Práctica 2: limpieza, análisis y representación de los datos

Cristina Liánez López y Manuel Padrón Martínez

3 de junio, 2020

Contents

1. Detalles de la actividad

En esta actividad se elabora un caso práctico, consistente en el tratamiento de un conjunto de datos (en inglés, dataset), 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.

2. Resolución

2.1. Descripción del dataset

El conjunto de datos objeto de análisis se ha obtenido a partir de un dataset libre disponible en Kaggle. Este conjunto de datos incluye información sobre diferentes marcas de vehículos nuevos y usados a la venta en los EE.UU. Los datos se obtuvieron haciendo uso de la técnica de *Web Scraping.* Está constituido por 2499 vehículos (filas o registros), de los que se han analizado 13 características (columnas) de cada uno.

Las características analizadas en este dataset son:

- **x**: valor para identificar las filas. Comienza en 0.
- **price**: precio de venta del vehículo en $.
- **brand**: marca del vehículo.
- **model**: modelo del vehículo.
- **year**: año de la primera matriculación del vehículo.
- **Title_Status**: Esta característica incluye dos posibles valores: *clean title* que significa que el vehículo es apto para circular; o *salvage insurance* en caso de que no sea apto para circular debido a que está dañado por un accidente, inundación, incendio, o cualquier otra circunstancia.
- **Mileage**: kilometraje del vehículo, expresado en millas.
- **Color**: Color del vehículo.
- **Vin**: Número de bastidor. Compuesto por 17 caracteres (números y letras)

- **Lot**: es un número de identificación asignado a una cantidad determinada o un lote de coches de un solo fabricante. En este caso, se combina un número de lote con un número de serie para formar el número de identificación del vehículo.
- **State**: estado o ciudad donde se encuentra el vehículo.
- **Country**: país donde se encuentra el vehículo.
- **Condition**: tiempo que hace que se publicó el anuncio de venta del vehículo en la página web.

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

Preguntas a responder con el estudio:

- ¿Qué influye más en el precio de un vehículo con menos de 20 años de antigüedad: su antigüedad o el kilometraje que tenga?
- ¿Los coches de segunda mano de color blanco son más caros que los de color negro?
- ¿Podría crearse un modelo o fórmula para calcular el precio de venta de los vehículos de segunda mano, de manera objetiva, en función de ciertas características de los vehículos? ¿Cuáles serían las características más relevantes a tener en cuenta en esa fórmula?

2.3. Limpieza de los datos

En primer lugar, procedemos a realizar la lectura del fichero en formato CSV en el que se encuentran los datos. A continuación, examinaremos el tipo de datos con los que R ha interpretado cada variable.

```
# Carga del archivo
setwd("../csv")
cars <- read.csv("USA_cars_datasets.csv",header=TRUE)

#muestra las primeras filas del dataset
head(cars)
```

```
##   X price     brand   model year  title_status mileage  color
## 1 0  6300    toyota cruiser 2008 clean vehicle  274117  black
## 2 1  2899      ford      se 2011 clean vehicle  190552 silver
## 3 2  5350     dodge     mpv 2018 clean vehicle   39590 silver
## 4 3 25000      ford    door 2014 clean vehicle   64146   blue
## 5 4 27700 chevrolet    1500 2018 clean vehicle    6654    red
## 6 5  5700     dodge     mpv 2018 clean vehicle   45561  white
##                  vin       lot      state country     condition
## 1   jtezu11f88k007763 159348797 new jersey     usa  10 days left
## 2   2fmdk3gc4bbb02217 166951262  tennessee     usa   6 days left
## 3   3c4pdcgg5jt346413 167655728    georgia     usa   2 days left
## 4   1ftfw1et4efc23745 167753855   virginia     usa 22 hours left
## 5   3gcpcrec2jg473991 167763266    florida     usa 22 hours left
## 6   2c4rdgeg9jr237989 167655771      texas     usa   2 days left
```

```
#Examino el tipo de datos de cada variable
str(cars)
```

```
## 'data.frame':    2499 obs. of  13 variables:
##  $ X           : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ price       : int  6300 2899 5350 25000 27700 5700 7300 13350 14600 5250 ...
##  $ brand       : Factor w/ 28 levels "acura","audi",..: 28 9 8 9 6 8 6 10 6 9 ...
##  $ model       : Factor w/ 127 levels "1500","2500",..: 26 93 76 33 1 76 87 33 72 76 ...
##  $ year        : int  2008 2011 2018 2014 2018 2018 2010 2017 2018 2017 ...
```

```
##  $ title_status: Factor w/ 2 levels "clean vehicle",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ mileage     : num  274117 190552 39590 64146 6654 ...
##  $ color       : Factor w/ 49 levels "beige","billet silver metallic clearcoat",..: 3 40 40 5 35 47 3
##  $ vin         : Factor w/ 2495 levels "  19uua96529a004646",..: 2393 1666 1886 968 2088 1650 1194 11
##  $ lot         : int  159348797 166951262 167655728 167753855 167763266 167655771 167753872 16769249
##  $ state       : Factor w/ 44 levels "alabama","arizona",..: 25 36 8 40 7 37 8 4 7 37 ...
##  $ country     : Factor w/ 2 levels " canada"," usa": 2 2 2 2 2 2 2 2 2 2 ...
##  $ condition   : Factor w/ 47 levels "1 days left",..: 4 40 17 21 21 17 21 19 21 17 ...
```

```
#Miramos un resumen de los datos
summary(cars)
```

```
##        X              price             brand          model
##  Min.   :   0.0   Min.   :    0   ford     :1235   door   : 651
##  1st Qu.: 624.5   1st Qu.:10200   dodge    : 432   f-150  : 219
##  Median :1249.0   Median :16900   nissan   : 312   doors  : 148
##  Mean   :1249.0   Mean   :18768   chevrolet: 297   caravan: 102
##  3rd Qu.:1873.5   3rd Qu.:25556   gmc      :  42   mpv    :  87
##  Max.   :2498.0   Max.   :84900   jeep     :  30   fusion :  65
##                                   (Other)  : 151   (Other):1227
##       year                title_status     mileage            color
##  Min.   :1973   clean vehicle    :2336   Min.   :      0   white  :707
##  1st Qu.:2016   salvage insurance: 163   1st Qu.:  21466   black  :516
##  Median :2018                            Median :  35365   gray   :395
##  Mean   :2017                            Mean   :  52299   silver :300
##  3rd Qu.:2019                            3rd Qu.:  63472   red    :192
##  Max.   :2020                            Max.   :1017936   blue   :151
##                                                            (Other):238
##                   vin            lot                      state
##    1g1al58f787159241:   2   Min.   :159348797   pennsylvania  : 299
##    1gndt13s632267445:   2   1st Qu.:167625331   florida       : 246
##    1gnevhkw8jj148388:   2   Median :167745058   texas         : 214
##    3gcrkse37ag234620:   2   Mean   :167691389   california    : 190
##    19uua96529a004646:   1   3rd Qu.:167779772   michigan      : 169
##    19xfb2f81fe252000:   1   Max.   :167805500   north carolina: 146
##  (Other)            :2489                       (Other)       :1235
##     country            condition
##   canada:   7   2 days left  :832
##   usa   :2492   21 hours left:492
##                 3 days left  :137
##                 14 hours left:108
##                 1 days left  : 91
##                 8 days left  : 82
##                 (Other)      :757
```

Puede observarse que los tipos de datos asignados automáticamente por R a las variables se corresponden con el dominio de estas.

