	37.85	52	1274	235	558	
5	-122.25	37.85	52	1627	280	565	
6	-122.25	37.85	52	919	213	413	

6 rows | 1-8 of 11 columns

0.1 Atributos / Nombres de columna

```
names(houses)
```

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

0.2 Dimensiones

```
dims = dim(houses)
dims
```

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

```
print(paste("Filas: ", dims[1]))
```

```
## [1] "Filas:  20640"
```

```
print(paste("Columnas: ", dims[2]))
```

```
## [1] "Columnas:  10"
```

0.3 Tipo de datos con los que R ha interpretado cada variable

```
sapply(houses,class)
```

```
##          longitude           latitude housing_median_age        total_rooms
##          "numeric"          "numeric"          "numeric"          "numeric"
##     total_bedrooms         population         households      median_income
##          "numeric"          "numeric"          "numeric"          "numeric"
## median_house_value    ocean_proximity
##          "numeric"           "factor"
```

0.4 Comprobar si hay valores perdidos

```
any(is.na(houses))
```

```
## [1] TRUE
```

0.5 Resumen de cada tipo de variable

```
summary(houses)
```

```
##    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
##
```

1. Descripción del dataset

¿Por qué es importante y qué pregunta/problema pretende responder?

Este conjunto de datos se utiliza en el segundo capítulo del libro de Aurélien Géron *'Hands-On Machine learning with Scikit-Learn and TensorFlow'*. Sirve como una excelente introducción a la implementación de algoritmos de Machine Learning porque requiere una limpieza de datos preliminar, tiene una lista de variables fácilmente comprensible y tiene un tamaño óptimo: no es demasiado de juguete y ni demasiado difícil.

Los datos contienen información sobre el censo de California de 1990. Aunque puede que no nos ayuden a predecir los precios actuales de la vivienda como el conjunto de datos Zillow Zestimate (https://www.kaggle.com/c/zillow-prize-1 (https://www.kaggle.com/c/zillow-prize-1)), si que proporciona un conjunto de datos introductorio y accesible para aprender los conceptos básicos del aprendizaje automático.

El dataset contiene datos refrentes a casas pertenecientes a distrito determinado de California y algunas estadísticas resumidas sobre ellas basadas en los datos del censo de 1990. Debemos tener en cuenta que los datos están limpios, es decir, requieren limpieza previa.

El dataset tiene 20640 filas y 10 columnas. Las columnas son las siguientes:

- **longitude**: una medida de qué tan al oeste está una casa; un valor más alto está más al oeste
- **latitude**: medida de la distancia al norte de una casa; un valor más alto está más al norte
- **housing_median_age**: edad promedio de una casa dentro de un bloque; un número menor es un edificio más nuevo
- **total_rooms**: número total de habitaciones dentro de un bloque
- **total_bedrooms**: número total de dormitorios dentro de un bloque
- **population**: número total de personas que residen dentro de un bloque
- **households**: número total de hogares, un grupo de personas que residen dentro de una unidad de vivienda, para un bloque

- **median_income**: ingresos medios para hogares dentro de un bloque de casas (medidos en decenas de miles de dólares estadounidenses)
- **median_house_value**: valor medio de la vivienda para los hogares dentro de un bloque (medido en dólares estadounidenses)
- **oceanProximity**: ubicación de la casa con respecto al océano / mar

Fuente: https://www.kaggle.com/camnugent/california-housing-prices (https://www.kaggle.com/camnugent/california-housing-prices)

2. Integración y selección de los datos de interés a analizar

La **integración o fusión** de los datos consiste en la combinación de datos procedentes de múltiples fuentes, con el fin de crear una estructura de datos coherente y única que contenga mayor cantidad de información.

Esa fusión puede hacerse de dos formas:

1. De forma horizontal, añadiendo nuevos atributos a la base de datos original
2. De forma vertical, incluyendo nuevos registros a la base de datos original

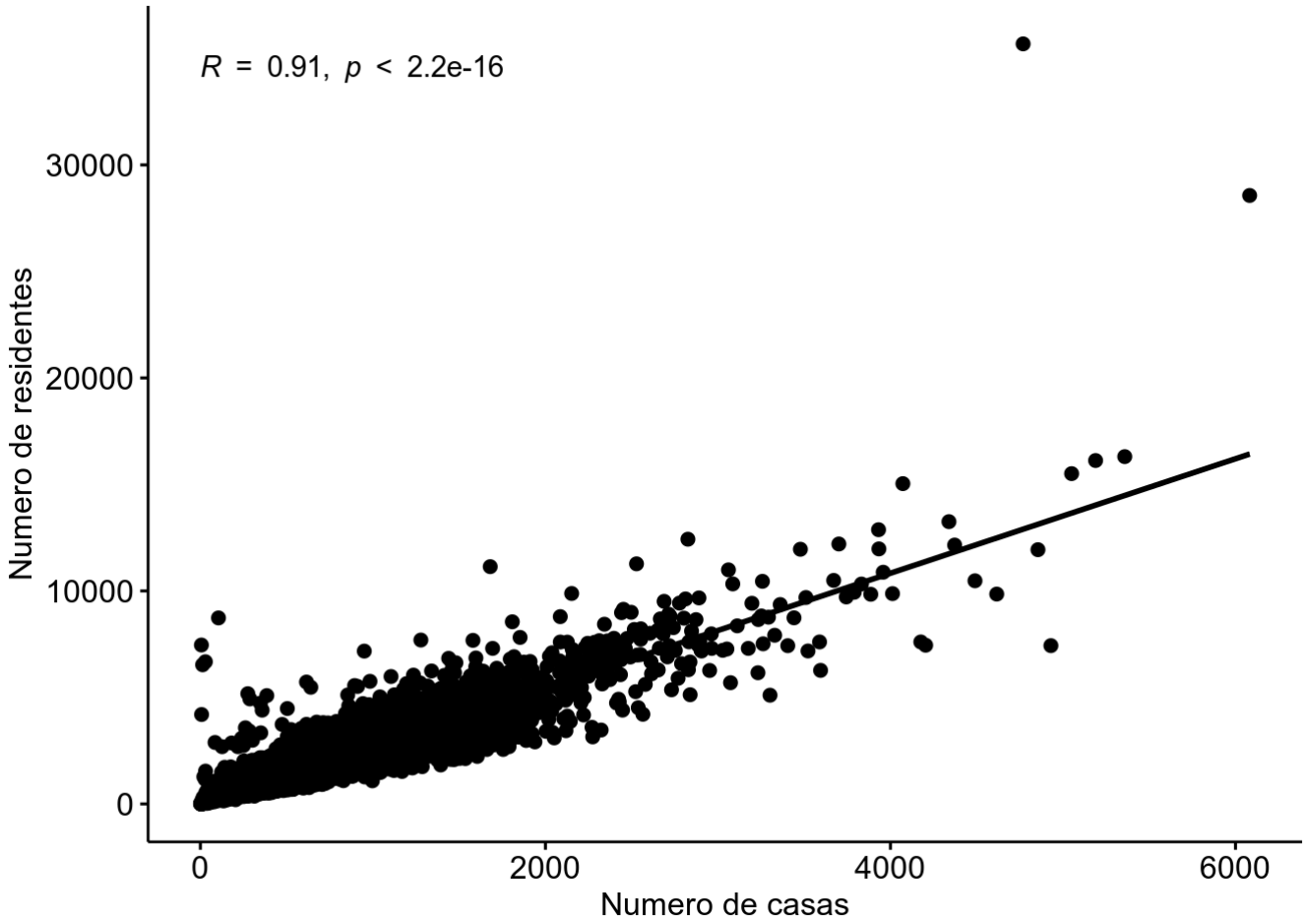
=> En este ejercicio NO vamos a incluir ningún tipo de integración de datos

La **selección** de datos consiste en la elección de aquellos registros y variables de interés o relevantes para el problema a resolver.

