Automobile Cleaning

Josu Alonso Castanedo & Emmanuel Rodríguez Belmonte

Diciembre 2020

Índice

1. Descripción del Dataset

Al realizar la búsqueda de una dataset lo suficientemente potente como para realizar un proyecto de minería de datos completo sobre ésta, diversas fuentes vinieron a la cabeza, por lo que se realizó una exploración de los principales resultados de sitios web tales como Kaggle o UCI ML Repository, quedándonos al final con la dataset Automobile de este último.

El porqué de dicha elección se basa en que esta dataset se presenta interesante para un proyecto de minería de datos debido a la presencia de valores nulos entre sus filas, lo que puede dar lugar a una limpieza más activa de los datos, así como al hecho de presentar una numerosa cantidad de variables, tanto categóricas como numéricas, entre las que destacamos como categórica `make` (fabricante del automóvil) y `price` (precio de éste) por su correlación e importancia a la hora de discriminar un coche de otro. Esta gran cantidad de variables también hace de la dataset una candidata a procesos de reducción de dimensionalidad, lo que implicaría un ejercico aún mayor sobre ésta y un valor añadido para la práctica.

Una vez comentado el atractivo de la dataset, podemos bajar a objetivos más concretos del análisis. Así pues, podríamos decir que el objetivo principal de este documento es discernir qué factores diferencian a los principales proveedores de vehículos automovilísticos, así como la relación entre la pertenencia a uno de estos grupos y el precio del automóvil en cuestión.

2. Integración y selección de los datos de interés

Con los objetivos en mente y las principales características de la dataset descritas verbalmente, es el momento de bajar a la programación y ver datos más específicos y objetivos de nuestra dataset.

Para continuar con la práctica, no obstante, primero deberemos cargar los datos dentro de nuestro entorno para poder trabajar con ellos:

```
# Importamos las librerías a utilizar durante el análisis
library(dplyr)
library(ggplot2)
library(reshape2)
library(corrgram)

# Establecemos una semilla por si queremos replicar el resultado
set.seed(555)


###### Carga del dataset ######

# Hardcodeamos el nombre de las columnas
dataset.names <- c(
  'symboling',
  'normalized-losses',
  'make',
  'fuel-type',
  'aspiration',
```

```
  'num-of-doors',
  'body-style',
  'drive-wheels',
  'engine-location',
  'wheel-base',
  'length',
  'width',
  'height',
  'curb-weight',
  'engine-type',
  'num-of-cylinders',
  'engine-size',
  'fuel-system',
  'bore',
  'stroke',
  'compression-ratio',
  'horsepower',
  'peak-rpm',
  'city-mpg',
  'highway-mpg',
  'price'
)

# Leemos nuestra versión del datase
dataset <- read.csv('../data/imports-85-OLD.data',
                    header = F,
                    col.names = dataset.names,
                    # En este dataset los NA están representados como '?'
                    na.strings = c('?','', ' '))

# Convertimos las columnas con variables categóricas a factores

factores <- c("make",
              "fuel.type",
              "aspiration",
              "num.of.doors",
              "body.style",
              "engine.type",
              "num.of.cylinders",
              "fuel.system",
              "drive.wheels",
              "engine.location")

for(f in factores){
  dataset[, f] <- as.factor(dataset[, f])
}
```

Una vez realizada la carga de nuestros datos, podemos pasar a realizar una serie de descripciones básicas:

```
###### Descripción del dataset ######

# Describimos la estructura de las variables
str(dataset)
```

```
## 'data.frame':    205 obs. of  26 variables:
##  $ symboling        : int  3 3 1 2 2 2 1 1 1 0 ...
##  $ normalized.losses: int  NA NA NA 164 164 NA 158 NA 158 NA ...
##  $ make             : Factor w/ 22 levels "alfa-romero",..: 1 1 1 2 2 2 2 2 2 2 ...
##  $ fuel.type        : Factor w/ 2 levels "diesel","gas": 2 2 2 2 2 2 2 2 2 2 ...
##  $ aspiration       : Factor w/ 2 levels "std","turbo": 1 1 1 1 1 1 1 1 2 2 ...
##  $ num.of.doors     : Factor w/ 2 levels "four","two": 2 2 2 1 1 2 1 1 1 2 ...
##  $ body.style       : Factor w/ 5 levels "convertible",..: 1 1 3 4 4 4 4 5 4 3 ...
##  $ drive.wheels     : Factor w/ 3 levels "4wd","fwd","rwd": 3 3 3 2 1 2 2 2 2 1 ...
##  $ engine.location  : Factor w/ 2 levels "front","rear": 1 1 1 1 1 1 1 1 1 1 ...
##  $ wheel.base       : num  88.6 88.6 94.5 99.8 99.4 ...
##  $ length           : num  169 169 171 177 177 ...
##  $ width            : num  64.1 64.1 65.5 66.2 66.4 66.3 71.4 71.4 71.4 67.9 ...
##  $ height           : num  48.8 48.8 52.4 54.3 54.3 53.1 55.7 55.7 55.9 52 ...
##  $ curb.weight      : int  2548 2548 2823 2337 2824 2507 2844 2954 3086 3053 ...
##  $ engine.type      : Factor w/ 7 levels "dohc","dohcv",..: 1 1 6 4 4 4 4 4 4 4 ...
##  $ num.of.cylinders : Factor w/ 7 levels "eight","five",..: 3 3 4 3 2 2 2 2 2 2 ...
##  $ engine.size      : int  130 130 152 109 136 136 136 136 131 131 ...
##  $ fuel.system      : Factor w/ 8 levels "1bbl","2bbl",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ bore             : num  3.47 3.47 2.68 3.19 3.19 3.19 3.19 3.19 3.13 3.13 ...
##  $ stroke           : num  2.68 2.68 3.47 3.4 3.4 3.4 3.4 3.4 3.4 3.4 ...
##  $ compression.ratio: num  9 9 9 10 8 8.5 8.5 8.5 8.3 7 ...
##  $ horsepower       : int  111 111 154 102 115 110 110 110 140 160 ...
##  $ peak.rpm         : int  5000 5000 5000 5500 5500 5500 5500 5500 5500 5500 ...
##  $ city.mpg         : int  21 21 19 24 18 19 19 19 17 16 ...
##  $ highway.mpg      : int  27 27 26 30 22 25 25 25 20 22 ...
##  $ price            : int  13495 16500 16500 13950 17450 15250 17710 18920 23875 NA ...
```