De las 13 características registradas de cada vehículo, se ha decido prescindir de **x**, **lot** y **condition**, ya que no son atributos propios de los vehículos, sino que hacen referencia a los anuncios en los que se publicitaban a los mismos.

```
# Prescindimos de las variables X, lot y condition
cars <- cars[,-(1)]
cars <- cars[,-(9)]
cars <- cars[,-(11)]
```

```r
str(cars)
```

```
## 'data.frame':    2499 obs. of  10 variables:
##  $ price       : int  6300 2899 5350 25000 27700 5700 7300 13350 14600 5250 ...
##  $ brand       : Factor w/ 28 levels "acura","audi",..: 28 9 8 9 6 8 6 10 6 9 ...
##  $ model       : Factor w/ 127 levels "1500","2500",..: 26 93 76 33 1 76 87 33 72 76 ...
##  $ year        : int  2008 2011 2018 2014 2018 2018 2010 2017 2018 2017 ...
##  $ title_status: Factor w/ 2 levels "clean vehicle",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ mileage     : num  274117 190552 39590 64146 6654 ...
##  $ color       : Factor w/ 49 levels "beige","billet silver metallic clearcoat",..: 3 40 40 5 35 47 3
##  $ vin         : Factor w/ 2495 levels "  19uua96529a004646",..: 2393 1666 1886 968 2088 1650 1194 11
##  $ state       : Factor w/ 44 levels "alabama","arizona",..: 25 36 8 40 7 37 8 4 7 37 ...
##  $ country     : Factor w/ 2 levels " canada"," usa": 2 2 2 2 2 2 2 2 2 2 ...
```

```r
head(cars)
```

```
##   price     brand   model year  title_status mileage  color                vin
## 1  6300    toyota cruiser 2008 clean vehicle  274117  black   jtezu11f88k007763
## 2  2899      ford      se 2011 clean vehicle  190552 silver   2fmdk3gc4bbb02217
## 3  5350     dodge     mpv 2018 clean vehicle   39590 silver   3c4pdcgg5jt346413
## 4 25000      ford    door 2014 clean vehicle   64146   blue   1ftfw1et4efc23745
## 5 27700 chevrolet    1500 2018 clean vehicle    6654    red   3gcpcrec2jg473991
## 6  5700     dodge     mpv 2018 clean vehicle   45561  white   2c4rdgeg9jr237989
##        state country
## 1 new jersey     usa
## 2  tennessee     usa
## 3    georgia     usa
## 4   virginia     usa
## 5    florida     usa
## 6      texas     usa
```

2.3.1. Normalización de variables

A continuación, mostraremos los valores de las variables cualitativas o categóricas mediante el uso de tablas de frecuencia. Ésto nos permitirá saber si hay valores fuera del rango o valores extraños en ellas.

```r
#variables cualitativas
table(cars$brand)
```

```
##
##           acura            audi             bmw           buick        cadillac
##               3               4              17              13              10
##       chevrolet        chrysler           dodge            ford             gmc
##             297              18             432            1235              42
## harley-davidson       heartland           honda         hyundai        infiniti
##               1               5              12              15              12
##          jaguar            jeep             kia            land           lexus
##               1              30              13               4               2
##         lincoln        maserati           mazda   mercedes-benz          nissan
##               2               1               2              10             312
##       peterbilt             ram          toyota
##               4               1               1
```

```r
table(cars$model)
```

```
##
##        1500        2500      2500hd         300        3500           5
```

```
##          39           8           1           6           4           1
##      acadia      altima      armada         bus         cab      camaro
##           1          21           4           2           8           6
##     caravan       cargo  challenger     charger     chassis    cherokee
##         102           2          44          42           4           3
##    colorado     compass     connect convertible    corvette     country
##          12           6           1           1           4           2
##       coupe     cruiser       cruze     cutaway        cx-3           d
##           6           1           2          12           1           2
##        dart   discovery        door       doors          dr         drw
##           1           1         651         148           1          10
##     durango     e-class    ecosport        edge          el     elantra
##          64           1           7          34           3           1
##     enclave      encore      energi     equinox      escape         esv
##           2           3           1          18          39           1
##  expedition    explorer       f-150       f-650       f-750      fiesta
##          28          39         219           3           1          14
##        flex       focus       forte    frontier      fusion      ghibli
##          33           9           2          14          65           1
##         glc         gle          gx      hybrid      impala     journey
##           1           2           1           5          12          61
##        juke       kicks          ld     limited           m      malibu
##           1           1           3           1           1          12
##         max      maxima         mdx         mpv      murano     mustang
##          41           3           1          87           5          29
##    nautilus        note         nvp    pacifica   passenger  pathfinder
##           1           2           1           3           3          22
##      pickup     pioneer          pk          q5         q70      ranger
##          15           1           5           1           1           6
## road/street       rogue          se       sedan      sentra      series
##           1          54           1           4          28           8
##    sl-class       sonic     sorento        soul       spark       sport
##           1           2           5           1           1          40
##    sportage         srw         srx    suburban    sundance         suv
##           1          38           1          16           1           1
##       tahoe      taurus       titan       trail     transit    traverse
##           9          10           5           1          41           6
##        trax       truck     utility         van        vans      vehicl
##           8           4           1          46           2           1
##       versa        volt       wagon          x3          xd         xt5
##          34           2          30           2           1           1
##      xterra
##           1
```

```
table(cars$title_status)
```

```
##
##     clean vehicle salvage insurance
##              2336               163
```

```
table(cars$color)
```

```
##
##                                  beige        billet silver metallic clearcoat
##                                      5                                       3
##                                  black                       black clearcoat
```

```
##                                   516                                     2
##                                  blue                   bright white clearcoat
##                                   151                                     2
##                                 brown                                burgundy
##                                    15                                     1
##                           cayenne red                                charcoal
##                                     2                                    18
##                                color:                   competition orange
##                                     5                                     1
##                              dark blue                         glacier white
##                                     1                                     1
##                                  gold                                 gray
##                                    19                                   395
##                                 green                                guard
##                                    24                                     1
##                           ingot silver                  ingot silver metallic
##                                     1                                     4
##                      jazz blue pearlcoat                     kona blue metallic
##                                     1                                     1
##                             light blue                         lightning blue
##                                     1                                     1
##                       magnetic metallic                                maroon
##                                     6                                     1
##                         morningsky blue                              no_color
##                                     1                                    61
##                              off-white                                orange
##                                     2                                    20
##                           oxford white                           pearl white
##                                     4                                     1
##                          phantom black                                purple
##                                     1                                     1
##                                   red royal crimson metallic tinted clearcoat
##                                   192                                     1
##                              ruby red      ruby red metallic tinted clearcoat
##                                     1                                     2
##                           shadow black                                silver
##                                     5                                   300
##                            super black                                   tan
##                                     3                                     1
##                           toreador red                  triple yellow tri-coat
##                                     1                                     3
##                              turquoise                   tuxedo black metallic
##                                     1                                     2
##                                 white        white platinum tri-coat metallic
##                                   707                                     2
##                                yellow
##                                     9
```

```
table(cars$state)
```

```
##
##        alabama        arizona       arkansas     california       colorado
##             17             33             12            190             21
##    connecticut        florida        georgia          idaho       illinois
##             25            246             51              2            113
```

```
##        indiana         kansas       kentucky      louisiana       maryland
##             14              4              9             11              4
##  massachusetts       michigan      minnesota    mississippi       missouri
##             27            169            119             24             46
##        montana       nebraska         nevada  new hampshire     new jersey
##              1              4             85              4             87
##     new mexico       new york north carolina           ohio       oklahoma
##              4             58            146             31             71
##        ontario         oregon   pennsylvania   rhode island south carolina
##              7             27            299              2             64
##      tennessee          texas           utah        vermont       virginia
##             26            214             10              2             90
##     washington  west virginia      wisconsin        wyoming
##             14             21             94              1
```

```
table(cars$country)
```

```
##
##  canada     usa
##       7    2492
```