En este paso intentaremos seleccionar solo las caracteristicas de cada muestra que creamos aportan valor en la busqueda de nuestro objetivo, predecir el precio de las casas en base a sus características. En nuestro caso todos los atributos pueden ser muy utiles a primera vista ya que pensamos que el precio de una casa vendra determinado por su localizacion (latitud, longitud, proximidad al oceano), su tamaño (numero de habitaciones, dormitorios, numero de casas por bloque), su edad… pero de entre todos ellos podriamos eliminar *"population"* ya que el numero de personas en media que vivira en un bloque de casas sera directamente proporcional al tamaño y por tanto se podria eliminar. Podemos hacer un analisis rapido de la correlacion mediante el analisis del coeficiente de Pearson para demostrarlo.

```
ggscatter(houses, x = "households", y = "population",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Numero de casas", ylab = "Numero de residentes")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Por tanto eliminamos la columna de *"population"*.

Ademas debido a su mas que probable correlacion con la columna *ocean_proximity*, decidimos eliminar la longitud y latitud donde se localiza las casas.

```
houses <- houses[ , -which(names(houses) %in% c("longitude","latitude","population"))]
head(houses)
```

total_rooms <dbl>	total_bedrooms <dbl>	households <dbl>	median_income <dbl>	median_house_value <dbl>	ocean_pr <fctr>
880	129	126	8.3252	452600	NEAR BA
7099	1106	1138	8.3014	358500	NEAR BA
1467	190	177	7.2574	352100	NEAR BA
1274	235	219	5.6431	341300	NEAR BA
1627	280	259	3.8462	342200	NEAR BA
919	213	193	4.0368	269700	NEAR BA

6 rows | 3-8 of 8 columns

3. Limpieza de los datos

3.1.1 ¿Los datos contienen ceros o elementos vacíos?

En este apartado buscamos los valores del dataset nulos y vacios. Para un correcto procesado a continuacion habrá que realizar un analisis de que valores son nulos y como se pueden tratar, llegado el caso se podrian inputar valores de la media o mediana, asi como eliminar las muestras con valores nulos.

```
# Obtenemos la cantidad de valores NA dentro de nuestro dataset por cada una de las columnas
sapply(houses, function(x) sum(is.na(x)))
```

```
## housing_median_age        total_rooms     total_bedrooms         households
##                  0                  0                207                  0
##      median_income median_house_value    ocean_proximity
##                  0                  0                  0
```

Podemos apreciar que hay **207** valores nulos en la columna *total_bedrooms*.

3.1.2 ¿Cómo gestionarías cada uno de estos casos?

Como hemos visto en el apartado anterior nos encontramos con 207 valores nulos en *total_bedrooms* pero ya que el resto de columnas no tienen valores NA decidimos imputar valores aproximados dependiendo del valor total de habitaciones ya que pensamos que el numero de dormitorios estara directamente relacionado con el total de habitaciones.

Para ello primero identificamos las posiciones de los valores NA dentro de nuestro dataset.

```
#obtenemos los indices de las muestras donde total_bedrooms es NA
nanIndexes= which(is.na(houses$total_bedrooms))
nanIndexes
```

```
##   [1]   291   342   539   564   697   739  1098  1351  1457  1494  1607  2029
##  [13]  2116  2302  2324  2335  2352  2413  2421  2579  2609  2648  2827  3025
##  [25]  3329  3355  3377  3483  3486  3530  3722  3779  3913  3922  3959  4044
##  [37]  4047  4187  4280  4310  4392  4448  4497  4592  4601  4630  4668  4692
##  [49]  4739  4744  4745  4768  4853  5060  5217  5223  5237  5655  5666  5679
##  [61]  5724  5752  5991  6053  6069  6221  6242  6254  6299  6422  6542  6591
##  [73]  6815  6836  6963  7098  7114  7169  7192  7229  7317  7331  7548  7655
##  [85]  7669  7764  7807  8338  8384  8531  8916  9150  9572  9621  9623  9815
##  [97]  9846  9878  9943  9971 10034 10217 10237 10386 10390 10429 10496 10762
## [109] 10886 10916 11097 11312 11352 11442 11450 11513 11742 12102 12415 12571
## [121] 12810 13016 13070 13312 13333 13337 13598 13657 13707 13926 13933 13934
## [133] 14016 14153 14174 14308 14332 14387 14463 14522 14642 14931 14971 14987
## [145] 15031 15061 15119 15138 15398 15480 15608 15664 15891 15976 16026 16039
## [157] 16105 16106 16331 16758 16880 16881 16886 17042 17199 17203 17640 17826
## [169] 17841 17924 17929 17974 18178 18247 18262 18333 18347 18467 18787 18874
## [181] 18915 19061 19072 19123 19151 19253 19333 19392 19403 19486 19560 19608
## [193] 19639 19767 19819 19834 19891 19933 19960 20047 20070 20126 20268 20269
## [205] 20373 20461 20485
```

Decidimos por tanto inputar valores medios de numero de dormitorios de las casas de tamaño similar en la ciudad. Para ello usamos el siguiente script en el que se divide el dataset en grupos de muestras que tienen el mismo numero de habitaciones totales que la muestra a ser inputada, y se obtiene la media del numero de dormitorios. Este valor es el que se inputa.

```
medians = c()
pos = 1

# Compute medians and store them in a vector
for (index in nanIndexes){
    tot_rooms = houses$total_rooms[index]

    # cut slice, with same zone and area
    slice = subset(houses, houses$total_rooms == tot_rooms)$total_bedrooms

    # Compute the median of the slice
    sliceMedian = median(slice, na.rm = TRUE)

    # Store median
    medians[pos] = sliceMedian
    pos = pos + 1
}

pos = 1

# Set the values from medians vector
for(index in nanIndexes){
    houses$total_bedrooms[index] = medians[pos]
    pos = pos + 1
}

#remove the NA values if any
houses <- na.omit(houses)
```

Comprobamos el numero de NA de nuevo en nuestro dataset y vemos que **ahora ya no hay valores NA**.

```
# Obtenemos la cantidad de valores NA dentro de nuestro dataset por cada una de las columnas
sapply(houses, function(x) sum(is.na(x)))
```