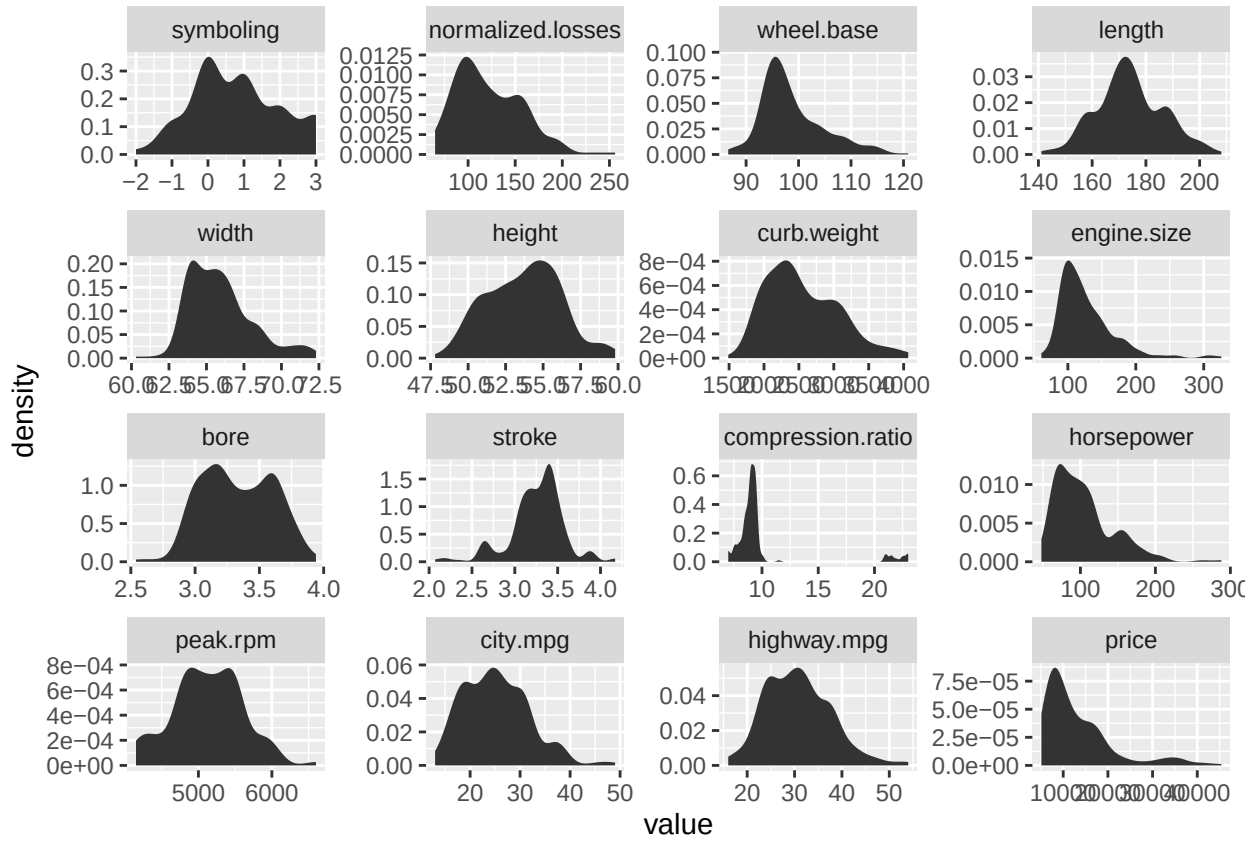
```
# La distribución de sus atributos
summary(dataset)
```

```
##    symboling       normalized.losses         make      fuel.type   aspiration
##  Min.   :-2.0000   Min.   : 65       toyota    : 32   diesel: 20   std  :168
##  1st Qu.: 0.0000   1st Qu.: 94       nissan    : 18   gas   :185   turbo: 37
##  Median : 1.0000   Median :115       mazda     : 17
##  Mean   : 0.8341   Mean   :122       honda     : 13
##  3rd Qu.: 2.0000   3rd Qu.:150       mitsubishi: 13
##  Max.   : 3.0000   Max.   :256       subaru    : 12
##                    NA's   :41        (Other)   :100
```

```
##  num.of.doors       body.style drive.wheels engine.location   wheel.base
##  four:114     convertible: 6   4wd:  9      front:202       Min.   : 86.60
##  two : 89     hardtop    : 8   fwd:120      rear :  3       1st Qu.: 94.50
##  NA's:  2     hatchback  :70   rwd: 76                      Median : 97.00
##               sedan      :96                                Mean   : 98.76
##               wagon      :25                                3rd Qu.:102.40
##                                                            Max.   :120.90
##
##      length          width           height       curb.weight   engine.type
##  Min.   :141.1   Min.   :60.30   Min.   :47.80   Min.   :1488   dohc : 12
##  1st Qu.:166.3   1st Qu.:64.10   1st Qu.:52.00   1st Qu.:2145   dohcv:  1
##  Median :173.2   Median :65.50   Median :54.10   Median :2414   l    : 12
##  Mean   :174.0   Mean   :65.91   Mean   :53.72   Mean   :2556   ohc  :148
##  3rd Qu.:183.1   3rd Qu.:66.90   3rd Qu.:55.50   3rd Qu.:2935   ohcf : 15
##  Max.   :208.1   Max.   :72.30   Max.   :59.80   Max.   :4066   ohcv : 13
##                                                                 rotor:  4
##  num.of.cylinders  engine.size     fuel.system      bore          stroke
##  eight :  5       Min.   : 61.0   mpfi   :94   Min.   :2.54   Min.   :2.070
##  five  : 11       1st Qu.: 97.0   2bbl   :66   1st Qu.:3.15   1st Qu.:3.110
##  four  :159       Median :120.0   idi    :20   Median :3.31   Median :3.290
##  six   : 24       Mean   :126.9   1bbl   :11   Mean   :3.33   Mean   :3.255
##  three :  1       3rd Qu.:141.0   spdi   : 9   3rd Qu.:3.59   3rd Qu.:3.410
##  twelve:  1       Max.   :326.0   4bbl   : 3   Max.   :3.94   Max.   :4.170
##  two   :  4                       (Other): 2   NA's   :4      NA's   :4
##  compression.ratio   horsepower       peak.rpm       city.mpg
##  Min.   : 7.00     Min.   : 48.0   Min.   :4150   Min.   :13.00
##  1st Qu.: 8.60     1st Qu.: 70.0   1st Qu.:4800   1st Qu.:19.00
##  Median : 9.00     Median : 95.0   Median :5200   Median :24.00
##  Mean   :10.14     Mean   :104.3   Mean   :5125   Mean   :25.22
##  3rd Qu.: 9.40     3rd Qu.:116.0   3rd Qu.:5500   3rd Qu.:30.00
##  Max.   :23.00     Max.   :288.0   Max.   :6600   Max.   :49.00
##                    NA's   :2       NA's   :2
##   highway.mpg        price
##  Min.   :16.00   Min.   : 5118
##  1st Qu.:25.00   1st Qu.: 7775
##  Median :30.00   Median :10295
##  Mean   :30.75   Mean   :13207
##  3rd Qu.:34.00   3rd Qu.:16500
##  Max.   :54.00   Max.   :45400
##                  NA's   :4
```

```
# Creamos una versión "melted" de nuestra dataset para visualizar más fácilmente
melt.dataset <- melt(dataset)

# Visualizamos la distribución de los atributos continuos de nuestra dataset
ggplot(data = melt.dataset, aes(x = value)) +
  stat_density() +
```

```
facet_wrap(~variable, scales = "free")
```



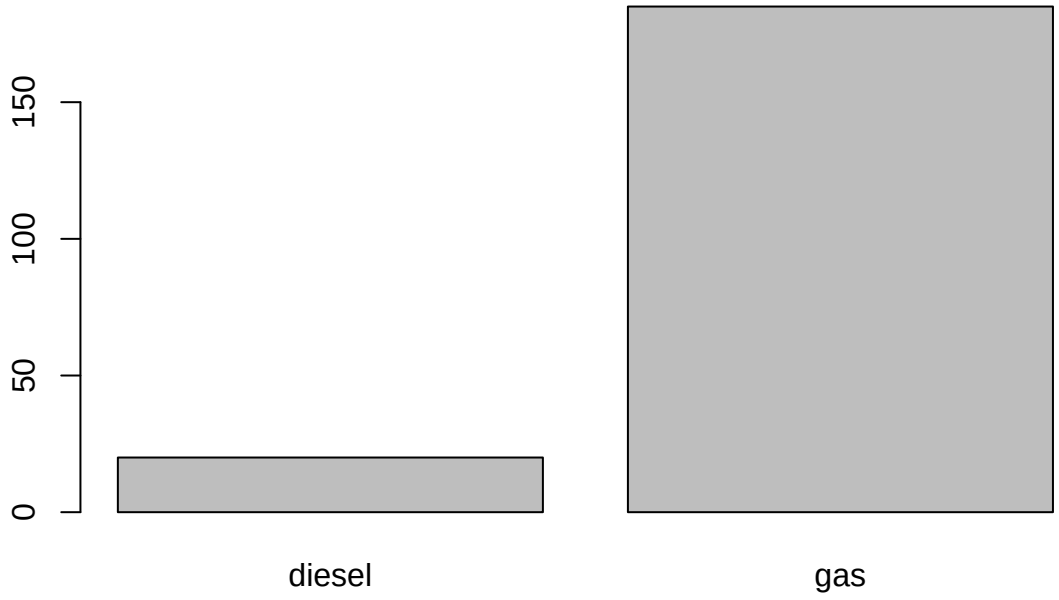
Como podemos observar, nuestra dataset está formada por **205 observaciones y 26 variables**, de las cuáles **16 son numéricas y 10 son categóricas**. también observamos que la mayoría de observaciones pertenecen a la clase `gas` de la variable `fuel.type`, así como que la mayoría también se encuentran dentro del subgrupo `std` de la variable `aspiration`.

Por lo que corresponde a las variables objetivo de nuestro análisis, vemos que la distribución de `make` es estable, ya que a pesar de tener el primer fabricante (*toyota*) 32 observaciones (15,6 % del total de las observaciones) el resto del top 6 de facbricantes se mantiene en el rango 18-12, por lo que podríamos coger estos 6 fabricantes como subgrupos en los cuales dividir las observaciones y aún así mantener una muestra bastante equitativa de los datos. El precio por contra, mantinene un rango **5118 - 45400**, lo que indica que hay observaciones por bastante por encima del 3º cuartil (16500), pero no muy por debajo del primero (7775).

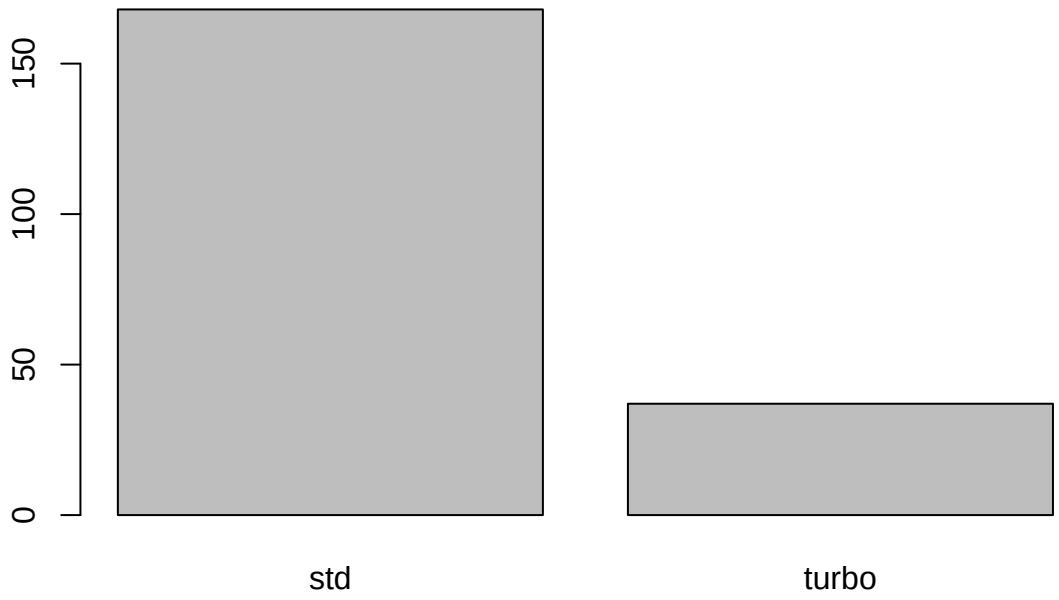
El resto de variables presentan una distribución bastante normal, a destacar el hecho de que `compression.ratio` presenta claros valores extremos alrededor de su máximo, como podemos ver en la gráfica de su distribución. Por su parte, `horsepower`, `engine.size` y `normalized.losses` presentan una distribución similar a `price`, alejándose de la normalidad, lo que podría indicar una correlación entre estas variables.

