

# Análisis - Compra-venta de coches usados en UK

Marta Gómez / Juan Fco Nieto

12/17/2021

## Contents

<b>1 Descripción del dataset</b>	<b>2</b>
1.1 Objetivos y descripción del dataset original . . . . .	2
<b>2 Integración y selección de los datos de interés a analizar</b>	<b>2</b>
2.1 Descripción de las variables . . . . .	3
<b>3 Limpieza de los datos</b>	<b>4</b>
3.1 Análisis de valores nulos o vacíos . . . . .	4
3.2 Análisis de valores atípicos . . . . .	5
<b>4 Análisis de los datos.</b>	<b>8</b>
4.1 Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar). . . . .	8
<b>5 Análisis. Caso práctico compra-venta</b>	<b>10</b>

```
# https://cran.r-project.org/web/packages/ggplot2/index.html
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')

# https://cran.r-project.org/web/packages/dplyr/index.html
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')
if (!require('tidyverse')) install.packages('tidyverse'); library('tidyverse')

# Comprobación de media winsor
if (!require('skimr')) install.packages('skimr'); library('skimr')

library(knitr)
```

### 0.0.0.0.1 Cargar librerías

# 1 Descripción del dataset

## 1.1 Objetivos y descripción del dataset original

El presente proyecto tiene como objetivo final el estudio comparativo de los coches de los principales fabricantes disponibles en el mercado de segunda mano de Reino Unido en el mes de julio del año 2020 que ayude en la toma de decisiones tanto del comprador como del vendedor.

Para poder llevar a cabo dicho objetivo se procede a la integración, limpieza, validación y análisis de un conjunto de datasets de coches usados en Reino Unido creados en Julio 2020 por el usuario ‘Aditya’ (<https://www.kaggle.com/adityadesai13>) a través de web scraping de portales de compraventa británicos. El objetivo inicial del usuario era la creación de un modelo de regresión lineal de coches usados para hacer predicciones sobre la variable target “price”, interpretándose como un análisis del precio de mercado. Citando al usuario: *“I collected the data to make a tool to predict how much my friend should sell his old car for compared to other stuff on the market, and then just extended the data set. Then made a more general car value regression model.”*

El resultado del web scraping son 13 ficheros individuales tipo CSV, entre los que seleccionamos un total de 9 ficheros, identificados por el nombre del fabricante, con las características de distintos modelos tales como el año, tipo de combustible, tipo de motor, kilometraje, precio actual, etc... Dichos ficheros se encuentran en el siguiente enlace: <https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes>. El resto de ficheros (“cclass.csv”, “focus.csv”, “unclean cclass.csv” y “unclean focus.csv”) no se tienen en cuenta para el presente análisis.

Para facilitar el estudio procedemos a la integración de los nueve ficheros de interés en un único fichero tipo csv al que llamaremos aprox100KUsedCars.csv. En dicho archivo incluiremos los campos de cada fichero más el campo “manufacturer” con el nombre del fabricante que extraeremos del nombre de cada fichero individual.

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

Para la integración de los ficheros hemos utilizado el script Ruby (ruby integration.rb) localizado en la carpeta “integration” en el enlace GitHub cuya ejecución crea en nuestra raíz del proyecto el fichero aprox100KUsedCars.csv. De este modo nos encontramos con un dataset con un total de 99187 filas y 10 columnas que representan 99187 ofertas en portales de compraventa de coches usados en Reino Unido. En la siguiente tabla se muestra un ejemplo del tipo de datos.

```
# Carga de los datos
used_cars <- read.csv("aprox100KUsedCars.csv", stringsAsFactors = TRUE)

sample_n(used_cars, 10) %>% knitr::kable()
```

manufacturer	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
hyundi	I20	2017	9990	Manual	16012	Petrol	30	55.4	1.2
merc	C Class	2017	25232	Automatic	15104	Diesel	145	58.9	2.1
audi	A6	2016	16700	Manual	28952	Diesel	30	64.2	2.0
audi	A5	2019	26480	Automatic	10936	Diesel	150	48.7	2.0
skoda	Yeti	2015	10480	Manual	32800	Diesel	165	49.6	2.0
ford	Fiesta	2017	9240	Manual	27667	Petrol	145	64.2	1.1
vauxhall	Zafira	2016	11950	Automatic	14085	Petrol	200	40.9	1.4
bmw	2 Series	2020	34950	Semi-Auto	1107	Diesel	145	62.8	2.0
merc	GLC Class	2018	49891	Semi-Auto	11744	Petrol	145	27.4	4.0

manufacturer	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
audi	Q3	2013	12490	Semi-Auto	64000	Diesel	200	47.9	2.0

Tabla 1. Ejemplo del tipo de datos almacenada en el dataset.