```
## housing_median_age        total_rooms     total_bedrooms         households
##                  0                  0                  0                  0
##      median_income median_house_value    ocean_proximity
##                  0                  0                  0
```

3.2. Identificación y tratamiento de valores extremos

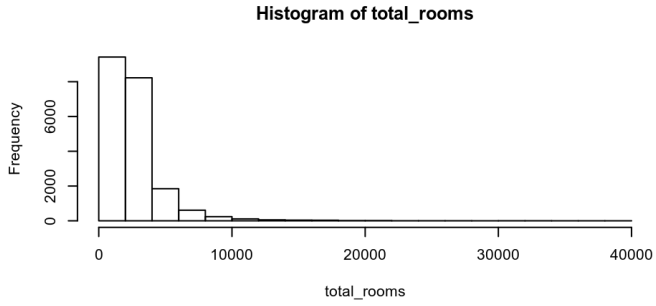
3.2.1 Identificación de valores extremos

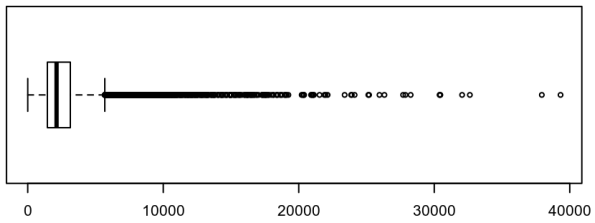
Ahora vamos a proceder al analisis de los valores extremos, este estudio permite identificar valores que debido a su lejania a la media estadistica se pueden considerar como no validos ya que ademas podrian afectar al analisis negativamente agregando una distorsion no deseable.

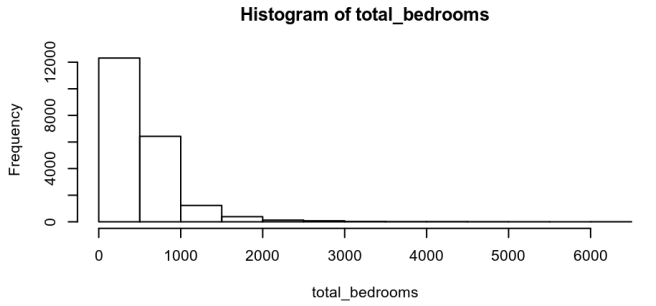
Para el analisis podemos usar los graficos de cajas donde se identifica la media el primer y tercer cuartil asi como el rango de hasta el 1.5*IQR (InterQuantile Range). Cualquier valor mas alejado de este rango se considerará outlier.

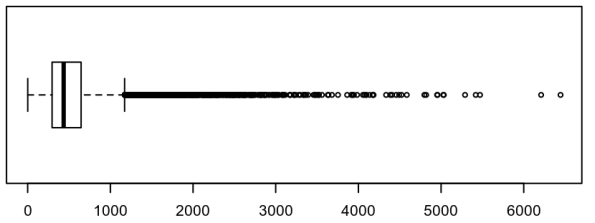
Podemos ver que las columnas analizadas a continuación tienen valores outliers (fuera del rango 3Q+1.5IQR) y debido a la cantidad de valores outliers estimamos que la mejor manera de solucionar el problema sería **modificando los valores extremos por el valor del 3º cuartil + 1.5 veces el IQR (interquartile range)**

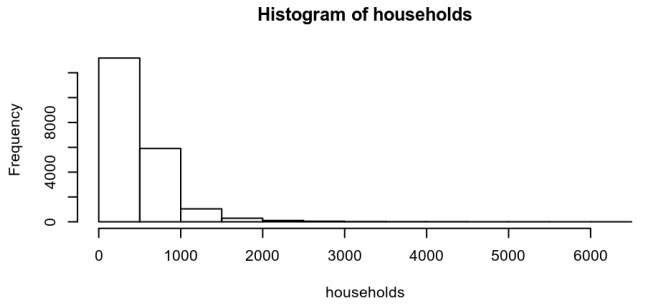
```
total_rooms <- houses$total_rooms
total_bedrooms <- houses$total_bedrooms
households <- houses$households
median_income <- houses$median_income
par(mfrow=c(4,2))
hist(total_rooms)
boxplot(total_rooms, horizontal=TRUE)
hist(total_bedrooms)
boxplot(total_bedrooms, horizontal=TRUE)
hist(households)
boxplot(households, horizontal=TRUE)
hist(median_income)
boxplot(median_income, horizontal=TRUE)
```

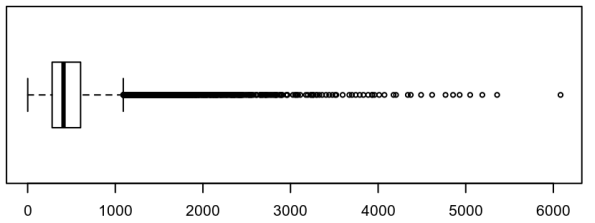


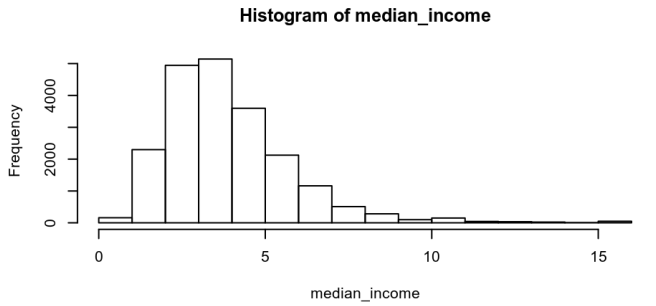


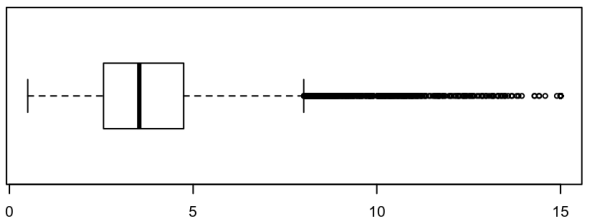












3.2.2 Tratamiento de valores extremos

Primeramente escribimos una funcion en la cual dada una columna de nuestro dataframe obtiene el valor máximo para considerar al valor como outlier (en base a su IQR y tercer cuartil) e inputa este valor maximo a todos los outliers de la columna.

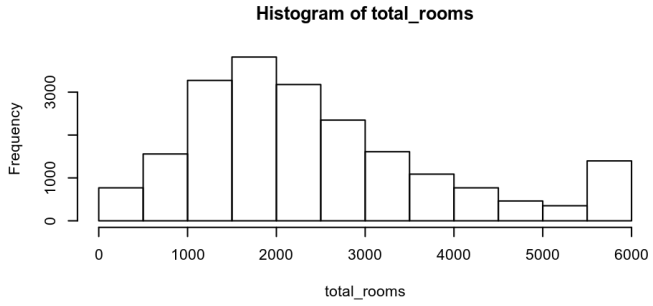
```
# reemplazo de outliers
outliersReplace <- function(dataColumn){
  summ_col = summary(dataColumn)
  iqr = summ_col["3rd Qu."]-summ_col["1st Qu."]
  highLimit = summ_col["3rd Qu."] + 1.5 * iqr
  dataColumn[dataColumn > highLimit] <- highLimit
  dataColumn     #devolvemos la columna
}
```

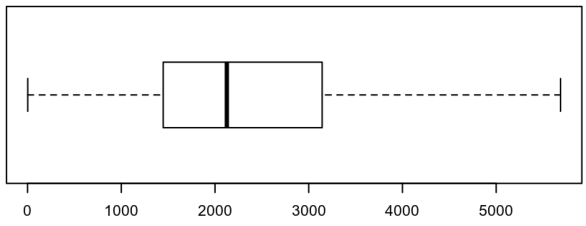
Aplicamos la funcion a todas las columnas con outliers.

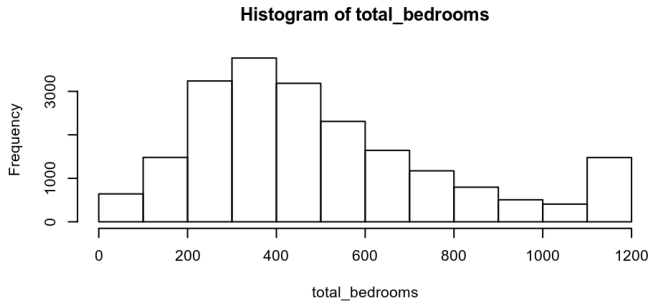
```
houses$total_rooms <- outliersReplace(houses$total_rooms)
houses$total_bedrooms <- outliersReplace(houses$total_bedrooms)
houses$households <- outliersReplace(houses$households)
houses$median_income <- outliersReplace(houses$median_income)
```

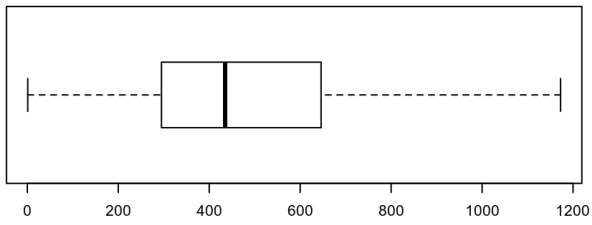
Vemos de nuevo los histogramas y los outliers en los diagramas de caja. Comprobando que ahora ya no existen outliers y que estos se han concentrado en el rango del Q3.

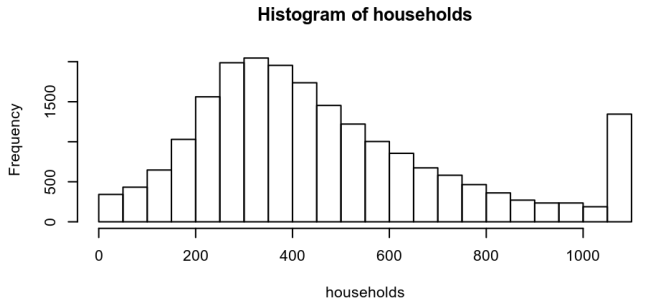
```
total_rooms <- houses$total_rooms
total_bedrooms <- houses$total_bedrooms
households <- houses$households
median_income <- houses$median_income
par(mfrow=c(4,2))
hist(total_rooms)
boxplot(total_rooms, horizontal=TRUE)
hist(total_bedrooms)
boxplot(total_bedrooms, horizontal=TRUE)
hist(households)
boxplot(households, horizontal=TRUE)
hist(median_income)
boxplot(median_income, horizontal=TRUE)
```

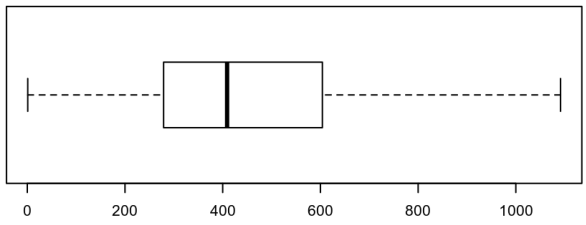


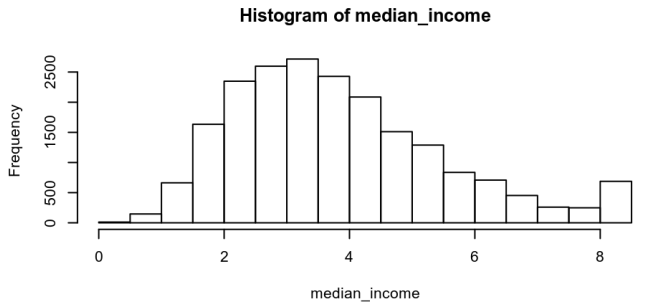


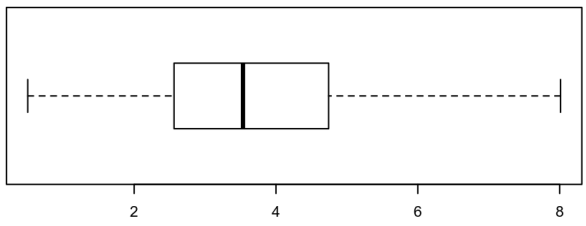












3.3. Exportacion de los datos preprocesados

Una vez realizado todo el preprocesado con los datos podemos guardarlos en este momento en un CSV.