Si visualizamos las variables categóricas que más sesgan la dataset (`fuel.type`,`aspiration` y `make`), ya que así podremos obtener un poco más de información sobre éstas:

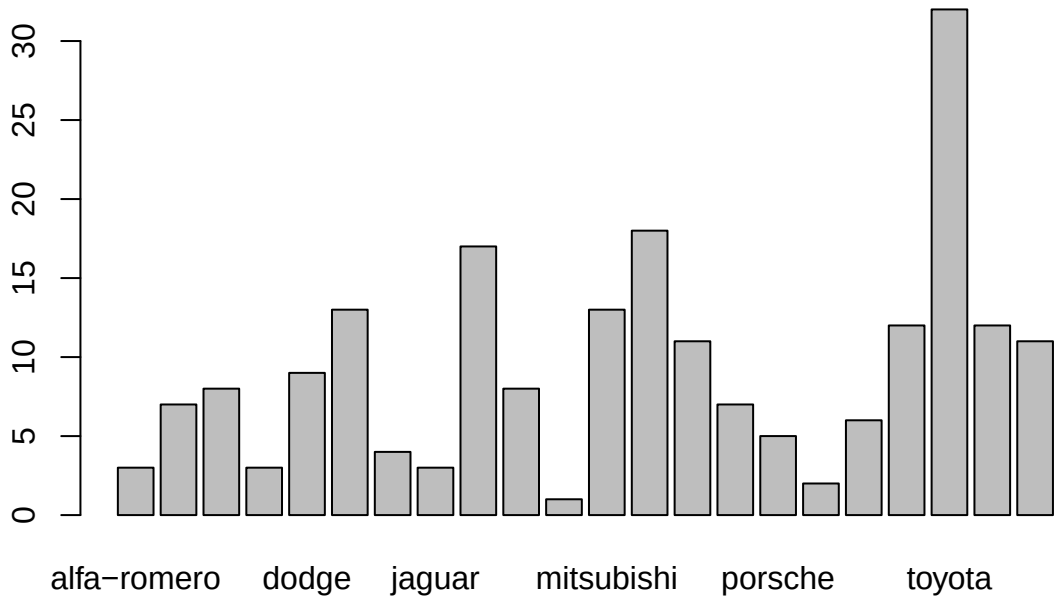
```r
# Visualizamos la distribución de un par de factores
barplot(with(dataset, table(fuel.type)))        # fuel.type
```




```r
barplot(with(dataset, table(aspiration)))       # aspiration
```

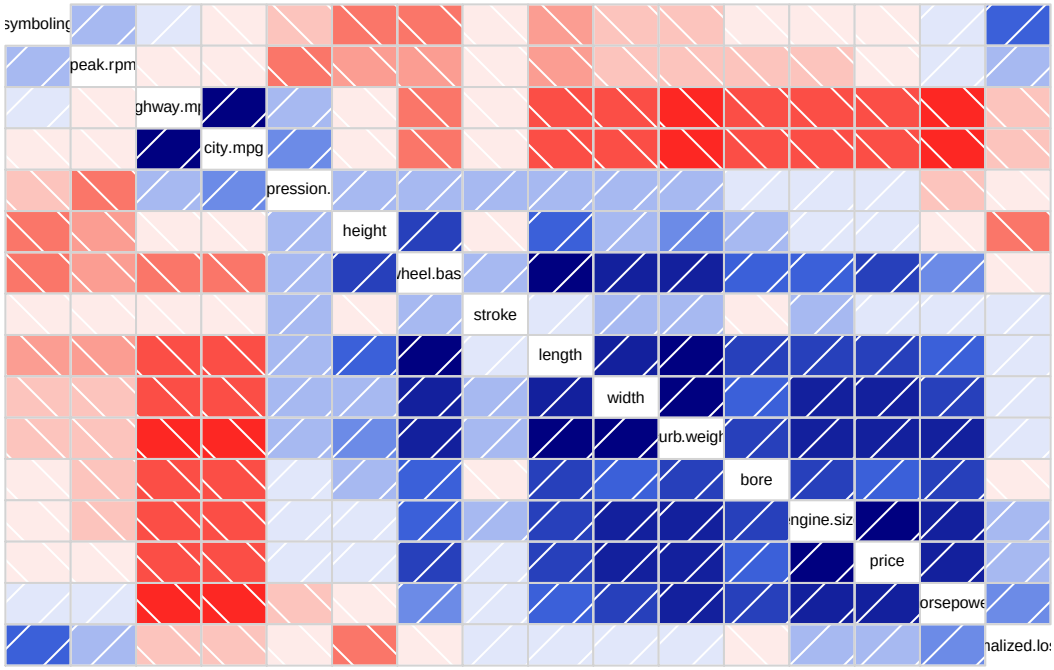



```
barplot(with(dataset, table(make)))              # make
```



En lo que respecta a la correlación entre las variables, también podemos realizar visualizaciones:

```
# Visualizamos la correlación de estos atributos
corrgram(dataset,
         order = T,
         lower.panel = panel.shade,
         text.panel = panel.txt)
```



Y comprobar que hay grandes correlaciones entre `price` y `engine.size`, como habíamos anunciado, así como entre otras variables, como `city.mpg` y `highway.mpg` (ya que si un automóvil tiene un gran consumo en ciudad también lo tendrá en autopista). A su vez, vemos como estas dos últimas variables afectan negativamente al precio, lo que también tiene sentido.

3. Limpieza de los datos

Una vez seleccionados los datos, hay una serie de pasos que tenemos que realizar para asegurar una cierta limpieza de éstos. Entre los pasos a realizar, los más importantes son el tratamiento de valores nulos, los cuáles ya hemos identificado y asignado el valor **NA** en la carga de datos, y el de los valores extremos, también identificados dentro de la variable `compression.ratio`.

3.1. Elementos vacíos

Primero, debemos comprobar qué columnas presentan valores vacíos y en qué proporción de su total para decidir qué estrategia tomar ante su presencia en nuestros datos. Para eso, veremos sus valores absolutos y relativos mediante la función `colSums`:

```
# Buscamos valores nulos
colSums(is.na(dataset))
```

```
##         symboling normalized.losses              make         fuel.type
##                 0                41                 0                 0
##        aspiration      num.of.doors        body.style      drive.wheels
##                 0                 2                 0                 0
##   engine.location        wheel.base            length             width
##                 0                 0                 0                 0
##            height       curb.weight       engine.type  num.of.cylinders
##                 0                 0                 0                 0
##       engine.size       fuel.system              bore            stroke
##                 0                 0                 4                 4
## compression.ratio        horsepower          peak.rpm          city.mpg
##                 0                 2                 2                 0
##       highway.mpg             price
##                 0                 4
```

```
# Representamos su peso sobre el total de las observaciones
colSums(is.na(dataset))/nrow(dataset)
```

```
##         symboling normalized.losses              make         fuel.type
##       0.000000000       0.200000000       0.000000000       0.000000000
##        aspiration      num.of.doors        body.style      drive.wheels
##       0.000000000       0.009756098       0.000000000       0.000000000
##   engine.location        wheel.base            length             width
##       0.000000000       0.000000000       0.000000000       0.000000000
##            height       curb.weight       engine.type  num.of.cylinders
##       0.000000000       0.000000000       0.000000000       0.000000000
##       engine.size       fuel.system              bore            stroke
##       0.000000000       0.000000000       0.019512195       0.019512195
## compression.ratio        horsepower          peak.rpm          city.mpg
##       0.000000000       0.009756098       0.009756098       0.000000000
##       highway.mpg             price
##       0.000000000       0.019512195
```

```
# Eliminamos observaciones con valor "?" en algunos atributos críticos
dataset <- dataset[(!is.na(dataset$horsepower)) & (!is.na(dataset$num.of.doors)) & (!is.na(datas

# E imputamos la media para los siguientes
for (column in c('normalized.losses', 'bore', 'stroke')){
  dataset[is.na(dataset[, column]), column] <- mean(dataset[, column], na.rm = T)
}
```