## 2.1 Descripción de las variables

A continuación mostramos un resumen del tipo de variables presentes en el dataset que constituyen los datos de interés a analizar en el que podemos observar la presencia de dos tipos principales de variables: categóricas (factor) y numéricas (integer o numeric).

```
# Tabla resumen del tipo de variables que conforman el dataset
tb_var <- sapply(used_cars, class)
kable(data.frame(variables = names(tb_var), clase = as.vector(tb_var)))
```

variables	clase
manufacturer	factor
model	factor
year	integer
price	integer
transmission	factor
mileage	integer
fuelType	factor
tax	integer
mpg	numeric
engineSize	numeric

Tabla 2. Tipo de variables

### 2.1.1 Variables categóricas (factor)

- manufacturer: Fabricante del automóvil. Variable categórica nominal con 9 categorías (Niveles) diferentes.
- model: Modelo del automóvil. Variable categórica nominal con 195 categorías diferentes.
- transmission: Tipo de transmisión. Variable categórica nominal con 4 categorías: manual, automática, semiautomática y otras.
- fuelType: Tipo de combustible. Variable categórica nominal con 5 categorías: diesel, eléctrico, híbrido, gasolina y otros.

```
sk <- skim(used_cars)
sk %>% yank('factor') %>% select(c(skim_variable, n_unique)) %>% rename(Variable=skim_variable, Niveles=n_unique)
```

Variable	Niveles
manufacturer	9
model	195
transmission	4

Variable	Niveles
fuelType	5

Tabla 3. Resumen de las variables categóricas y sus posibles valores

### 2.1.2 Variables numéricas

- year: Año de matriculación del coche. Variable numérica discreta.
- price: Precio en Libras que se colocó en el portal de compraventa a fecha Julio 2020. Variable numérica continua.
- mileage: kilométraje. Millas que el coche ha recorrido desde su puesta en funcionamiento. (En España utilizamos Kilometraje, porque medimos esta distancia en kilómetros). Variable numérica continua.
- tax: Impuesto de circulación (en Libras). Dependiendo de los años del vehículo, emisión de gases (sobre todo) y otros factores este impuesto varía. Variable numérica continua
- mpg: Consumo de combustible del vehículo en millas por galón. Variable numérica continua
- engineSize: Tamaño del motor en litros. Variable numérica continua.

```
sk %>% yank('numeric') %>% select(c(skim_variable, mean, sd, p25, p50, p75)) %>% rename(Variable=skim_
```

Variable	Media	Desviación Típica	Q1	Q2/Mediana	Q3
year	2017.088	2.124	2016.0	2017.0	2019.0
price	16805.348	9866.773	9999.0	14495.0	20870.0
mileage	23058.914	21148.524	7425.0	17460.0	32339.0
tax	120.300	63.151	125.0	145.0	145.0
mpg	55.167	16.139	47.1	54.3	62.8
engineSize	1.663	0.558	1.2	1.6	2.0

Tabla 4. Análisis descriptivo de las variables numéricas

## 3 Limpieza de los datos

### 3.1 Análisis de valores nulos o vacíos

Se comprueba la no existencia de ceros o elementos vacíos en el dataset mediante la ejecución de la función `is_blank` creada para tal propósito.

```
# Se crea la función is_blank que devuelve la presencia (TRUE) o no (FALSE) de valores vacíos ("" o null)

is_blank <- function(x){
  return (any(x=="") || anyNA(x))
}

used_cars %>% summarise_all(.funs=c('is_blank'), )

##   manufacturer model  year price transmission mileage fuelType    tax    mpg
## 1          FALSE FALSE FALSE FALSE        FALSE FALSE    FALSE FALSE FALSE
##   engineSize
## 1      FALSE
```