Los colores se van a clasificar en los siguientes valores: *beige, black, blue, brown, orange, gold, red, silver, white, gray, green, purple, yellow, no-color*

```
cars$color <- str_replace(cars$color, ".*beige.*", "beige")
cars$color <- str_replace(cars$color, ".*black.*", "black")
cars$color <- str_replace(cars$color, ".*blue.*", "blue")
cars$color <- str_replace(cars$color, ".*brown.*", "brown")
cars$color <- str_replace(cars$color, ".*orange.*", "orange")
cars$color <- str_replace(cars$color, ".*gold.*", "gold")
cars$color <- str_replace(cars$color, ".*red.*", "red")
cars$color <- str_replace(cars$color, ".*silver.*", "silver")
cars$color <- str_replace(cars$color, ".*white.*", "white")
cars$color <- str_replace(cars$color, ".*gray.*", "gray")
cars$color <- str_replace(cars$color, ".*green.*", "green")
cars$color <- str_replace(cars$color, ".*purple.*", "purple")
cars$color <- str_replace(cars$color, ".*yellow.*", "yellow")
cars$color <- str_replace(cars$color, "burgundy", "red")
cars$color <- str_replace(cars$color, "charcoal", "black")
cars$color <- str_replace(cars$color, "color:", "no_color")
cars$color <- str_replace(cars$color, "maroon", "brown")
cars$color <- str_replace(cars$color, "magnetic metallic", "black")
cars$color <- str_replace(cars$color, "royal crimson metallic tinted clearcoat", "purple")
cars$color <- str_replace(cars$color, "turquoise", "blue")
cars$color <- str_replace(cars$color, "tan", "beige")
cars$color <- str_replace(cars$color, "guard", "black")

table(cars$color)
```

```
##
##    beige    black     blue    brown     gold     gray    green no_color
##        6      554      158       16       19      395       24       66
##   orange   purple      red   silver    white   yellow
##       21        2      199      308      719       12
```

2.3.2. Valores perdidos

¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

```
# valores perdidos(0 o vacíos)
#Con la siguiente instrucción vemos si hay registros que están incompletos
complete.cases(cars)
```

```
##    [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##   [15] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##   [29] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##   [43] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##   [57] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##   [71] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##   [85] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##   [99] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [113] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [127] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [141] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [155] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [169] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [183] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [197] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [211] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [225] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [239] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [253] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [267] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [281] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [295] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [309] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [323] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [337] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [351] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [365] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [379] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [393] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [407] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [421] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [435] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [449] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [463] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [477] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [491] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [505] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [519] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [533] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [547] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [561] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [575] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [589] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [603] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [617] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [631] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [645] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [659] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [673] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
##  [687] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [701] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [715] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [729] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [743] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [757] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [771] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [785] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [799] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [813] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [827] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [841] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [855] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [869] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [883] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [897] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [911] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [925] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [939] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [953] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [967] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [981] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [995] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1009] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1023] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1037] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1051] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1065] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1079] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1093] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1107] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1121] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1135] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1149] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1163] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1177] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1191] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1205] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1219] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1233] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1247] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1261] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1275] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1289] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1303] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1317] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1331] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1345] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1359] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1373] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1387] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1401] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1415] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1429] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
## [1443] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1457] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1471] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1485] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1499] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1513] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1527] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1541] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1555] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1569] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1583] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1597] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1611] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1625] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1639] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1653] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1667] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1681] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1695] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1709] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1723] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1737] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1751] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1765] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1779] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1793] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1807] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1821] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1835] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1849] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1863] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1877] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1891] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1905] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1919] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1933] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1947] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1961] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1975] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [1989] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2003] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2017] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2031] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2045] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2059] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2073] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2087] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2101] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2115] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2129] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2143] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2157] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2171] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2185] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
## [2199] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2213] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2227] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2241] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2255] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2269] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2283] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2297] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2311] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2325] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2339] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2353] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2367] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2381] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2395] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2409] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2423] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2437] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2451] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2465] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2479] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2493] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
#Analizamos valores perdidos
sapply(cars, function(x) sum(is.na(x)))
```

```
##        price        brand        model         year title_status      mileage
##            0            0            0            0            0            0
##        color          vin        state      country
##            0            0            0            0
```

```
sapply(cars, function(x) sum(x == 0))
```

```
##        price        brand        model         year title_status      mileage
##           43            0            0            0            0            6
##        color          vin        state      country
##            0            0            0            0
```

En primer lugar, hemos comprobado si hay registros incompletos, es decir, en los que en alguno de sus atributos no se haya introducido valor. Con la instrucciones ejecutada, se comprueba que no hay ninguno, ya que no se ha obtenido ningún valor a **FALSE**.

En segundo lugar, se ha analizado en cada una de las variables cuantitativas si existen valores almacenados equivalentes a 0 o NA. En el caso del precio, se han detectado 43 registros cuyo precio es 0; y para el caso del kilometraje, se han encontrado 6 registros con este mismo valor.

En el caso de la variable *price*, son claramente valores perdidos ya que no tiene sentido que el precio de venta fijado sea de 0$ cuando la naturaleza de los anuncios es la venta de los vehículos. Será necesario imputar los valores de estas variables en estos registros.

En el caso de la variable *mileage* cuyo valor es 0, se ha mirado el valor del atributo *year*, puesto que si éste valor se corresponde con coches del 2020, el valor registrado en el atributo *mileage* puede ser correcto ya que se trataría de vehículos nuevos que no han recorrido ninguna milla aún.

```
years <- subset(cars$year, subset = cars$mileage == 0)
print(years)
```

```
## [1] 2004 1994 2012 1993 2013 2017
```

Tras realizar la comprobación, vemos que no es así en ninguno de los casos, es decir, son vehículos con cierta antigüedad, por lo que el valor a 0 de *mileage* se corresponde con un valor perdido, que deberíamos de imputar.

Para la imputación de los valores perdidos se empleará un método basado en la similitud o diferencia entre los registros: la imputación basada en k vecinos más próximos (en inglés, kNN-imputation). La elección de esta alternativa se realiza bajo la hipótesis de que nuestros registros guardan cierta relación. No obstante, es mejor trabajar con datos "aproximados" que con los propios elementos vacíos, ya que obtendremos análisis con menor margen de error.

```
#primero hay que sustituir los valores 0 por NA
cars$price <- ifelse(cars$price == 0, NA, cars$price)
cars$mileage <- ifelse(cars$mileage == 0, NA, cars$mileage)
sapply(cars, function(x) sum(is.na(x)))
```

```
##        price        brand        model         year title_status      mileage
##           43            0            0            0            0            6
##        color          vin        state      country
##            0            0            0            0
```

```
#imputamos los valores NA usando el método kNN
cars$price <- kNN(cars)$price
cars$mileage <- kNN(cars)$mileage

#comprobamos
sapply(cars, function(x) sum(is.na(x)))
```

```
##        price        brand        model         year title_status      mileage
##            0            0            0            0            0            0
##        color          vin        state      country
##            0            0            0            0
```