```
# Exportamos los datos una vez estan libres de NA y sin outliers
write.csv(houses, "./data/housing_preprocessed.csv")
```

4. Análisis de los datos

4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar)

En esta sección hemos elegido distintos grupos de datos.

Primeramente, escogemos todas las variables para comprobar cuales siguen una distribución normal y cuales no.

Seguidamente seleccionamos la variable “numero de casas” (households) para estudiar la homogeneidad de su varianza dependiendo de si la población está cerca de la bahía o cerca del océano.

Finalmente elegimos las variables median_income y median_house_value para estudiar si existe o no correlación entre ambas.

4.2. Comprobación de la normalidad y homogeneidad de la varianza

4.2.0 Compración de la normalidad

Hipótesis

- H0: La muestra proviene de una distribución normal
- H1: La muestra no proviene de una distribución normal

Para pruebas de normalidad siempre se plantean así las hipótesis.

Nivel de Significancia

El nivel de significancia que se trabajará es de 0.05. Alfa=0.05

Criterio de Decisión

Si P < Alfa Se rechaza H0

Si p >= Alfa NO se rechaza H0

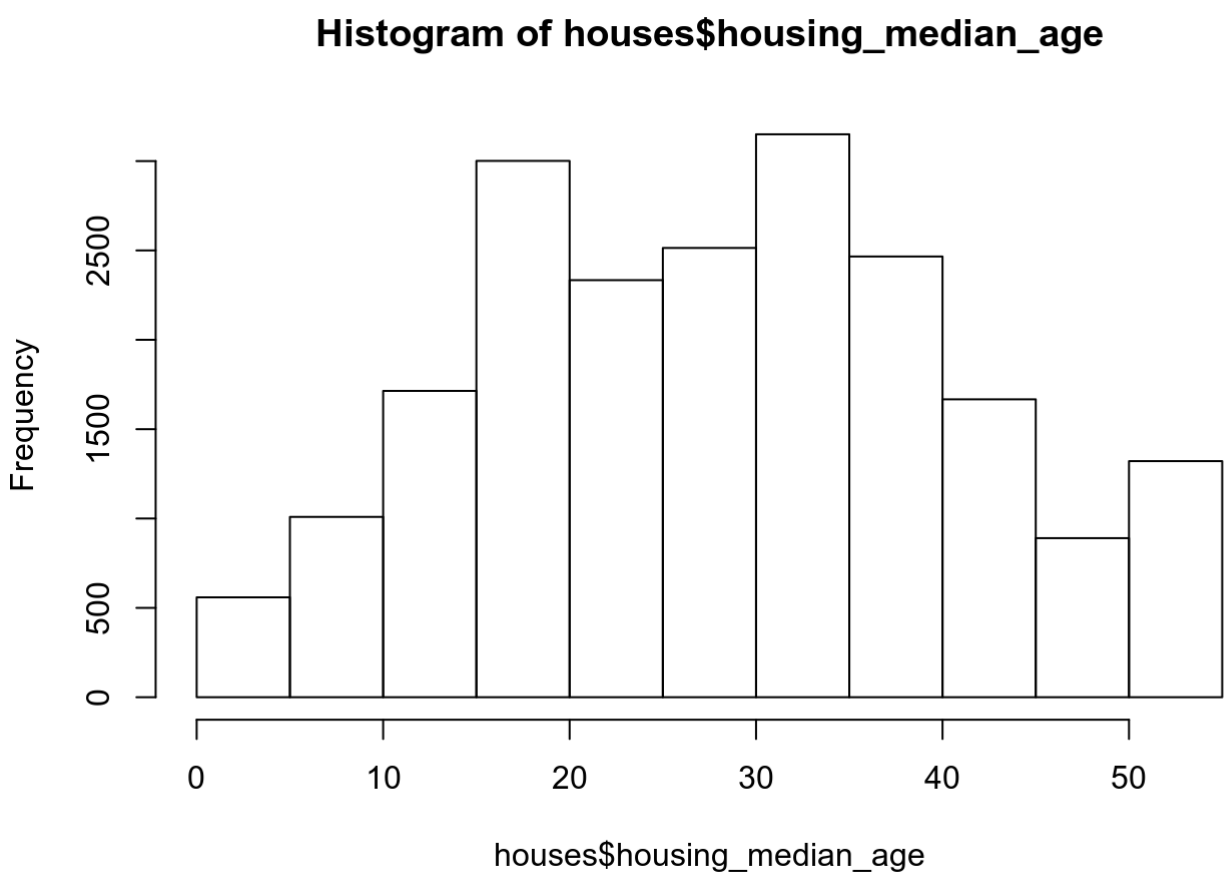
Donde P = p-valor

Test a aplicar

Vamos a aplicar el test de normalidad de Anderson-Darling, que funciona para variables con mas de 5000 muestras.

4.2.1 Comprobación de la normalidad de la variable housing_median_age

```
hist(houses$housing_median_age)
```



A primera vista el histograma no nos dice mucho si la variable sigue una distribución normal o no.

Apliquemos ahora el test de normalidad.

```
ad.test(houses$housing_median_age)
```

```
##
##  Anderson-Darling normality test
##
## data:  houses$housing_median_age
## A = 87.827, p-value < 2.2e-16
```

El p-valor es menor a Alpha (0.05), se rechaza la hipótesis nula. La variable housing_median_age NO sigue un distribución normal.

4.2.2 Comprobación de la normalidad para el resto de variables

En esta ocasión hemos creado un programa que comprueba la normalidad de todas las variables del conjunto de datos.