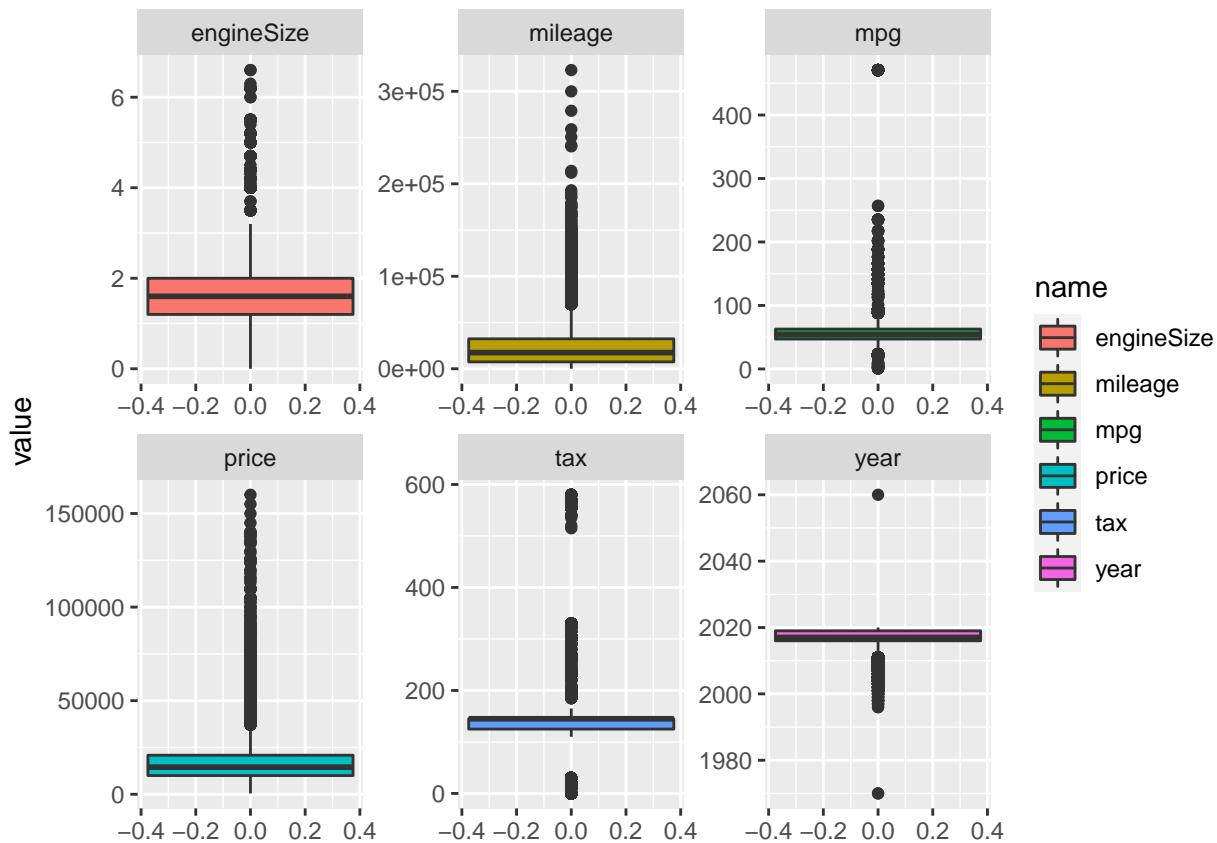
## 3.2 Análisis de valores atípicos

La representación gráfica en forma de diagrama de barras (boxplot) identifica la presencia de valores extremos superiores en todas las variables numéricas y sólo en algunas se observan también valores extremos inferiores.

```
# Boxplot para cada una de las variables numéricas (vars)
vars <- c("year", "price", "mileage", "tax", "mpg", "engineSize")

used_cars %>% select(vars) %>% pivot_longer(cols=vars, values_drop_na = TRUE) %>% ggplot(.) + facet_wrap(~name)
```

## Note: Using an external vector in selections is ambiguous.  
## i Use ‘all\_of(vars)’ instead of ‘vars’ to silence this message.  
## i See <<https://tidyselect.r-lib.org/reference/faq-external-vector.html>>.  
## This message is displayed once per session.



### 3.2.1 Identificación y tratamiento de outliers

Un análisis más profundo nos permite identificar aquellos valores extremos y valorar su tratamiento en función de la información almacenada en dicha variable.

Para ello creamos la función `get_outlier` que toma como referencia el método de la diferencia intercuartil (IQR) (ver <https://www.r-bloggers.com/2020/01/how-to-remove-outliers-in-r/>)

```
# Creación de la función get_outliers para la identificación de los valores extremos de cada una de las
get_outliers <- function(x){
  q1 = quantile(x, c(0.25))
  q3 = quantile(x, c(0.75))
  iqr = q3-q1
  result = sapply(x, function(y){
    if(y < q1 - 1.5*iqr){
      y
    }else if(y > q3+1.5*iqr){
      y
    }else{
      NA
    }
  })
  return(list(result, q1 - 1.5*iqr, q3+1.5*iqr))
}
```