3.2. Identificación y tratamiento de valores extremos

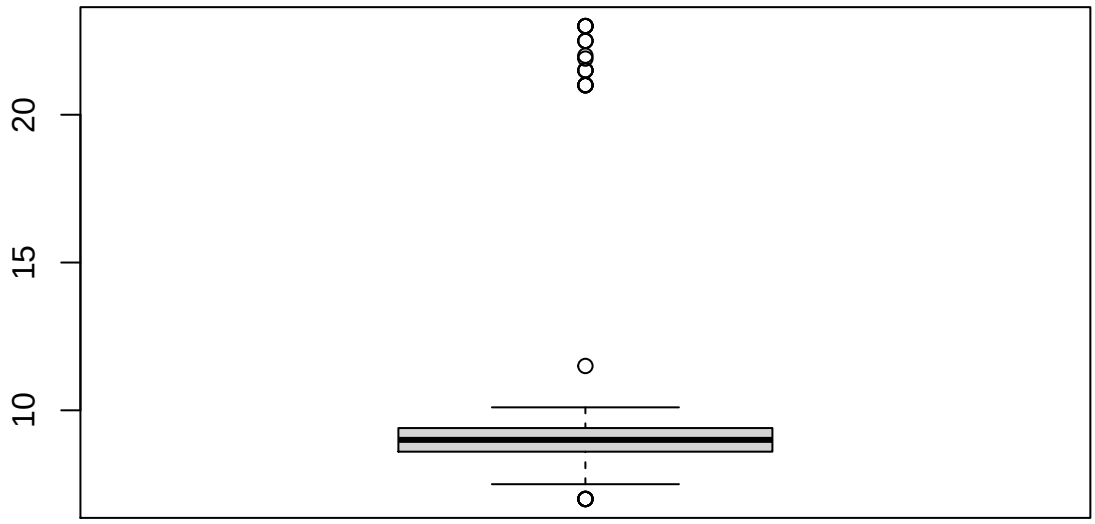
Al describir la dataset nos percatamos de la presencia de valores extremos en la variable `compression.ratio` alrededor de su valor máximo. Pero, a pesar de parecer clara su presencia observando el histograma de la variable, podemos asegurarnos todavía más de su presencia

mediante la realización de **boxplots** que nos muestren la relación de estos valores con sus cuartiles, y cómo de extremos son dichos valores:

```
# Recordamos la distribución de compression.ratio
summary(dataset$compression.ratio)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00    8.60    9.00   10.13    9.40   23.00
```

```
# Realizamos nuestro plot
boxplot(dataset$compression.ratio)
```



Como podemos observar, estos valores están por encima de **20**, cuando el tercer cuartil de la variable es **9,40**, por lo que suponen una desviación de más de **4,58 veces el rango intercuartílico** desde dicho cuartil, por lo que podemos confirmar su estatus de valores extremos.

Una vez confirmada la presencia de valores extremos en esta variable, tenemos que decidir qué estrategia seguir para su tratamiento. Teniendo en cuenta su lejanía del 3 cuartil, así como su escaso número, creemos que el tratamiento por eliminación es el más acertado, ya que la cantidad de información a perder es mínima y el riesgo de imputar un valor medio sin estar seguros de un correcto registro del resto de variables es alto. Por lo tanto, procedemos a la eliminación de estas observaciones de nuestra dataset:

```r
# Eliminamos los datos que corresponden a dicho outlier
dataset = dataset[!dataset$compression.ratio >= 20, ]
```

4. Análisis de los datos

Una vez hemos observado las variables en general, vamos a realizar un estudio estadístico para sacar conclusiones con respecto a los atributos que consideremos importantes al inicio del documento (precio y marca según continente, ya que cada marca individual no es lo bastante grande como para generar grupos con suficiente información).

4.1. Selección de los grupos

Vamos a estudiar cómo afectan las marcas según continente en los distintos parámetros, haciendo especial énfasis en el precio.

Como vamos a poner especial atención en el precio, crearemos un nuevo atributo price.range, para poder realizar análisis categóricos con el precio, tomando como referencia los siguientes rangos:

- low: Precio entre 0 y el primer cuartil (7775).
- medium: Precio entre el primer y tercer cuartil (7775 y 16500).
- high: Precio mayor al tercer cuartil (16500).

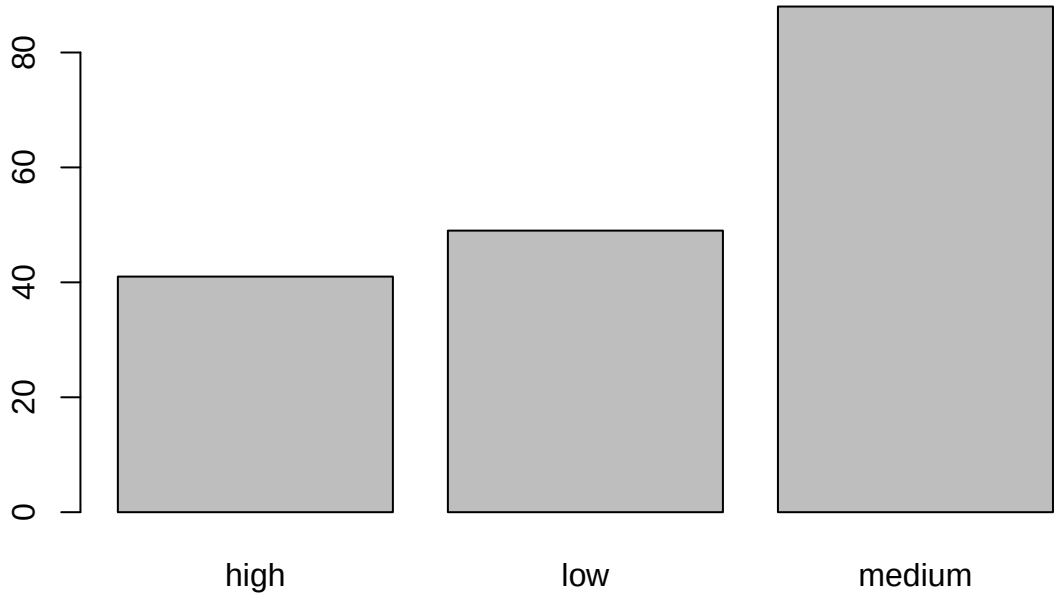
```r
nrow(dataset)
```

```
## [1] 178
```

```r
# Creación del nuevo campo price.range para los rangos definidos
for (i in 1:nrow(dataset)) {
  if (dataset[i, 'price'] <= 7775){
    dataset[i, 'price.range'] = 'low'
  } else if(dataset[i, 'price'] <= 16500){
    dataset[i, 'price.range'] = 'medium'
  } else{
    dataset[i, 'price.range'] = 'high'
  }
}

# Guardamos como factor el nuevo atributo
dataset[,'price.range'] = as.factor(dataset[,'price.range'])
```

Podemos ver cómo queda el nuevo atributo categórico en la siguiente tabla:

```r
# Visualizamos la distribución de price.range
barplot(with(dataset, table(price.range)))
```

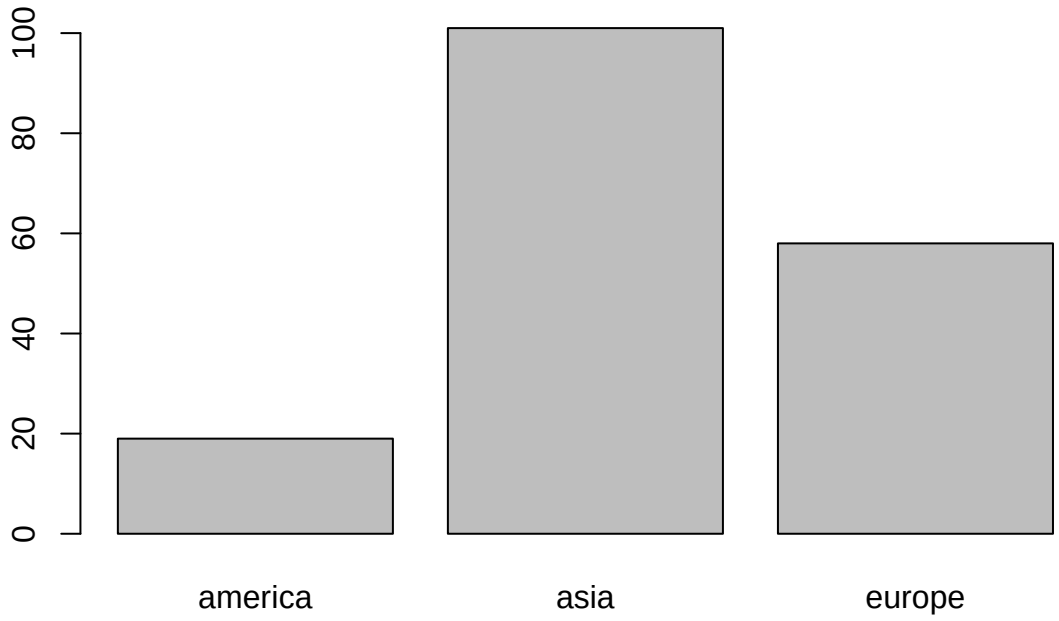


Por último, vamos a crear **n** grupos, uno por cada proveedor, para poder ver cómo la marca según continente afecta a los distintos atributos (y principalmente al precio), por lo que realizaremos tres tipos de análisis:

- Correlación de los atributos con el precio por marca según continente, comparando con la general.
- Predicción del precio con regresión lineal por marca según continente y en general, comparando los resultados obtenidos entre marcas según continente y con el general.
- Predicción del precio con regresión logística por marca según continente y en general, comparando los resultados obtenidos entre marcas según continente y con el general.

```r
maker_imputer <- function(maker){
  if(maker %in% c('alfa-romero', 'audi', 'bmw', 'mercedes-benz', 'volvo',
                  'peugot', 'porsche', 'saab', 'volkswagen', 'jaguar')){
    x <- 'europe'
  } else if (maker %in% c('chevrolet', 'dodge', 'mercury', 'plymouth')){
    x <- 'america'
  } else {
    x <- 'asia'
```

```
  }
  return(x)
}

dataset$make.continent <- sapply(dataset$make, maker_imputer)
barplot(with(dataset, table(make.continent)))
```




```
# Separamos el dataset por marcas
continents <- c('america', 'asia', 'europe')

dataset_by_continent <- list()

for (continent in continents) {
  dataset_by_continent[continent] =
    list(dataset[dataset$make.continent == continent, ])
}
```

4.2. Comprobación de la normalidad y homogeneidad de la varianza

Vamos a realizar la comprobación de la normalidad de todos los datos y de la homogeneidad de la varianza utilizando el teorema del límite central y análisis de la varianza (ANOVA) respectivamente.