```
df = houses
alpha = 0.05

for (i in 2:ncol(df)){
    if(!is.numeric(df[,names(houses)[i]])){
      next
    }

    p = ad.test(df[,names(houses)[i]])$p.value

    if (p < alpha){
      print(paste(names(df)[i], "NO sigue una distribución normal"))
    }else if(p>=alpha){
      print(paste(names(df)[i], "SIGUE una distribución normal"))
    }
}
```

```
## [1] "total_rooms NO sigue una distribución normal"
## [1] "total_bedrooms NO sigue una distribución normal"
## [1] "households NO sigue una distribución normal"
## [1] "median_income NO sigue una distribución normal"
## [1] "median_house_value NO sigue una distribución normal"
```

Como vemos, el test de normalidad de Anderson-Darling da negativo para todas las variables, es decir, ninguna sigue una distribución normal.

4.2.3 Compración de la homogeneidad de la varianza

Hipótesis

- H0: La varianza es igual entre los grupos
- H1: La varianza NO es igual entre los grupos

Nivel de Significancia

El nivel de significancia que se trabajará es de 0.05. Alfa=0.05

Criterio de Decisión

Si P < Alfa Se rechaza H0

Si p >= Alfa NO se rechaza H0

Donde P = p-valor

Test a aplicar

Vamos a aplicar el test de Fligner-Killeen puesto que es uno de los más adecuados cuando no se cumple la condición de normalidad en las muestras.

4.2.4 Compración de la homogeneidad de la varianza de población entre casas cerca de la bahía y cerca del océano

```
a <- houses[houses$ocean_proximity =="NEAR BAY", "households"]
b <- houses[houses$ocean_proximity =="NEAR OCEAN", "households"]
fligner.test(x = list(a,b))
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  list(a, b)
## Fligner-Killeen:med chi-squared = 2.7541, df = 1, p-value = 0.097
```

El p-valor (0.097) es menor a Alpha (0.05), se confirma la hipótesis nula. Las varianzas son iguales entre los dos grupos (cerca del mar y lejos del mar).

4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Primeramente vamos a empezar calculando el intervalo de confianza para la media de la variable median_house_value. El intervalo de confianza permite calcular dos valores alrededor de una media muestral (uno superior y otro inferior). Estos dos valores van a acotar un rango dentro del cual, con una determinada probabilidad, se va a localizar el parámetro de la media poblacional.

**El intervalo de confianza calculado será por defecto del 95%.

```
t.test(houses$median_house_value)
```

```
##
##  One Sample t-test
##
## data:  houses$median_house_value
## t = 257.42, df = 20624, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  205256.9 208406.7
## sample estimates:
## mean of x
##  206831.8
```

El intervalo de confianza para la variables median_house_value nos indica que la probabilidad de que la media poblacional μ pertenezca a un intervalo de la forma: [205256.9, 208406.7] es de 0.95. O lo que es lo mismo: noventa y cinco de cada cien veces que escogemos una muestra aleatoria simple y calculamos el valor de la media muestral, el intervalo que obtendremos sustituyendo el valor de $\bar{X}$ por la media correspondiente a la muestra de la que disponemos contendrá el verdadero valor de μ.

Seguidamente vamos a utilizar el contraste de hipótesis para evaluar si el valor medio de la vivienda para los hogares dentro de un bloque (median_house_value) es superior en los bloques cerca de la bahía con respecto a los que NO están cerca de la bahía.

```
nearBay <- houses[houses$ocean_proximity == "NEAR BAY", "median_house_value"]
farBay <- houses[houses$ocean_proximity != "NEAR BAY", "median_house_value"]
```

Hipótesis nula

La hipótesis nula (H0) afirma que los valores de las medias de las dos poblaciones son iguales. Es decir, la media poblacional del valor medio de la vivienda para los hogares dentro de un bloque es igual en los bloques cerca de la bahía que los que NO están cerca de la bahía: $\mu 1 = \mu 2$.

Otra manera de ver la hipótesis nula es $\mu 1 - \mu 2 = 0$.

Hipótesis alternativa

La hipotesis alternativa (H1) afirma que la media de la población 1 es superior a la media de la población 2. Es decir, que la media el del valor medio de la vivienda para los hogares dentro de un bloque (median_house_value) es superior en los bloques cerca de la bahía con respecto a los que NO están cerca de la bahía.: $\mu 1 > \mu 2$

Otra manera de ver la hipótesis alternativa es $\mu 1 - \mu 2 > 0$.

Test a aplicar

Dado que no podemos asegurar que la variable median_house_value siga una distribuación normal, sólo podremos contrastar la diferencia de medias si los tamaños de las muestras son superiores a treinta => que en este caso se cumple.

En resumen, el test a aplicar es el contraste sobre la diferencia de medias en el caso de tener muestras grandes no normales.

Cálculos

Con un nivel de significación del 5%, ¿podemos asegurar que el valor medio de la vivienda es el mismo?

```
x1 = mean(nearBay)
s1 = sd(nearBay)
n1 = length(nearBay)

x2 = mean(farBay)
s2 = sd(farBay)
n2 = length(farBay)

alpha = 0.05

# estadístico de contraste
z = (x1-x2)/sqrt((s1*s1)/n1 + (s2*s2)/n2)

# p-valor
p = 1 - pnorm(z)

print(paste("p-value is", p))
```

```
## [1] "p-value is 0"
```

```
if(p < alpha){
  print("p less than alpha")
} else if (p == alpha){
  print("p equal to alpha")
} else {print("p greater than alpha")}
```

```
## [1] "p less than alpha"
```

El p-valor obtenido es cero. Y en concreto menor que el nivel de significación.

Diremos que el p-valor es significativo y rechazamos la hipótesis nula en favor de la hipótesis alternativa. Por ende se puede afirmar que el valor medio de la vivienda para los hogares dentro de un bloque (median_house_value) es superior en los bloques cerca de la bahía con respecto a los que NO están cerca de la bahía, y esto se afirma con un 95% de nivel de confianza.

Finalmente realizamos un analisis global de correlaciones entre las columnas numericas de nuestro dataset. Este análisis sera una manera muy gráfica y facil de entender las dependencias de nuestra variable objetivo (precio de las casas con respecto a las atributos predictores)

Para ello primeramente obtenemos el coeficiente de correlacion entre atributos usando Pearson's el cual nos dirá el grado de relacion lineal entre las columnas de nuestro dataset.

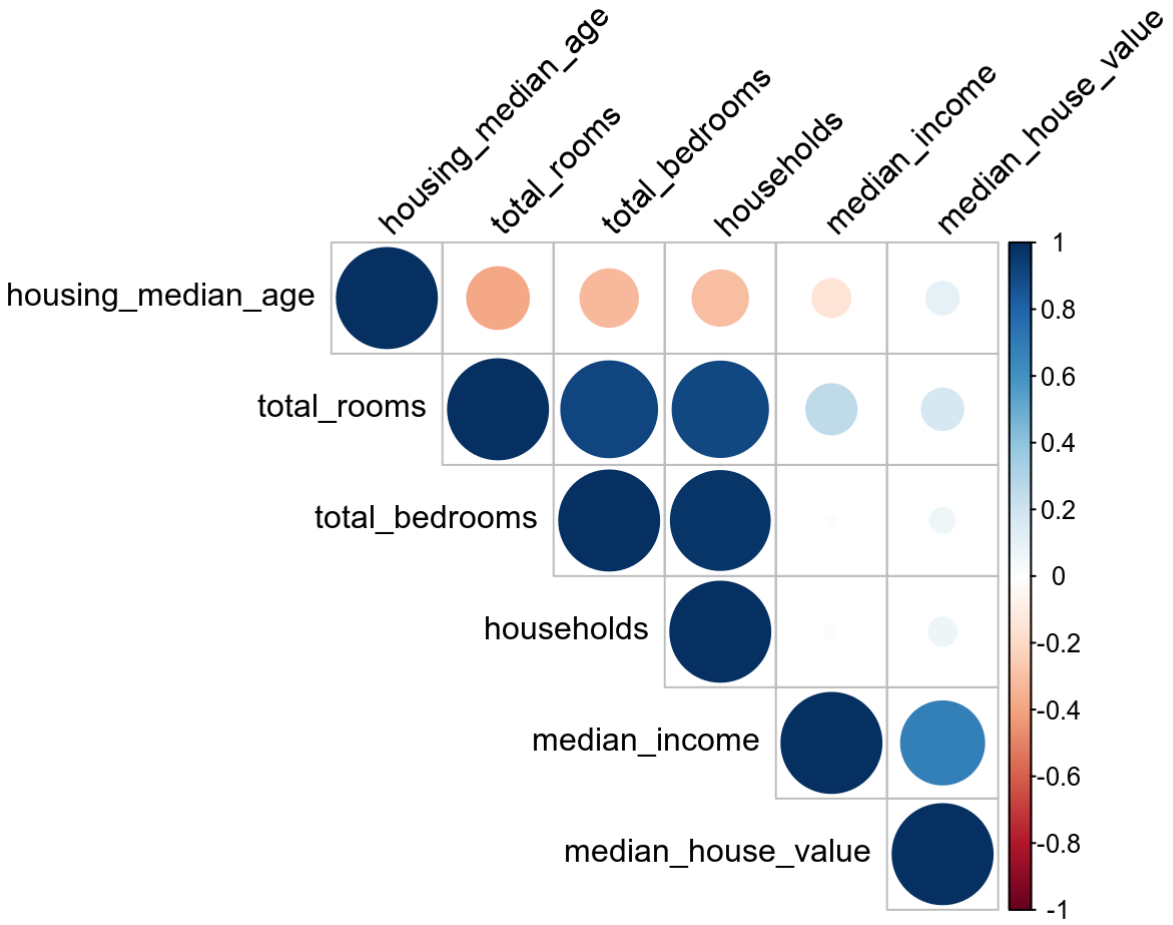
```
res <- cor(as.matrix(houses[ , -which(names(houses) %in% c("ocean_proximity"))]))
res
```