La ejecución de la función `get_outliers` nos permite identificar los valores extremos para cada una de las variables de interés, cuyo resultado mostramos en forma de tabla junto con el rango inferior, rango superior y el número total de outliers identificados y tres ejemplos para cada una de las variables.

```
# Ejecución de la función get_outliers y su resultado en forma de tabla
final <- list()
outliers_analysis <- used_cars %>% select(vars) %>% sapply(., get_outliers)
for(i in vars){
  ej <- na.omit(outliers_analysis[,i][[1]])

  final <- cbind(final, c(paste(sample(ej, 3), sep=" ", collapse=" / "), outliers_analysis[,i][[2]], o
}
colnames(final) = vars
rownames(final) = c("Ejemplos", "Inferior", "Superior", "Número total")
data.frame(final) %>% knitr::kable()
```

	year	price	mileage	tax	mpg	engineSize
Ejemplos	2010 / 2007 / 2011	53490 / 47980 / 37507	84000 / 93000 / 95000	30 / 20 / 20	94.1 / 134.5 / 20.5	3.5 / 4.2 / 5.5
Inferior	2011.5	-6307.5	-29946	95	23.55	-
						2.22044604925031e-16
Superior	2023.5	37176.5	69710	175	86.35	3.2
Número total	1737	3669	3902	28815	939	650

A continuación detallamos el tratamiento de los valores extremos en función del tipo de variable así como las inconsistencias encontradas tras su análisis.

- Year: El año de matriculación no puede ser superior al año de la recogida de los datos (Julio 2020), por lo que eliminaremos del dataset aquellos coches con fecha de matriculación superior al 2020. Por otra parte, aunque el análisis muestra como valores extremos los coches matriculados antes del 2011, consideramos estos coches minoritarios pero válidos y los mantendremos en el dataset.

```
# Eliminamos los coches matriculados más tarde del 2020
original_used_cars <- used_cars
used_cars <- used_cars %>% filter(year <= 2020)
```

- Price: El análisis de valores extremos pone en evidencia una gran dispersión entre los distintos precios de los coches, con un rango inferior negativo. Dado que un valor negativo en el precio sería un claro error lo comprobamos y observamos que no existen. Sin embargo vemos que hay un porcentaje minoritario de coches, probablemente de alta gama, con un precio muy elevado sobre la media que en principio son reconocidos como valores extremos. Sin embargo, los consideramos válidos.

```
# Comprobamos que no hay coches con precio negativo
used_cars_neg_price <- used_cars %>% filter(price <= 0)
nrow(used_cars_neg_price)
```

```
## [1] 0
```

- Mileage: de igual manera no podemos aceptar kilometraje negativo, por lo que lo comprobamos y observamos que no existen. Al igual que en la variable price, hay una gran dispersión en esta variable. Dado que se trata de compra-venta de coches de segunda mano los consideramos válidos y no los eliminamos.

```
used_cars_neg_mil <- used_cars %>% filter(mileage <= 0)
nrow(used_cars_neg_mil)
```

```
## [1] 0
```

- Tax: El diagrama de barras muestra una clara estratificación en los impuestos en el que observamos tres tramos. Consideramos válidos los valores extremos mostrados en el análisis.
- Mpg: debemos tener en cuenta que nuestro dataset contiene tanto coches híbridos como eléctricos por lo que habrá que mantener los datos cuyo mpg sea bajo o 0. Se podría hacer una manipulación distinta de los outliers para los diferentes segmentos de fuelType pero para no aumentar complejidad la mantendremos tal como está.
- engineSize: de igual manera un motor eléctrico deberá tener volumen 0, otro sería considerado una inconsistencia. Comprobamos por tanto que aquellos coches con volumen 0 sean eléctricos en caso contrario eliminaremos estas filas.

```
# Comprobación del tipo de coche con tamaño de EngineSize =0
used_cars <- used_cars %>% filter(engineSize != 0.0 | (engineSize == 0.0 & fuelType == "Electric"))
```

Tras el análisis de outliers el conjunto de datos numérico que tenemos tiene la siguiente estadística descriptiva:

```
sk2 <- skim(used_cars)
sk2 %>% yank('numeric') %>% select(c(skim_variable, mean, sd, p25, p50, p75, p0, p100)) %>% rename(Vari...
```

Variable	Media	Desviación Típica	Q1	Q2/Mediana	Q3	Mínimo	Máximo
year	2017.087	2.114	2016.0	2017.0	2019.0	1970.0	2020.0
price	16805.023	9867.720	9999.0	14495.0	20873.0	450.0	159999.0