La razón de esto, es que no tenemos necesidad de saber la normalidad u homogeneidad por los grupos, ya que no son condición para ninguno de los anális que vamos a realizar.

El teorema del límite central, de forma resumida, nos permite asumir la normalidad de los datos cuando el número de muestras supera un cierto número (30).

Por tanto, vamos a comprobar el número de observaciones de las que disponemos tras la limpieza de los datos, para confirmar que tenemos más de 30.

```
# Mostramos el número de elementos tras la limpieza
nrow(dataset)
```

```
## [1] 178
```

Tras la limpieza, el número de observaciones es 178, por lo que gracias al teorema del límite central, podemos asumir la normalidad del conjunto de datos.

Para acabar con este punto, realizamos el análisis de la homocedasticidad, u homogeneidad de la varianza, sobre el dataset. Para poder realizar esta comprobación, usaremos el test de Barlett, ya que la diferencia en tamaño de los diferentes grupos a analizar es notable y este test nos asegura unos resultados comparables aún así siempre y cuando se cumpla el principio de normalidad (constatado en el punto anterior).

```
# Recorremos cada una de las variables más interesantes del
# dataset y cada uno de los continentes
variables.to.test<- c('price', 'normalized.losses',
                      'horsepower','peak.rpm','engine.size')

for(variable in variables.to.test){
  asia <- dataset_by_continent[['asia']][,variable]
  europe <- dataset_by_continent[['europe']][,variable]
  america <- dataset_by_continent[['america']][,variable]
  print(bartlett.test(list(asia, europe, america)))
}
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  list(asia, europe, america)
## Bartlett's K-squared = 92.366, df = 2, p-value < 2.2e-16
##
##
##  Bartlett test of homogeneity of variances
##
## data:  list(asia, europe, america)
## Bartlett's K-squared = 3.0892, df = 2, p-value = 0.2134
##
##
```

```
##  Bartlett test of homogeneity of variances
##
## data:  list(asia, europe, america)
## Bartlett's K-squared = 4.7154, df = 2, p-value = 0.09464
##
##
##  Bartlett test of homogeneity of variances
##
## data:  list(asia, europe, america)
## Bartlett's K-squared = 13.77, df = 2, p-value = 0.001023
##
##
##  Bartlett test of homogeneity of variances
##
## data:  list(asia, europe, america)
## Bartlett's K-squared = 38.34, df = 2, p-value = 4.727e-09
```

Como podemos comprobar, el resultado obtenido muestra que con un p-value aceptable para `horsepower` y `normalized.losses` aceptamos la hipótesis nula, que nos indica que las varianzas son iguales. En cambio, para las otras 3 variables, vemos unos valores de p-value bastante bajos, que indican que la varianza de los atributos es diferente en los grupos elegidos.

4.3. Aplicación de las pruebas estadísticas

Para la primera prueba, la comparación de la correlación entre los atributos y el precio por marca (según continente) y de forma general, por lo que realizamos los cálculos y los guardamos para visualizarla y sacar conclusiones de ello en los siguientes apartados.

```
# Calculamos las correlaciones
cuantitative_fields <- c('symboling', 'normalized.losses',
                         'wheel.base', 'length', 'width',
                         'height', 'curb.weight', 'engine.size',
                         'bore', 'stroke', 'compression.ratio',
                         'horsepower', 'peak.rpm', 'city.mpg',
                         'highway.mpg')

corr_by_continent <- list()

for (continent in continents) {
 for (field in cuantitative_fields) {
  corr_by_continent[[continent]][field] =
    cor(dataset_by_continent[[continent]][,field],
        dataset_by_continent[[continent]][,'price'])
 }
}

general_correlation <- list()
```

```
for (field in cuantitative_fields) {
  general_correlation[field] = cor(dataset[,field], dataset$price)
}
```

Lo siguiente es hacer un modelo de regresión lineal para cada uno de los continentes y el general que intente entender la causística detrás de las variaciones en el precio:

```
# Creación de los modelos de regresión lineal por continente
lm_asia <- lm (formula = price ~ symboling +
                 normalized.losses + wheel.base + length +
                 width + height + curb.weight + engine.size +
                 bore + stroke + compression.ratio + horsepower +
                 peak.rpm + city.mpg + highway.mpg,
               data = dataset_by_continent[['asia']])

lm_america <- lm (formula = price ~ symboling +
                    normalized.losses + wheel.base + length +
                    width + height + curb.weight + engine.size +
                    bore + stroke + compression.ratio + horsepower +
                    peak.rpm + city.mpg + highway.mpg,
                  data = dataset_by_continent[['america']])

lm_europe <- lm (formula = price ~ symboling +
                   normalized.losses + wheel.base + length +
                   width + height + curb.weight + engine.size +
                   bore + stroke + compression.ratio + horsepower +
                   peak.rpm + city.mpg + highway.mpg,
                 data = dataset_by_continent[['europe']])

lm_general <- lm (formula = price ~ symboling +
                    normalized.losses + wheel.base + length +
                    width + height + curb.weight + engine.size +
                    bore + stroke + compression.ratio + horsepower +
                    peak.rpm + city.mpg + highway.mpg, data = dataset)

# Obtenemos sus valores para R^2 ajustado
r2 <- c(summary(lm_asia)$adj.r.squared, summary(lm_america)$adj.r.squared,
        summary(lm_europe)$adj.r.squared, summary(lm_general)$adj.r.squared)
```

Por último, los modelos de regresión logística:

```
# Creación de los modelos de regresión logística por continente
glm_asia <- glm (formula = price.range ~ symboling +
                   normalized.losses + wheel.base + length +
                   width + height + curb.weight + engine.size +
                   bore + stroke + compression.ratio + horsepower +
                   peak.rpm + city.mpg + highway.mpg,
```

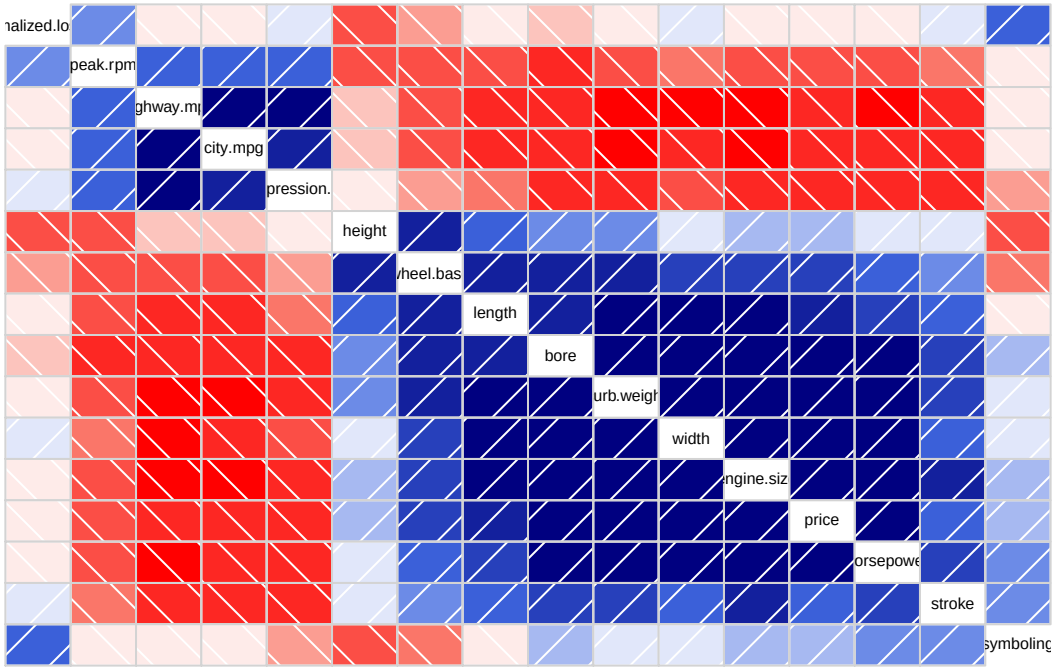
```
                  data = dataset_by_continent[['asia']],
                  family=binomial)

glm_america <- glm (formula = price.range ~ symboling +
                      normalized.losses + wheel.base + length +
                      width + height + curb.weight + engine.size +
                      bore + stroke + compression.ratio + horsepower +
                      peak.rpm + city.mpg + highway.mpg,
                    data = dataset_by_continent[['america']],
                    family=binomial)

glm_europe <- glm (formula = price.range ~ symboling +
                     normalized.losses + wheel.base + length +
                     width + height + curb.weight + engine.size +
                     bore + stroke + compression.ratio + horsepower +
                     peak.rpm + city.mpg + highway.mpg,
                   data = dataset_by_continent[['europe']],
                   family=binomial)

glm_general <- glm (formula = price.range ~ symboling +
                      normalized.losses + wheel.base + length +
                      width + height + curb.weight + engine.size +
                      bore + stroke + compression.ratio + horsepower +
                      peak.rpm + city.mpg + highway.mpg,
                      data = dataset, family=binomial)

# Obtenemos el parámetro AIC
aic <- c(summary(glm_asia)$aic, summary(glm_america)$aic,
         summary(glm_europe)$aic, summary(glm_general)$aic)
```