```
##                     housing_median_age total_rooms total_bedrooms  households
## housing_median_age          1.0000000  -0.3810978    -0.32985158 -0.30676066
## total_rooms                -0.3810978   1.0000000     0.91811048  0.90930412
## total_bedrooms             -0.3298516   0.9181105     1.00000000  0.97703995
## households                 -0.3067607   0.9093041     0.97703995  1.00000000
## median_income              -0.1421722   0.2517503    -0.01048092  0.01732516
## median_house_value          0.1058603   0.1718334     0.06165455  0.07856971
##                     median_income median_house_value
## housing_median_age   -0.14217224         0.10586033
## total_rooms           0.25175027         0.17183344
## total_bedrooms       -0.01048092         0.06165455
## households            0.01732516         0.07856971
## median_income         1.00000000         0.68902787
## median_house_value    0.68902787         1.00000000
```

Una vez tenemos los coeficientes de correlacion pasamos a representar la matriz de correlación. En la cual vemos lo siguiente:

- *total_bedrooms*, *total_rooms* y *households* están fuertemente correlacionados, por lo que podriamos reducir complejidad de nuestro dataset eliminando dos de ellas sin perder calidad, de igual manera que con el atributo *population* eliminado al inicio de esta practica.
- Tambien podemos ver como nuestra variable objetivo tiene una correlacion alta con *median_income* es decir con el salario de los habitantes de las casas en cuestion.
- Como curiosidad se podria destacar tambien la correlacion inversa que existe entre la edad de las casas y el tamaño de las casas (total habitaciones, dormitorios etc), lo cual se podria explicar con que casas mas antiguas eran mas pequeñas o incluso unifamiliares.

```
corrplot(res, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```



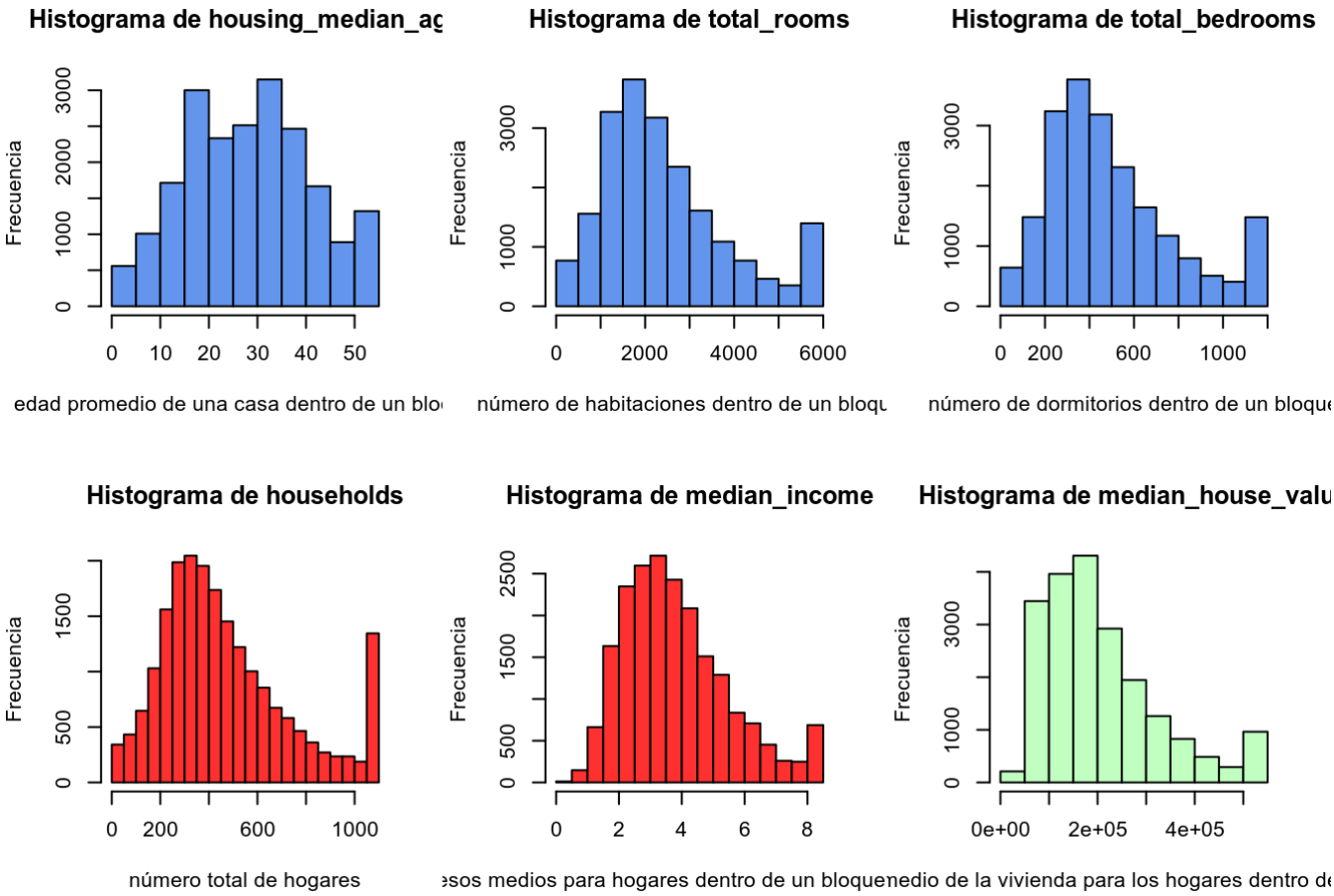
5. Representación de los resultados a partir de tablas y gráficas

5.1 Descriptiva y visualización

A continuación vamos a realizar una visualización gráfica de los datos del conjunto de datos. También explicaremos brevemente los gráficos y lo que se puede observar a partir de ellos.