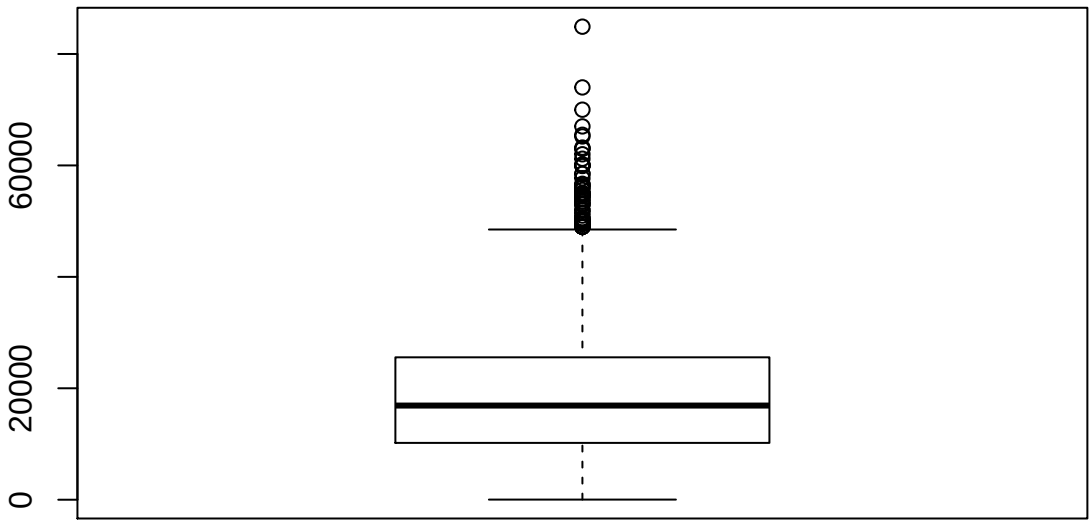
```
sapply(cars, function(x) sum(x == 0))
```

```
##        price        brand        model         year title_status      mileage
##            0            0            0            0            0            0
##        color          vin        state      country
##            0            0            0            0
```

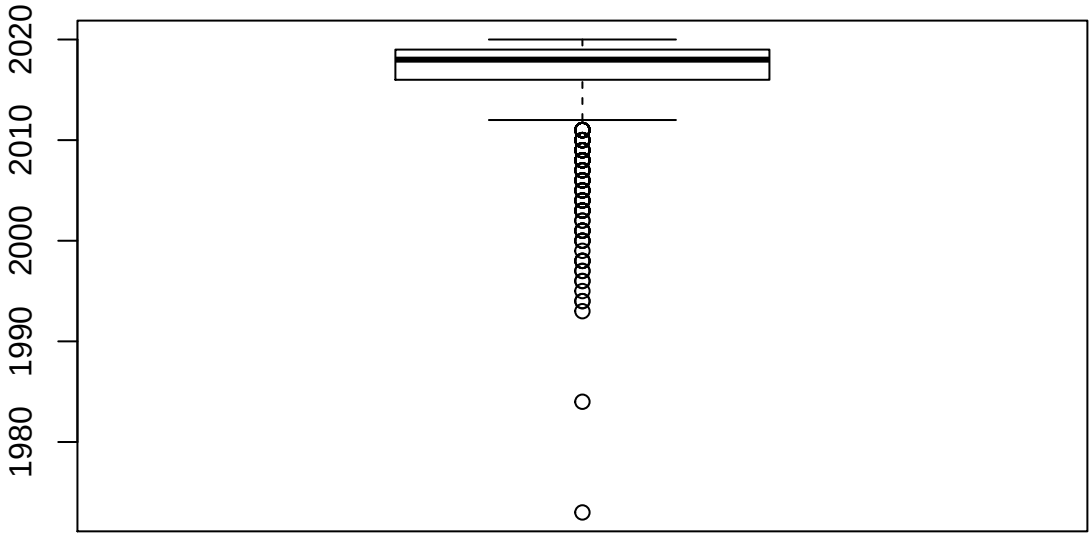
2.3.3. Valores extremos.

Los valores extremos o outliers son aquellos que parecen no ser congruentes si los comparamos con el resto de los datos. Para identificarlos, utilizaremos la representación mediante un diagrama de caja.

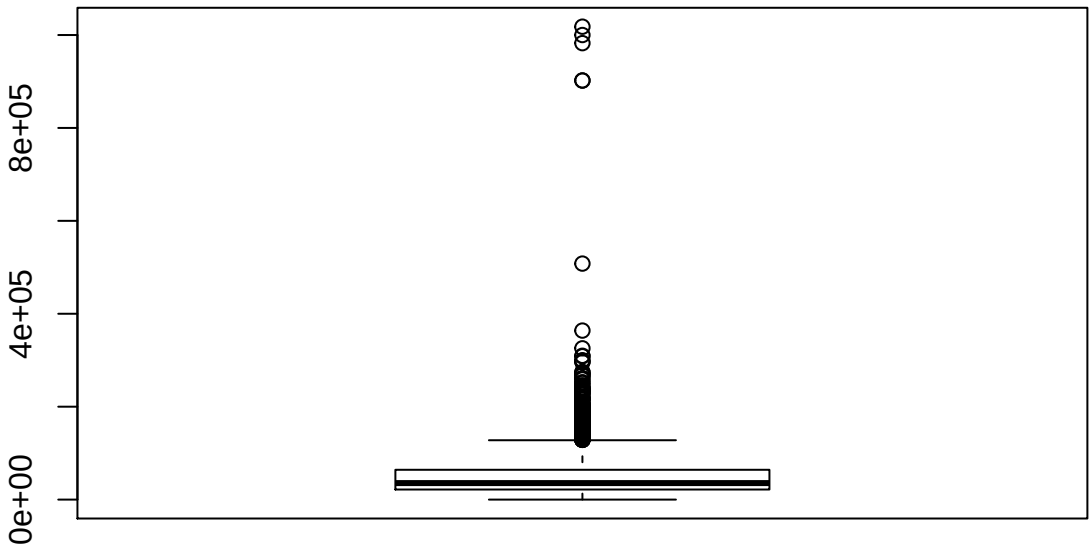
```
boxplot(cars$price)
```




```
boxplot(cars$year)
```




```
boxplot(cars$mileage)
```




```
summary(cars$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      25   10200   16900   18787   25556   84900
```

```
summary(cars$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1973    2016    2018    2017    2019    2020
```

```
summary(cars$mileage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1   21686   35493   52966   64140 1017936
```

En los tres diagramas anteriores se observan bastantes valores outliers. Ésto es debido a que el rango de valores de las tres variables es bastante amplio. Si revisamos los datos y los comparamos con los valores resumen de cada variable, llegamos a la conclusión de que son valores posibles. Por tanto, se mantendrán tal y como están recogidos.

2.4. Análisis de los datos

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

A continuación, se seleccionan los grupos dentro de nuestro conjunto de datos que pueden resultar interesantes para analizar y/o comparar.

```
#Agrupar por año de primera matriculación, diferenciando los que tienen
#una antigüedad de 20 años o más.

year <- subset(cars$year, subset = cars$year >= 2000)
price <- subset(cars$price, subset = cars$year >= 2000)
mileage <- subset(cars$mileage, subset = cars$year >= 2000)
cochesMenos20 <- data.frame(price, year, mileage)
#head(cochesMenos20)

year <- subset(cars$year, subset = cars$year < 2000)
price <- subset(cars$price, subset = cars$year < 2000)
mileage <- subset(cars$mileage, subset = cars$year < 2000)
```

```r
cochesMas20 <- data.frame(price, year, mileage)
#head(cochesMas20)


#Agrupar coches por color blanco o negro.

precioCochesBlancos <- subset(cars$price, subset = cars$color == "white")
precioCochesNegros <- subset(cars$price, subset = cars$color == "black")
```