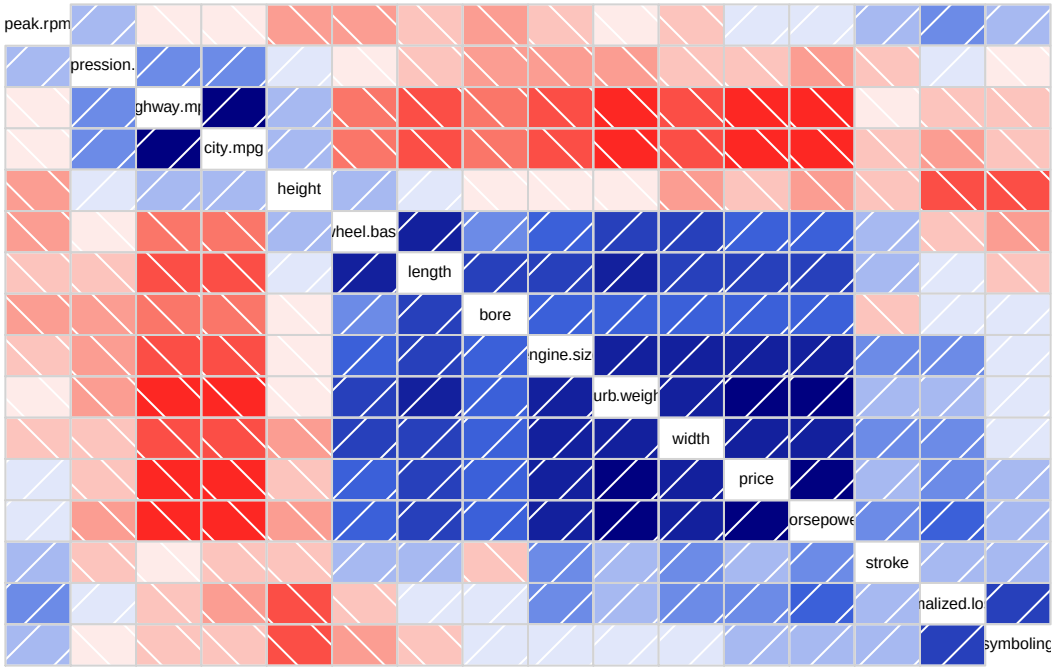
5. Representación de los resultados

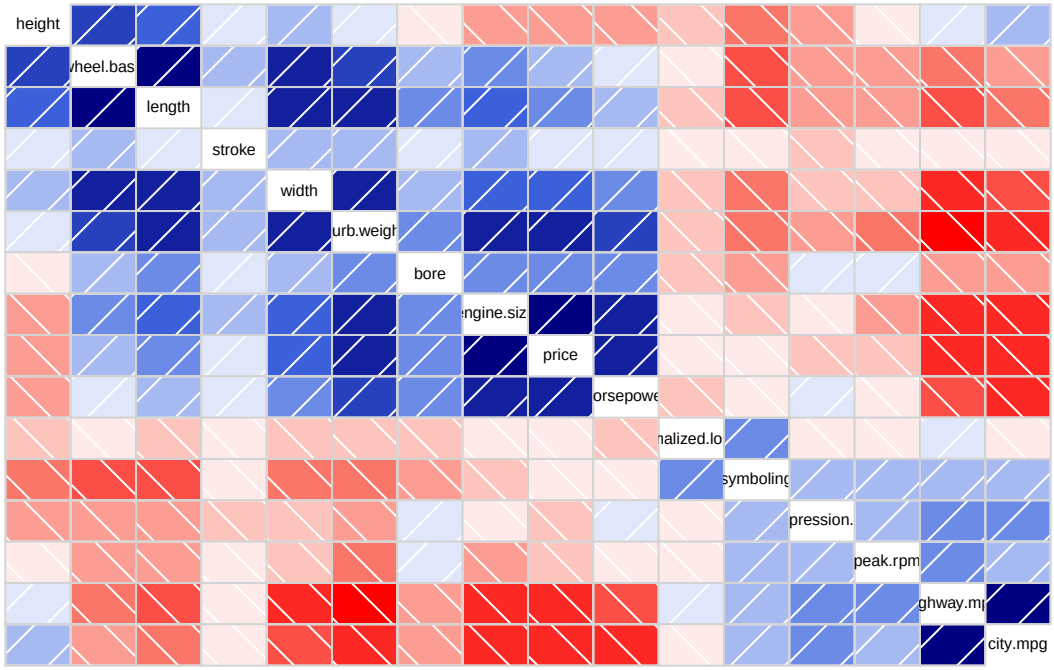
Una vez resueltas las pruebas estadísticas, podemos pasar a visualizar los resultados del análisis del apartado anterior para tener una referencia sobre la que obtener conclusiones al final.

Empezamos con la grafícación de la correlación entre atributos de cada continente.

```
# Mostramos la correlación de los atributos numéricos de cada continente
for (continent in continents){
  corrgram(dataset_by_continent[[continent]],
         order = T,
         lower.panel = panel.shade,
         text.panel = panel.txt)
}
```

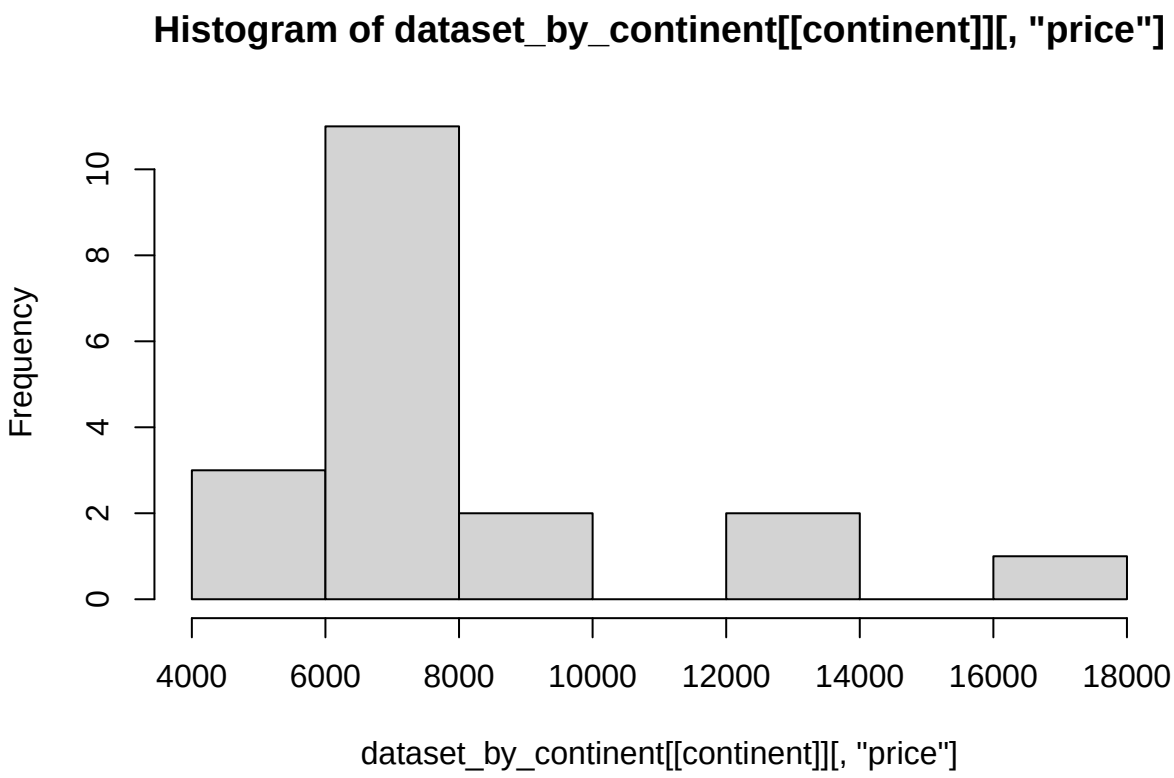

alized.lo
peak.rpm
ghway.mp
city.mpg
pression.
height
wheel.bas
length
bore
urb.weigh
width
ngine.siz
price
orsepowe
stroke
symboling