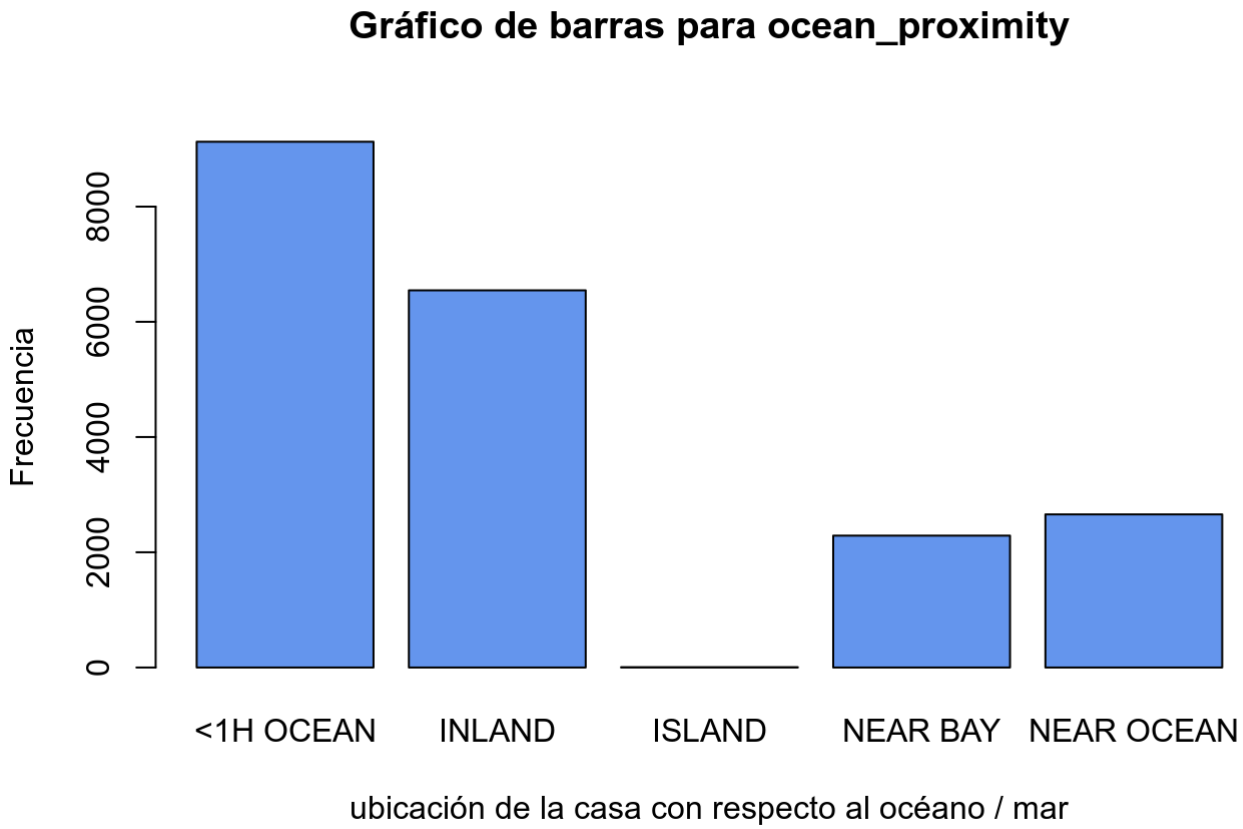
5.1.1 Histogramas

```
par(mfrow=c(2,3))

hist(houses$housing_median_age,
main="Histograma de housing_median_age",
xlab="edad promedio de una casa dentro de un bloque",
ylab="Frecuencia",
col="cornflowerblue",
)


hist(houses$total_rooms,
main="Histograma de total_rooms",
xlab="número de habitaciones dentro de un bloque",
ylab="Frecuencia",
col="cornflowerblue",
)

hist(houses$total_bedrooms,
main="Histograma de total_bedrooms",
xlab="número de dormitorios dentro de un bloque",
ylab="Frecuencia",
col="cornflowerblue",
)

######

hist(houses$households,
main="Histograma de households",
xlab="número total de hogares",
ylab="Frecuencia",
col="firebrick1",
)

hist(houses$median_income,
main="Histograma de median_income",
xlab="ingresos medios para hogares dentro de un bloque de casas",
ylab="Frecuencia",
col="firebrick1",
)

####

hist(houses$median_house_value,
main="Histograma de median_house_value",
xlab="valor medio de la vivienda para los hogares dentro de un bloque",
ylab="Frecuencia",
col="darkseagreen1",
)
```



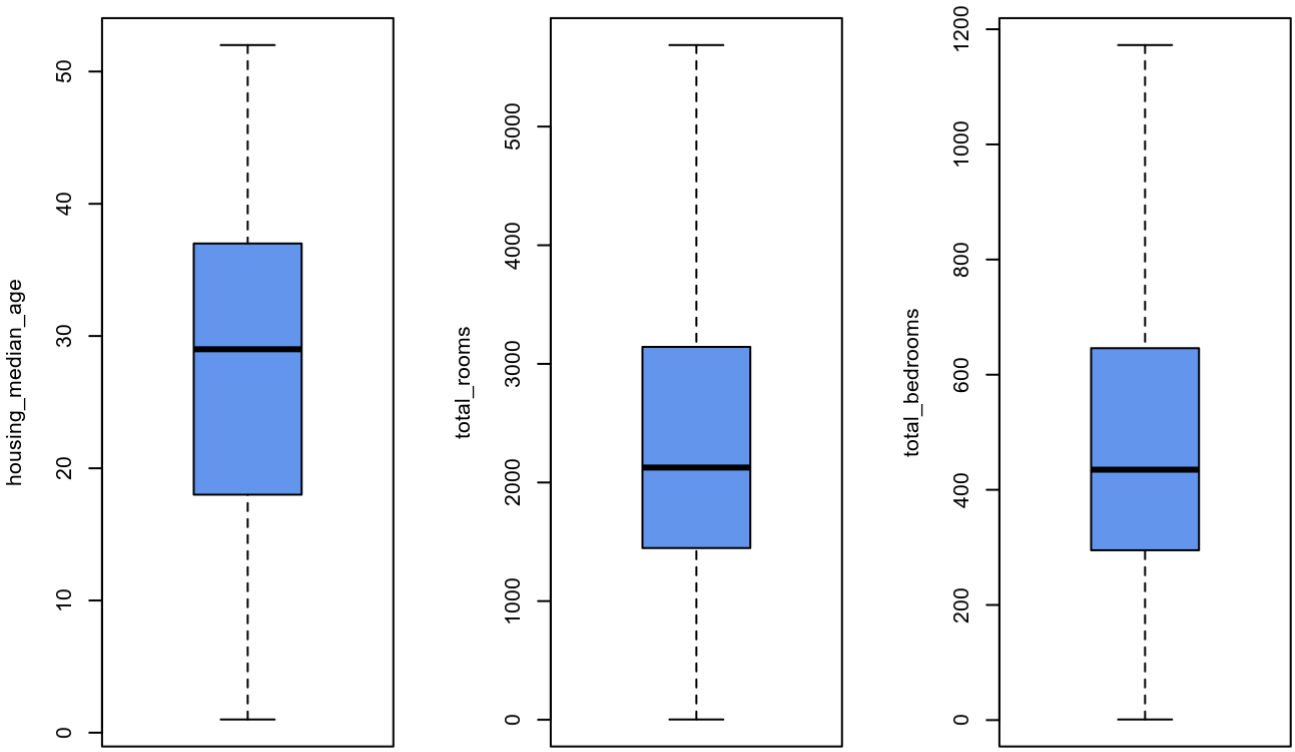
5.1.2 Gráficos de barras

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

Barplot(houses$ocean_proximity, main= "Gráfico de barras para ocean_proximity", xlab="ubicaci
ón de la casa con respecto al océano / mar", ylab="Frecuencia", col="cornflowerblue")
```



5.1.3 Diagramas de caja