2.4.2. Comprobación de la normalidad y homogeneidad de la varianza.

Vamos a comprobar la suposición de normalidad y la homogeneidad de la varianza de las variables. Para ello, vamos a usar el test de Kolmogorov-Smirnov y luego lo contrastaremos con el test de Shapiro-Wilk. Con esta prueba si obtenemos un p-value mayor a 0,05 asumiremos que los datos siguen una distribución normal.

```r
#Comprobamos la normalidad de la variable año
ks.test(as.array(unique(cars$year)), pnorm, mean(as.array(unique(cars$year))),
        sd(as.array(unique(cars$year))))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  as.array(unique(cars$year))
## D = 0.076426, p-value = 0.9891
## alternative hypothesis: two-sided
```

```r
shapiro.test(as.array(cars$year))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  as.array(cars$year)
## W = 0.67439, p-value < 2.2e-16
```

```r
#Comprobamos la normalidad de la variable precio
ks.test(as.array(unique(cars$price)), pnorm, mean(as.array(unique(cars$price))),
        sd(as.array(unique(cars$price))))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  as.array(unique(cars$price))
## D = 0.08809, p-value = 9.617e-06
## alternative hypothesis: two-sided
```

```r
shapiro.test(as.array(cars$price))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  as.array(cars$price)
## W = 0.94933, p-value < 2.2e-16
```

```r
#Comprobamos la normalidad de la variable kilometraje
ks.test(as.array(unique(cars$mileage)), pnorm, mean(as.array(unique(cars$mileage))),
        sd(as.array(unique(cars$mileage))))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  as.array(unique(cars$mileage))
## D = 0.19944, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
shapiro.test(as.array(cars$mileage))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  as.array(cars$mileage)
## W = 0.55487, p-value < 2.2e-16
```

```
#Comprobamos la normalidad de la variable precioCochesBlancos
ks.test(as.array(unique(precioCochesBlancos)), pnorm,
        mean(as.array(unique(precioCochesBlancos))),
        sd(as.array(unique(precioCochesBlancos))))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  as.array(unique(precioCochesBlancos))
## D = 0.071149, p-value = 0.04224
## alternative hypothesis: two-sided
```

```
shapiro.test(as.array(precioCochesBlancos))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  as.array(precioCochesBlancos)
## W = 0.96784, p-value = 1.812e-11
```

```
#Comprobamos la normalidad de la variable precioCochesNegros
ks.test(as.array(unique(precioCochesNegros)), pnorm,
        mean(as.array(unique(precioCochesNegros))),
        sd(as.array(unique(precioCochesNegros))))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  as.array(unique(precioCochesNegros))
## D = 0.084059, p-value = 0.01505
## alternative hypothesis: two-sided
```

```
shapiro.test(as.array(precioCochesNegros))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  as.array(precioCochesNegros)
## W = 0.94011, p-value = 3.853e-14
```

Como vemos, a excepción de la variable año, las otras variables claramente no cumplen con la normalidad. En el caso de la variable año, debido al teorema central del límite, podemos considerar que tenderá a comportarse como una distribución normal.

También vamos a hacer un análisis de homocedasticidad entre las variables precioCochesBlancos y precioCoches negros que son las que vamos a comprobar más adelante. Como nos han fallado los test de normalidad vamos a usar el test de Fligner-Killeen. Para usar este test crearemos un nuevo dataframe donde recogeremos los precios de los coches blancos y negros y los evaluaremos.

```
df <- data.frame(price=c(precioCochesBlancos,precioCochesNegros),
                 color=c(rep(1,each=719),rep(2,each=554)))
fligner.test(price ~ color, data = df)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  price by color
## Fligner-Killeen:med chi-squared = 3.6859, df = 1, p-value = 0.05487
```

Nos da un valor mayor que 0,05 por lo que se acepta la hipótesis nula de homocedasticidad. Con lo que podremos usar un test t mas adelante entre estas dos variables.

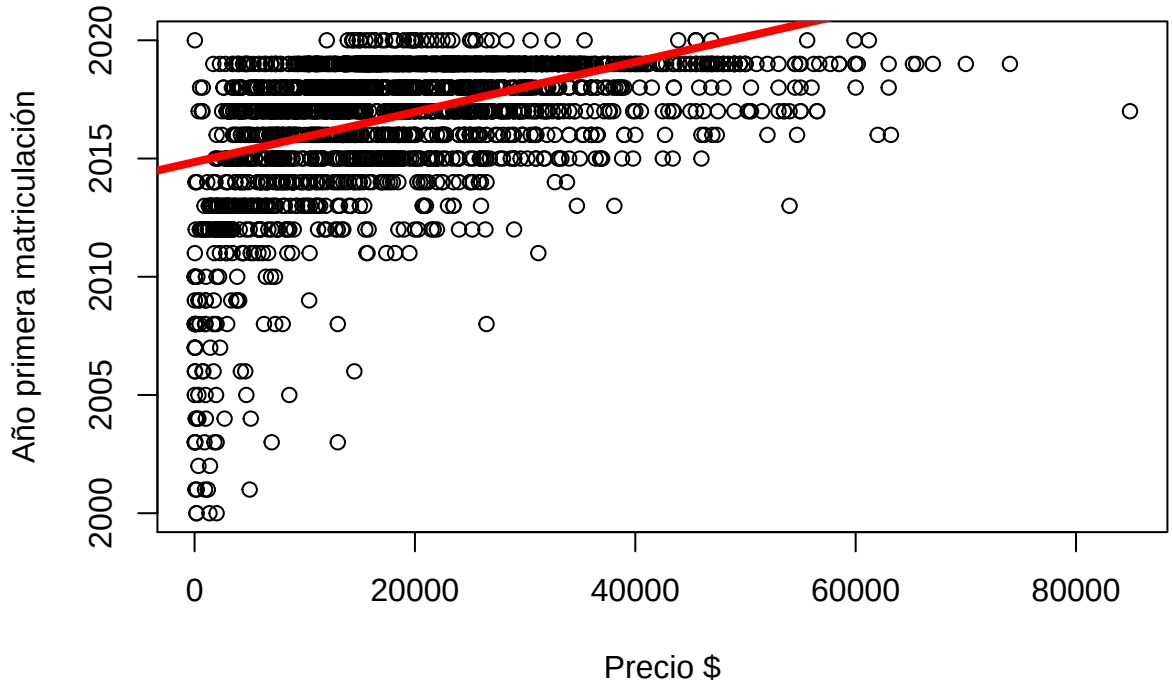
2.4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos.

a). Estudiar visualmente y analíticamente las posibles correlaciones entre:

- las variables precio y el año de la primera matriculación.
- las variables precio y el kilometraje.

En los dos casos anteriores, solo se van a tener en cuenta los vehículos con una antigüedad inferior a 20 años, es decir, cuya fecha de primera matriculación sea igual o posterior al 2000.

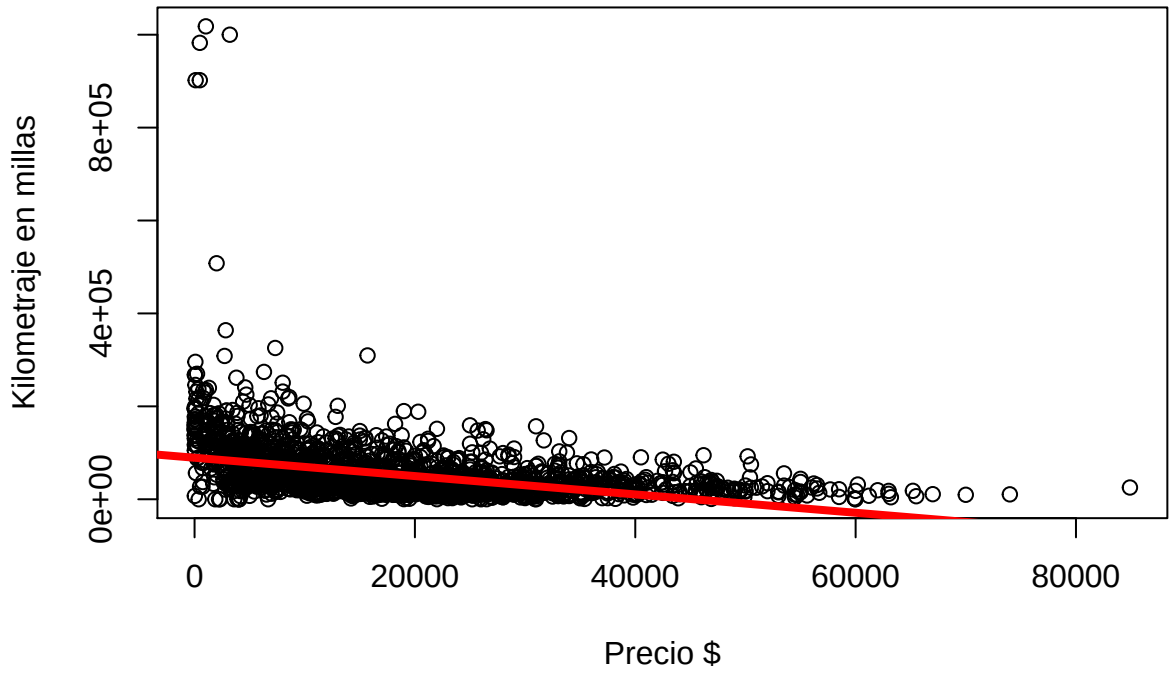
```
#price y year

plot(cochesMenos20$price, cochesMenos20$year, xlab = "Precio $",
     ylab = "Año primera matriculación")
abline(lm(cochesMenos20$year~cochesMenos20$price),col="red",lwd=4)
```



```r
cor(x=cochesMenos20$price, y=cochesMenos20$year,method="spearman")
```

```
## [1] 0.4690157
```

```r
#price y mileage
plot(cochesMenos20$price, cochesMenos20$mileage, xlab = "Precio $",
     ylab = "Kilometraje en millas")
abline(lm(cochesMenos20$mileage~cochesMenos20$price),col="red",lwd=4)
```




```r
cor(x=cochesMenos20$price, y=cochesMenos20$mileage,method="spearman")
```

```
## [1] -0.5342928
```