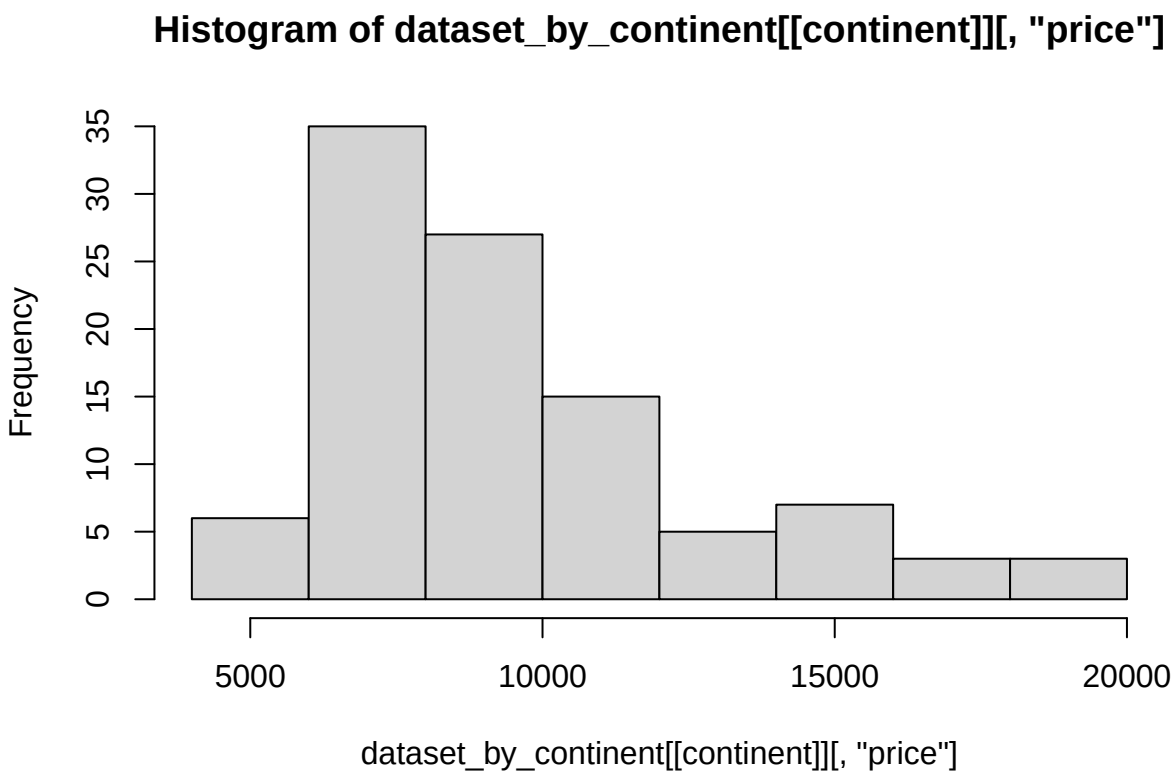

peak.rpm
pression.
ghway.mp
city.mpg
height
wheel.bas
length
bore
ngine.siz
urb.weigh
width
price
orsepowe
stroke
nalized.lo
symboling

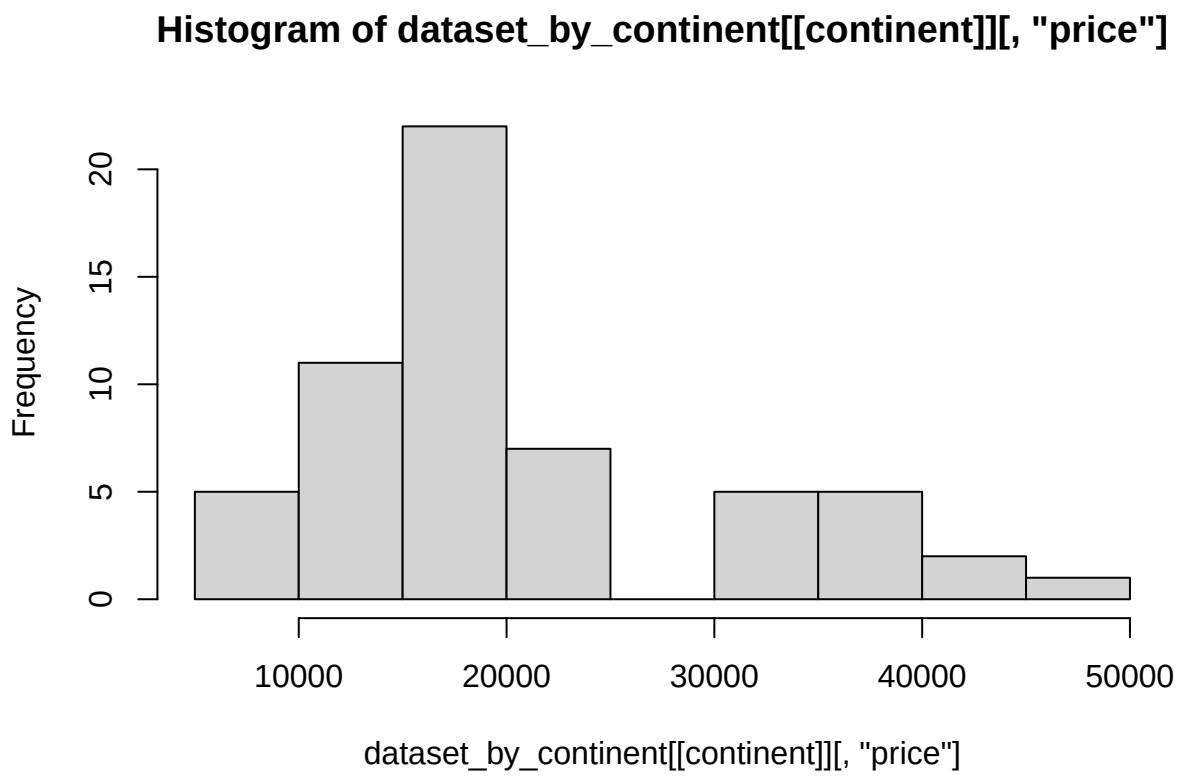


Además, para tener más información visual de la que sacar conclusiones, vamos a mostrar la distribución de precio por continente con sus cuartiles.

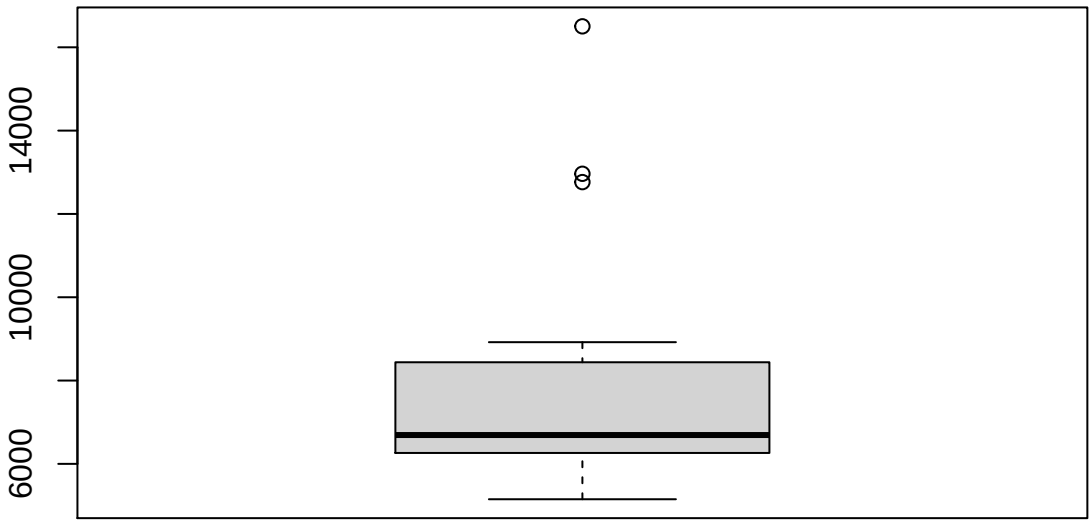
```
# Mostramos la distribución de precio de cada continente
for (continent in continents){
  hist(dataset_by_continent[[continent]][,'price'])
}
```

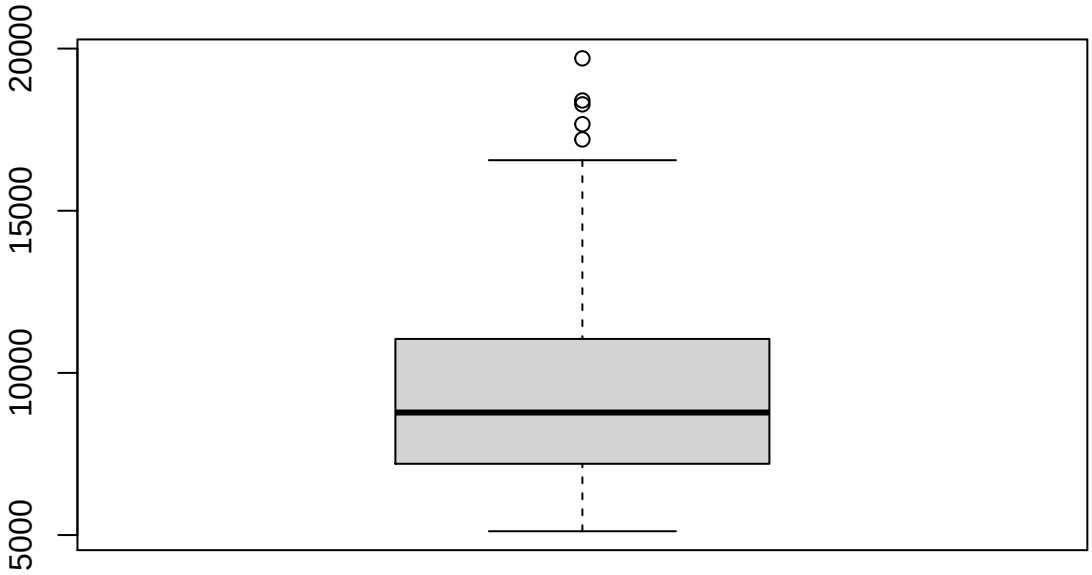

Histogram of dataset_by_continent[[continent]][, "price"]
Frequency
0
2
4
6
8
10
4000
6000
8000
10000
12000
14000
16000
18000
dataset_by_continent[[continent]][, "price"]