```
par(mfrow=c(1,3))

boxplot(houses$housing_median_age, ylab = "housing_median_age", col = "cornflowerblue")
boxplot(houses$total_rooms, ylab = "total_rooms", col = "cornflowerblue")
boxplot(houses$total_bedrooms, ylab = "total_bedrooms", col = "cornflowerblue")
```

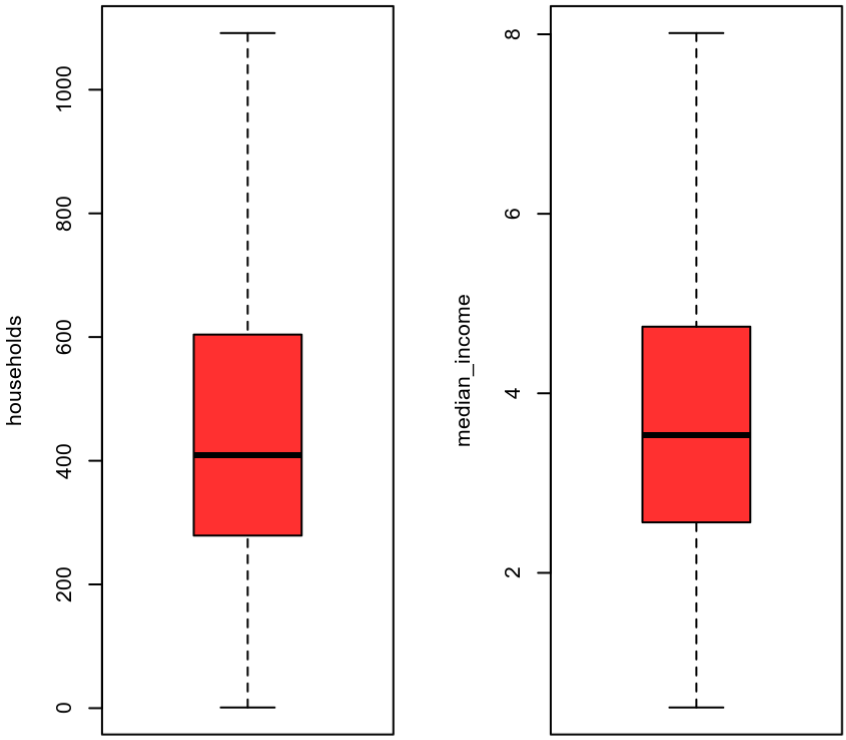


En el gráfico de **housing_median_age** se observa una mediana aproximada de 29, un mínimo de 1, un máximo de 52 y ningún valor atípico.

En el gráfico de **total_rooms** se observa una mediana aproximada de 2127, un mínimo de 2, un máximo de 5694 y numerosos valores atípicos entre 4883 y 5694.

En el gráfico de **total_bedrooms** se observa una mediana de 435, un mínimo de 1, un máximo de 1163 y numerosos valores atípicos entre 993 y 1163.

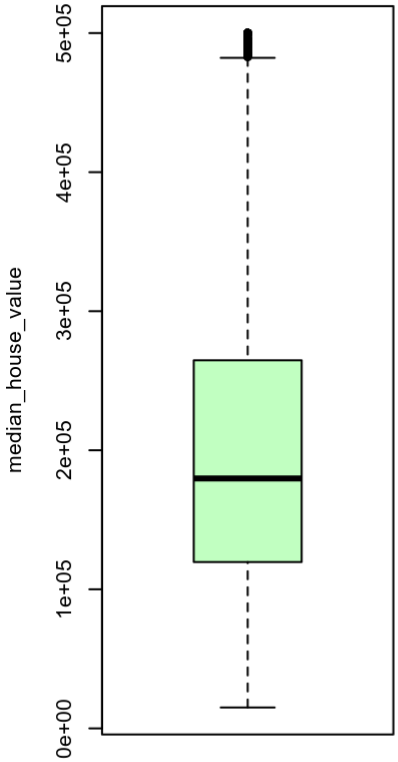
```
par(mfrow=c(1,3))
boxplot(houses$households, ylab = "households", col = "firebrick1")
boxplot(houses$median_income, ylab = "median_income", col = "firebrick1")
```



El gráfico de **households** presenta numerosos valores atípicos en el rango [941, 1092]. La mediana es 409, el mínimo 1 y el máximo 1092.

En el gráfico de **median_income** se observa una mediana aproximada de 3.53, un mínimo de 0.5, un máximo de 8.01 y 247 valores atípicos en el rango [7.52, 8.0137].

```
par(mfrow=c(1,3))
boxplot(houses$median_house_value, ylab = "median_house_value", col = "darkseagreen1")
```

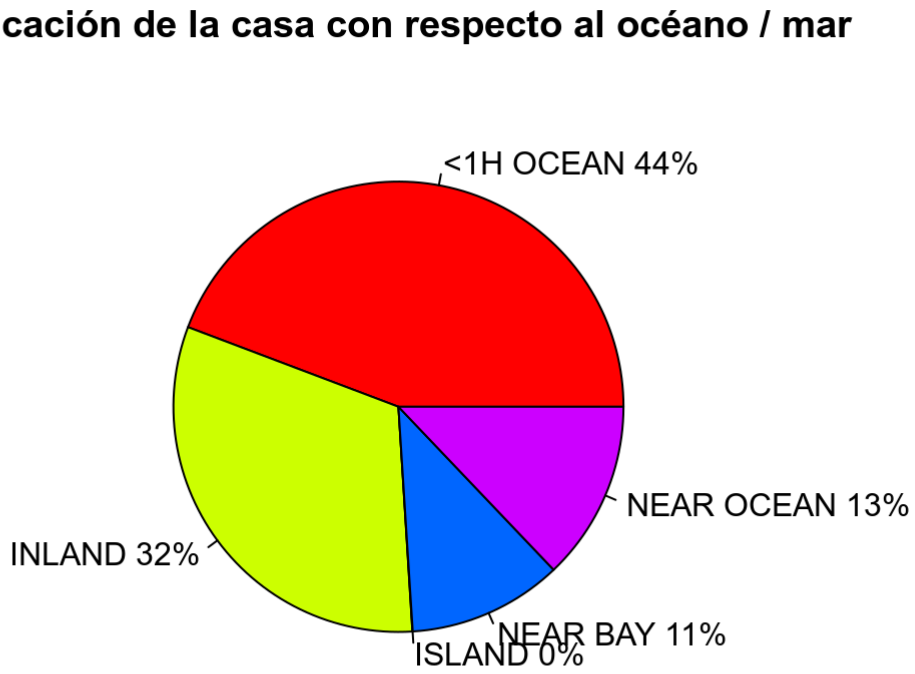


En el gráfico de **median_house_value** se observa una mediana aproximada de 179700, un mínimo de 14999, un máximo de 482200 y 435 valores atípicos en el rango [424400, 482200].

5.1.2 Gráficos cirulares

```
customPie <- function(slices, theTitle) {
  lbls = levels(slices)
  slices = table(slices)
  pct <- round(slices/sum(slices)*100)
  lbls <- paste(lbls, pct) # add percents to labels
  lbls <- paste(lbls,"%",sep="") # ad % to labels
  pie(slices,labels = lbls, col=rainbow(length(lbls)),main=theTitle)
}
```

```
customPie(houses$ocean_proximity, "ubicación de la casa con respecto al océano / mar")
```



En el gráfico de **ocean_proximity** se aprecia que el valor que más aparece es *Ocean*, con el 44% de la veces, seguido del valor *Inland* que aparece el 32% de las veces, el valor *Near ocean* con un 13%, el valor **Near bay* con un 11% y finalmente el valor *Island* que aparece el 0.00024%.

6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Como se menciona en el primer apartado, el objetivo del dataset es obtener un modelo que nos ayude a predecir precios de las casas en California, en base a sus caracteristicas. El preprocesado nos ha ayudado por una parte a localizar valores vacios que no permitirian aplicar el modelo correctamente, así como localizar valores outliers que debido a su lejania a la media de los valores pueden introducir un sesgo en el modelo usado (por ejemplo si usamos arboles de decision o clusterizamos con K-means).

Por otra parte el analisis estádistico llevado a cabo sobre la muestra nos arroja varias conclusiones importantes que sin ser el modelo final si puede servirnos de guia:

- Muchas columnas estan correlacionadas fuertemente y por tanto el dataset puede reducirse bastante mejorando el tiempo de procesado futuro.
- Podemos afirmar que el valor medio de la vivienda para los hogares dentro de un bloque es superior en los bloques cerca de la bahía con respecto a los que NO están cerca de la bahía.
- El valor medio de la vivienda tiene una correlacion con el valor del salario medio (median_income)

Asi por ejemplo para una muestra de casa localizada cerca de la bahia y con salarios medios altos de los habitantes podremos asegurar con un nivel alto de confianza que el precio de la casa tambien será alto. Esto se podria ver facilmene una vez apliquemos un modelo a nuestro dataset.

7. Código

El código de la práctica se encuentra disponible en https://github.com/Cs4r/california-housing-prices (https://github.com/Cs4r/california-housing-prices)

8. Contribuciones

Contribuciones	Firma
Investigación previa	César A. y Daniel V.
Redacción de las respuestas	César A. y Daniel V.
Desarrollo código	César A. y Daniel V.