Del análisis anterior se extrae que existe una correlación positiva entre las variables precio y año de primera matriculación. Es decir, cuanto mayor es el año de primera matriculación, más nuevo es el coche, mayor es el precio.

En cuanto a la relación entre las variables precio y kilometraje, ésta es negativa ya que un mayor kilometraje del vehículo, influye bajando el precio de venta del mismo.

Mediante la función cor(), que utiliza el coeficiente de correlación de Spearman, se observa que en ambos casos, la relación entre los pares de variables puede considerarse de fortaleza media, siendo mayor para la relación entre el precio y el kilometraje.

b). Contraste de hipótesis

Si hay un color que manda actualmente en el mercado de ocasión, ese es el blanco y no otros pigmentos que, tradicionalmente, han levantado más pasiones, como el negro y el rojo. Su espectacular repunte se puede atribuir a que el blanco es una pintura más económica pero, actualmente, la demanda supera ligeramente a la oferta, de modo que, ¿influirá el color del coche en el precio de venta en vehículos de segunda mano?

La siguiente prueba consistirá en un contraste de hipótesis sobre dos muestras para determinar si el precio del coche es superior si éste es color blanco a si lo es negro. Para ello, tendremos dos muestras: la primera de ellas se corresponderá con los precios de los coches de color blanco y, la segunda, con aquellos que presentan el color negro.

Por aplicación del Teorema del límite central, para muestras con tamaño superior a 30, se puede suponer que los datos son normales. Como en este caso, n > 30, el contraste de hipótesis siguiente es válido.

Se plantea el siguiente **contraste de hipótesis de dos muestras sobre la diferencia de medias**:

$H_0 : \mu_1 = \mu_2$

$H_1 : \mu_1 > \mu_2$

donde μ_1 es la media de la población de la que se extrae la primera muestra donde el color de los coches es blanco y μ_2 es la media de la población de la que extrae la segunda donde el color de los coches es el negro.

Se trata de un test unilateral. Consideramos el nivel de significación $\alpha = 0.05$

```
#Contraste de hipótesis color blanco Vs negro

t.test(precioCochesBlancos, precioCochesNegros, alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  precioCochesBlancos and precioCochesNegros
## t = -0.96319, df = 1093.8, p-value = 0.1678
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 473.8822
## sample estimates:
## mean of x mean of y
##  19500.14  20168.36
```

Dado que obtenemos un p-valor mayor que el nivel de significación $pValor > \alpha$, entonces no rechazamos la hipótesis nula. Por tanto, podemos concluir que el precio de un coche de color blanco no tiene por qué ser mayor al precio de un coche de color negro.

c). Modelo de regresión

A continuación, se va a crear un modelo o fórmula para predecir el precio de venta de los vehículos de segunda mano en función de ciertas características de los mismos. Así, se calcularán varios modelos de regresión lineal utilizando regresores cuantitativos, en el primero de ellos, y se añadirán regresores cualitativos a los anteriores, en los posteriores modelos, con los que poder realizar las predicciones de los precios.

En primer lugar, se estimará por mínimos cuadrados ordinarios un modelo lineal que explique la variable precio del vehículo en función del año de primera matriculación y el kilometraje.

En los siguientes modelos de regresión lineal múltiple, se utilizarán regresores cuantitativos, los mismos que se han utilizado en el primer modelo, y se añadirán regresores cualitativos.

De entre los modelos que obtengamos, escogeremos el mejor utilizando como criterio aquel que presente un mayor coeficiente de determinación R^2.

```
#predecir precio en función del año de matriculación y kilometraje
modelo1 <- lm(formula = price ~ year + mileage, data = cars)

#predecir precio en función del año de matriculación, kilometraje y marca
modelo2 <- lm(formula = price ~ year + mileage + brand, data = cars)

#predecir precio en función del año de matriculación, kilometraje, marca y color
modelo3 <- lm(formula = price ~ year + mileage + brand + color, data = cars)

#Tabla con los coeficientes de determinación
coeficientes <- matrix(c( 1, summary(modelo1)$r.squared,
```

```
                        2, summary(modelo2)$r.squared,
                        3, summary(modelo3)$r.squared),
                        ncol = 2, byrow = TRUE)
print(coeficientes)
```

```
##      [,1]      [,2]
## [1,]    1 0.2073647
## [2,]    2 0.3588051
## [3,]    3 0.3651046
```

Podemos decir, que el tercer modelo es el más conveniente dado que tiene un mayor coeficiente de determinación. Empleando este modelo y haciendo uso de la **función predict()**, podemos realizar predicciones de precios de vehículos a partir del año de matriculación, kilometraje, marca y color, pero como los coeficientes son muy bajos esta predicción no será correcta.

Para comprobar lo efectiva que sería la predicción, vamos a calcular el error sobre el conjunto de datos que tenemos:

```
# Usamos la función predict con el modelo que más se nos ajusta para evaluar todo
# nuestro conjunto de datos
prediccion = predict(modelo3,cars)

# Comprobamos cuantos coches se diferencian en más de 1000$ respecto al resultado estimado
summary(abs(prediccion-cars$price)>1000)
```

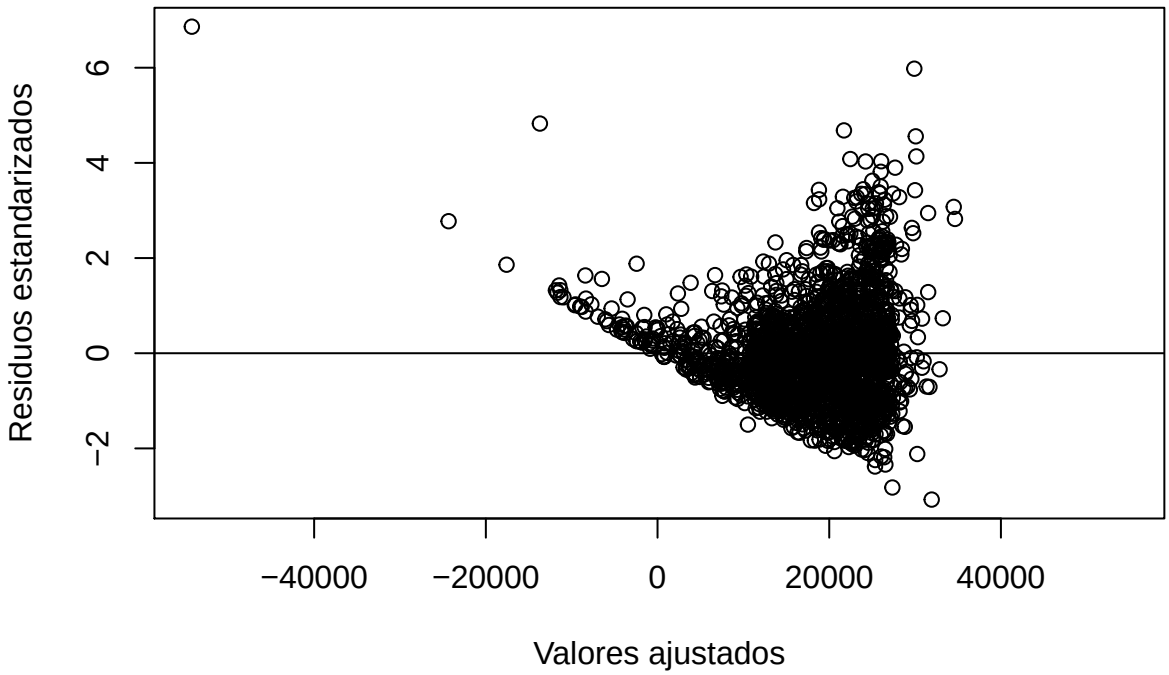
```
##    Mode   FALSE    TRUE
## logical     235    2264
```