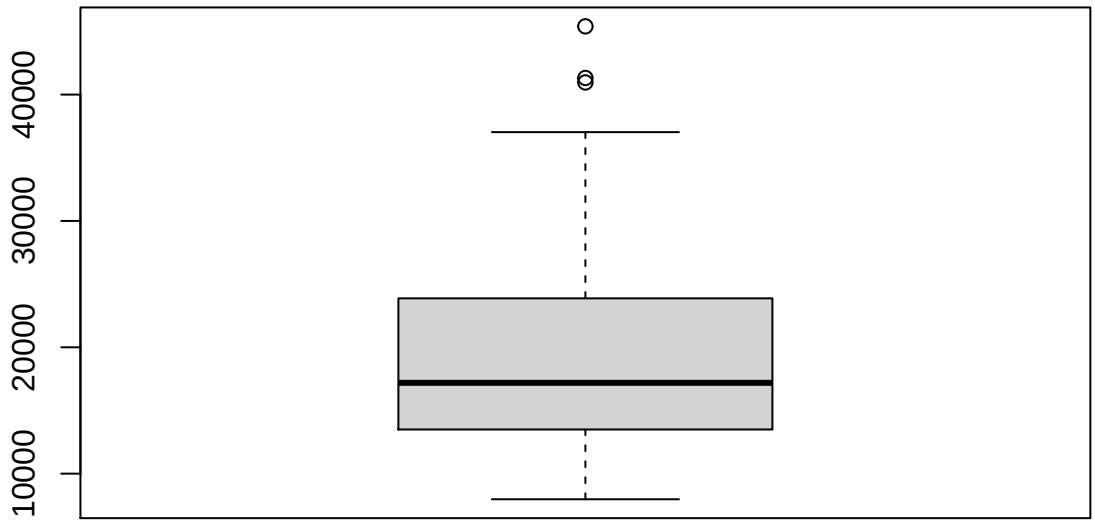

Histogram of dataset_by_continent[[continent]][, "price"]
Frequency
0
5
10
15
20
25
30
35
5000
10000
15000
20000
dataset_by_continent[[continent]][, "price"]




```
# Y la comparamos con sus quartiles con los boxplots
for (continent in continents){
  boxplot(dataset_by_continent[[continent]][,'price'])
}
```

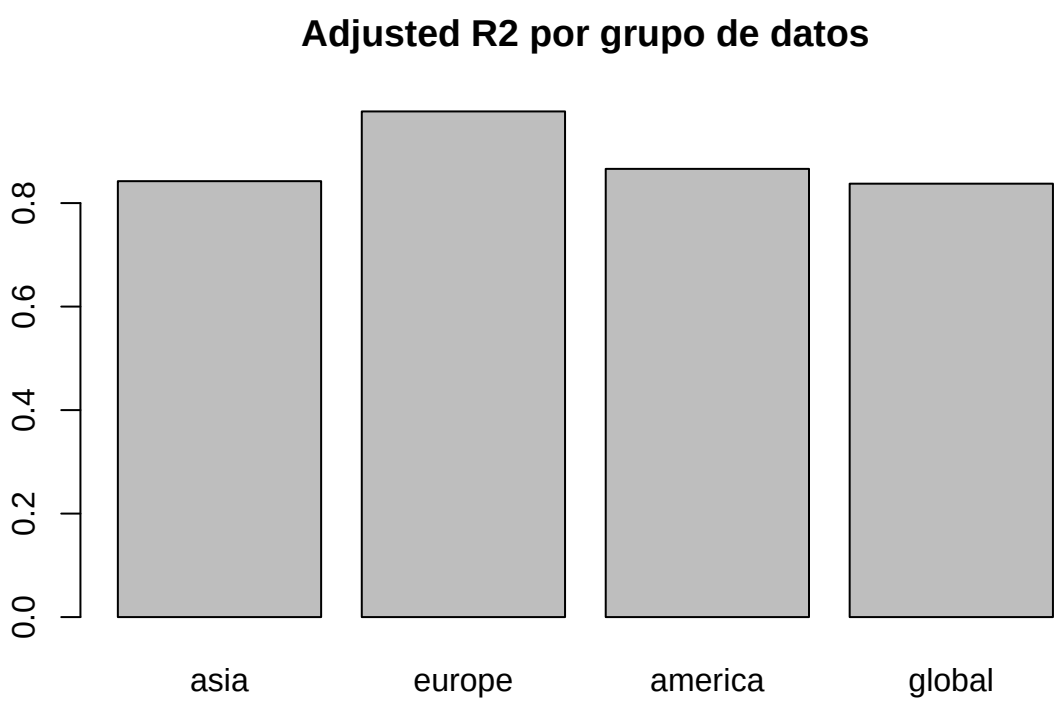

14000
10000
6000


20000
15000
10000
5000

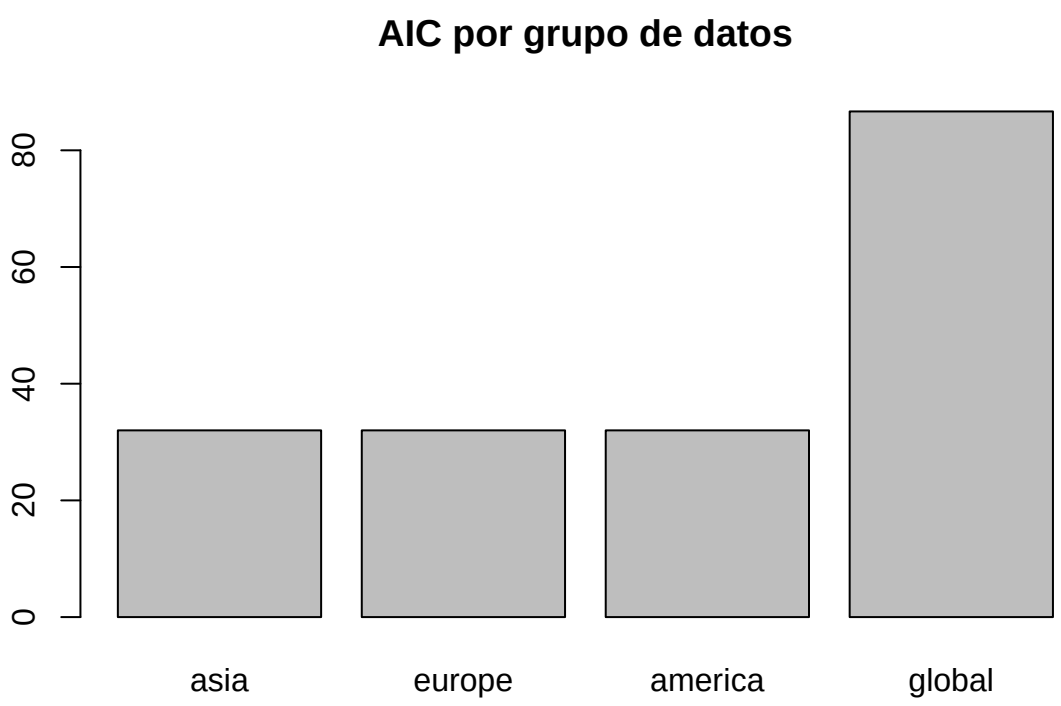


Por último, representaremos los resultados obtenidos en los 2 métodos de clasificación para ver qué conjunto de datos se adecúa más a los modelos de aprendizaje supervisado que hemos desarrollado:

```
barplot(height=r2, names.arg=c('asia','europe','america','global'),
        main = 'Adjusted R2 por grupo de datos')
```




```
barplot(height=aic, names.arg=c('asia','europe','america','global'),
        main = 'AIC por grupo de datos')
```



6. Resolución del problema

Como podemos comprobar, los elementos de los automóviles europeos tienen correlaciones menos fuertes entre ellos, llegando a ver colores mucho más apagados en el gráfico perteneciente a este continente. Además, incluso nuestra variable original `price` recibe una influencia por el resto de sus compañeras mucho más pequeña que en el resto de continentes, lo que podría indicar una predisposición hacia un modelo de coche más estándar y menos guiado por variables extrapoladas que empujasen al resto a variar en mayor grado.

Continuando con un razonamiento similar, observamos que los fabricantes americanos tienden a incorporar una mayor variabilidad en los atributos de sus coches, llegando a mostrar coeficientes de corelación mucho más altos. Vemos en especial que la variable precio es fuertemente influida por muchas de las variables, dejando `height` fuera de esta ecuación. En lo relativo al precio, observamos que los coches europeos son los que tienen un rango de precio mayor, llegando sus precios hasta superar los 500.000, mientras que en Asia y América llegaríamos a 200.000 y 180.000 respectivamente para el último bin de su histograma. Además, podemos ver también una predisposición de los coches europeos a la polaridad, al observar 2 grupos claros entorno a los precios bajos y altos del histograma, cosa que también observamos en menor medida en los otros 2 continentes a los que pertenecen los fabricantes, aunque si bien es cierto que en estos otros 2 grupos la evolución de la distribución es más gradual y menos diferenciada en grupos.

En cuanto a valores extremos, observamos que América es la cual presenta valores más alejados de su 3º cuartil posicionándose entre 130.000 y 160.000, llegando a ser considerados estos valores

como extremos per se. En los otros 2 grupos, los valores cercanos al máximo si bien superan el 3º cuartil por una buena diferencia, no sería claro el considerarlos extremos al estar a una distancia todavía prudencial.

Por último, vemos que el separar los datos por continentes no nos da una mejora en las predicción (a excepción de Europa, cosa que puede tener relación con las diferencias en las correlaciones), siendo que si usamos el modelo general para la regresión logística, los resultados son sustancialmente mejores.

```
# Guardamos la data utilizada en su carpeta correspondiente
write.csv(dataset, file = '../data/dataset_NEW.csv')
write.csv(dataset_by_continent[['asia']],
          file = '../data/dataset_continent_asia.csv')
write.csv(dataset_by_continent[['america']],
          file = '../data/dataset_continent_america.csv')
write.csv(dataset_by_continent[['europe']],
          file = '../data/dataset_continent_europa.csv')
```