Como vemos, 2264 coches, el 98%, de los coches de nuestra muestra distan más de 1000 dolares de su precio real.

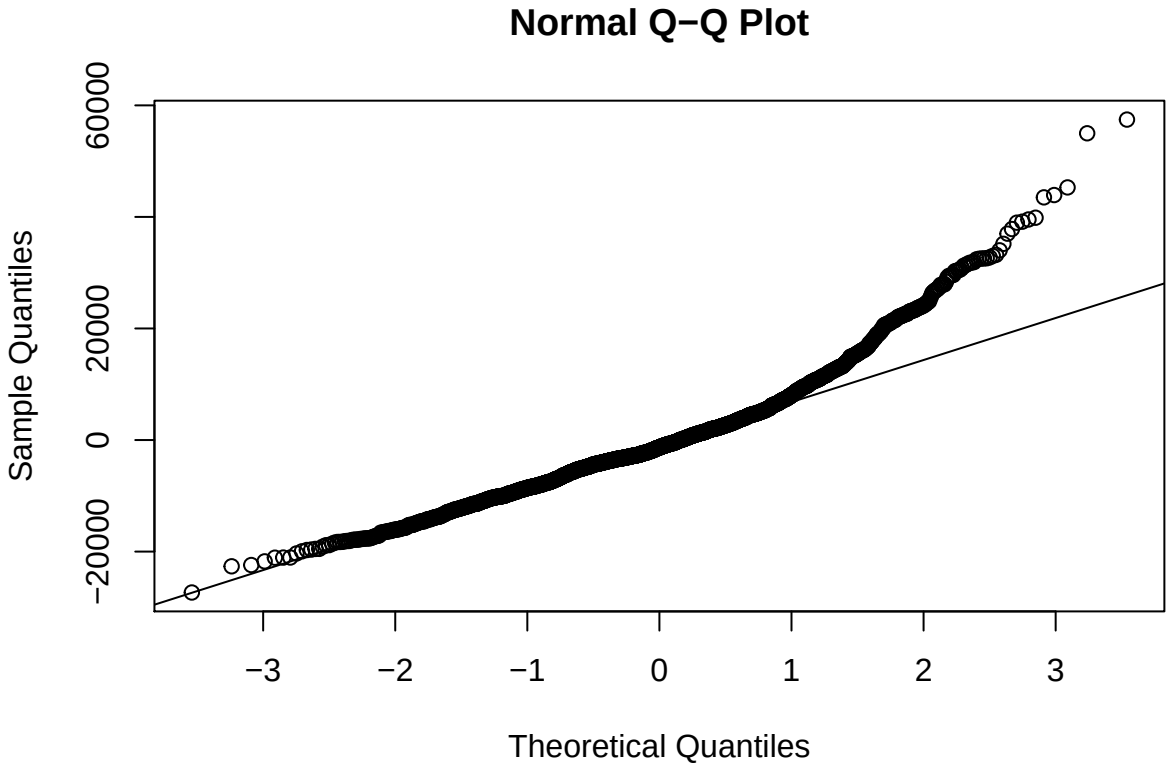
2.5. Representación de los resultados a partir de tablas y gráficas.

Para la diagnosis del modelo de regresión lineal múltiple escogido se harán dos gráficos: uno con los valores ajustados frente a los residuos (que nos permitirá ver si la varianza es constante) y el gráfico cuantil-cuantil que compara los residuos del modelo con los valores de una variable que se distribuye normalmente(QQ plot).

```
#Varianza de los errores/residuos constante
plot(fitted.values(modelo3),rstandard(modelo3), xlab="Valores ajustados",
     ylab="Residuos estandarizados")
abline(h=0)
```




```r
#Distribución normal de los residuos
qqnorm(modelo3$residuals)
qqline(modelo3$residuals)
```



El **gráfico de los residuos en función de los valores ajustados por el modelo** permite evaluar 3 cuestiones principalmente:

- Si has utilizado el tipo de relación adecuada, es decir, si el modelo debería ser no lineal en lugar de lineal. Si el tipo de modelo que utilizaste no es el adecuado encontrarás sesgos o tendencias en los

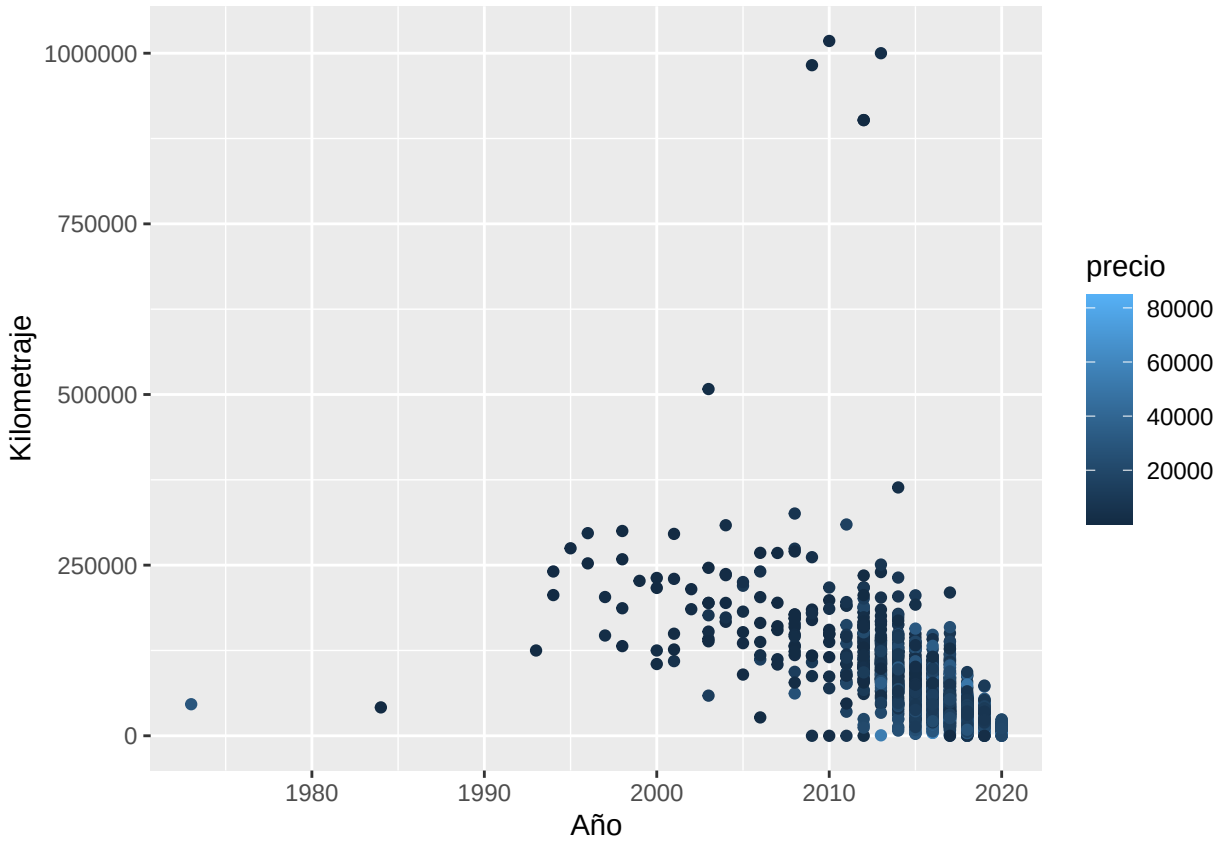
residuos.

- Si la varianza es constante o por el contrario tienes problemas de dispersión irregular. Los residuos deben distribuirse al azar alrededor del valor cero.
- Si existen datos extremos (outliers) que puedan perturbar e invalidar tu modelo.

El **gráfico cuantil-cuantil (Normal Q-Q)** permite comparar la distribución de los residuos con la distribución normal teórica. Por lo tanto, si los residuos tienen una distribución normal deberían seguir aproximadamente la línea recta diagonal en el gráfico Q-Q normal, en caso contrario los residuos se van a apartar de la diagonal.

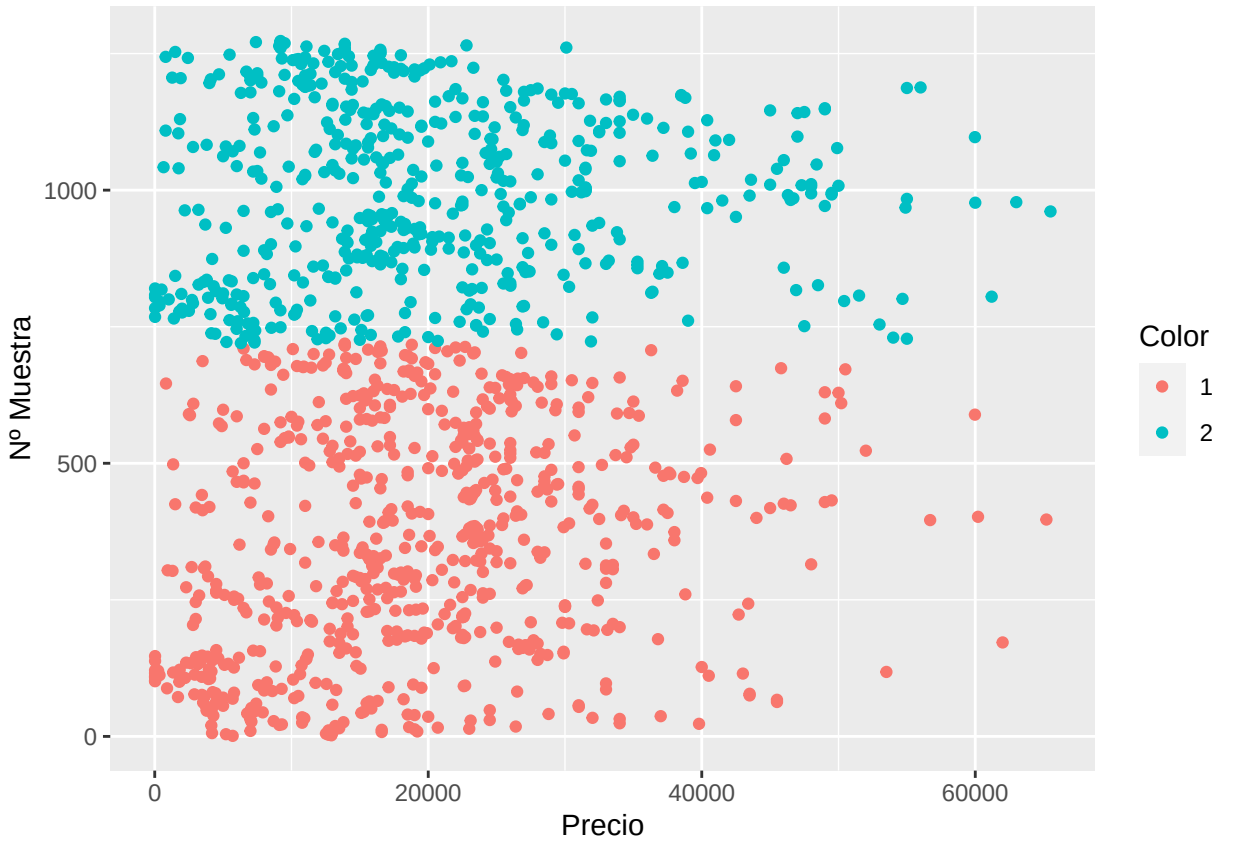
De las gráficas anteriores se puede extraer la presencia de outliers que están perturbando el modelo y valores alejados del comportamiento de la mayoría de los puntos (sobre todo en valores pequeños y altos de x).

```
ggplot(cars,aes(x=year,y=mileage, colour=price))+geom_point()+
 labs(x="Año", y="Kilometraje", colour="precio")
```



En la **gráfica de puntos**, vemos como, incluso visualmente se puede apreciar que mientras menos kilómetros tiene el coche más caro es.

```
ggplot(df,aes(x=price,y=1:nrow(df),colour=as.factor(color)))+geom_point()+
  labs(x="Precio",y="Nº Muestra", colour="Color")
```



En esta segunda **gráfica de puntos** podemos observar como los coches blancos, etiquetados como 1, y los coches negros, etiquetados como 2, están distribuidos en los mismos rangos de precios, tal como corrobora el estudio.

2.6. Conclusiones

Finalmente, a partir de todo el estudio realizado, anotamos las siguientes conclusiones dando respuesta a las preguntas inicialmente planteadas.

- ¿Qué influye más en el precio de un vehículo con menos de 20 años de antigüedad: su antigüedad o el kilometraje que tenga?

Pues parece ser, que el kilometraje del vehículo tiene una repercusión mayor en el precio que la antigüedad de éste.

- ¿Los coches de color blanco son más caros que los de color negro?

En este estudio, y con un 95% de nivel de confianza, el precio de los coches de color blanco es similar al precio de los coches de color negro.

- ¿Podría crearse un modelo o fórmula para calcular el precio de venta de los vehículos de segunda mano, de manera objetiva, en función de ciertas características de los vehículos? ¿Cuáles serían las características más relevantes a tener en cuenta en esa fórmula?

Teniendo en cuenta que los valores de los coeficientes de la regresión nos han dado muy cercanos a cero, con este tipo de regresión no podemos realizar predicciones de precios de vehículos a partir del año de matriculación, kilometraje, marca y color.

2.7. Crear el archivo procesado.

```
write.csv2(cars, file = "USA_cars_datasets_processed.csv")
```

3. Recursos

1. Calvo M., Subirats L., Pérez D. (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.
2. Dalgaard, P. (2008). Introductory statistics with R. Springer Science & Business Media.
3. Megan Squire (2015). Clean Data. Packt Publishing Ltd.
4. Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
5. Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
6. Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc.
7. Vegas, E. (2017). Preprocesamiento de datos. Material UOC.
8. Gibergans, J. (2017). Regresión lineal múltiple. Material UOC.
9. Rovira, C. (2008). Contraste de hipótesis. Material UOC.

4. Tabla de contribuciones al trabajo

CONTRIBUCIONES	FIRMA
Investigación previa	CLL, MPM
Redacción de respuestas	CLL, MPM
Desarrollo código	CLL, MPM