Variable	Media	Desviación Típica	Q1	Q2/Mediana	Q3	Mínimo	Máximo
mileage	23064.406	21156.809	7425.0	17456.0	32346.0	1.0	323000.0
tax	120.321	63.121	125.0	145.0	145.0	0.0	580.0
mpg	55.029	14.226	47.1	54.3	62.8	0.3	470.8
engineSize	1.668	0.552	1.2	1.6	2.0	0.0	6.6

## 4 Análisis de los datos.

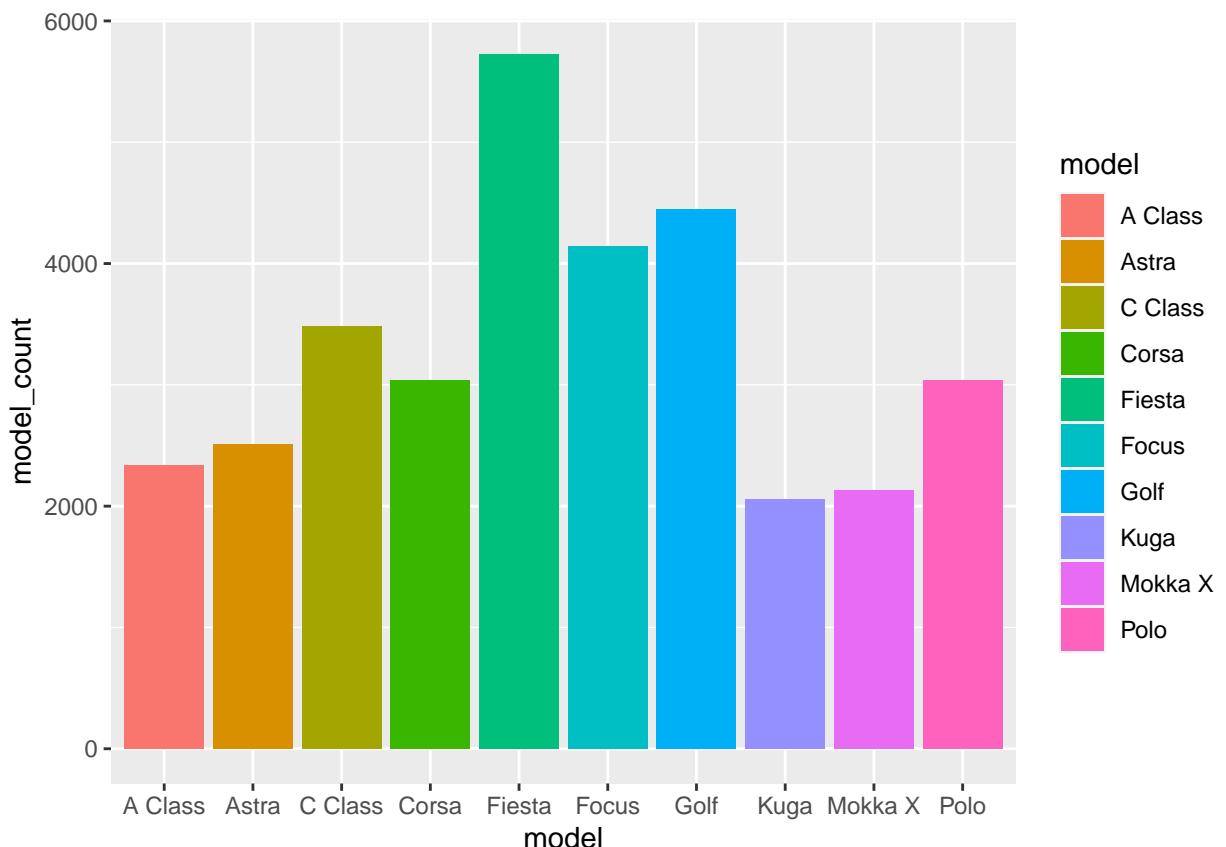
En esta sección vamos a detallar ejemplos a preguntas concretas que podemos resolver con nuestro dataset. Para ello realizamos una selección previa de los grupos en nuestro dataset.

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

- 1. Análisis para la toma de decisión de la compra de un coche con 5 años de antiguedad.

Creamos una función que nos devuelva el modelo más frecuente de coche por fabricante.

```
top_10_used_cars <- used_cars %>% filter(year >= 2015) %>% group_by(manufacturer, model) %>% summarize(model_count = n())
top_10_used_cars[,2:3] %>% ggplot(.) + geom_col(aes(model, model_count, fill=model))
```



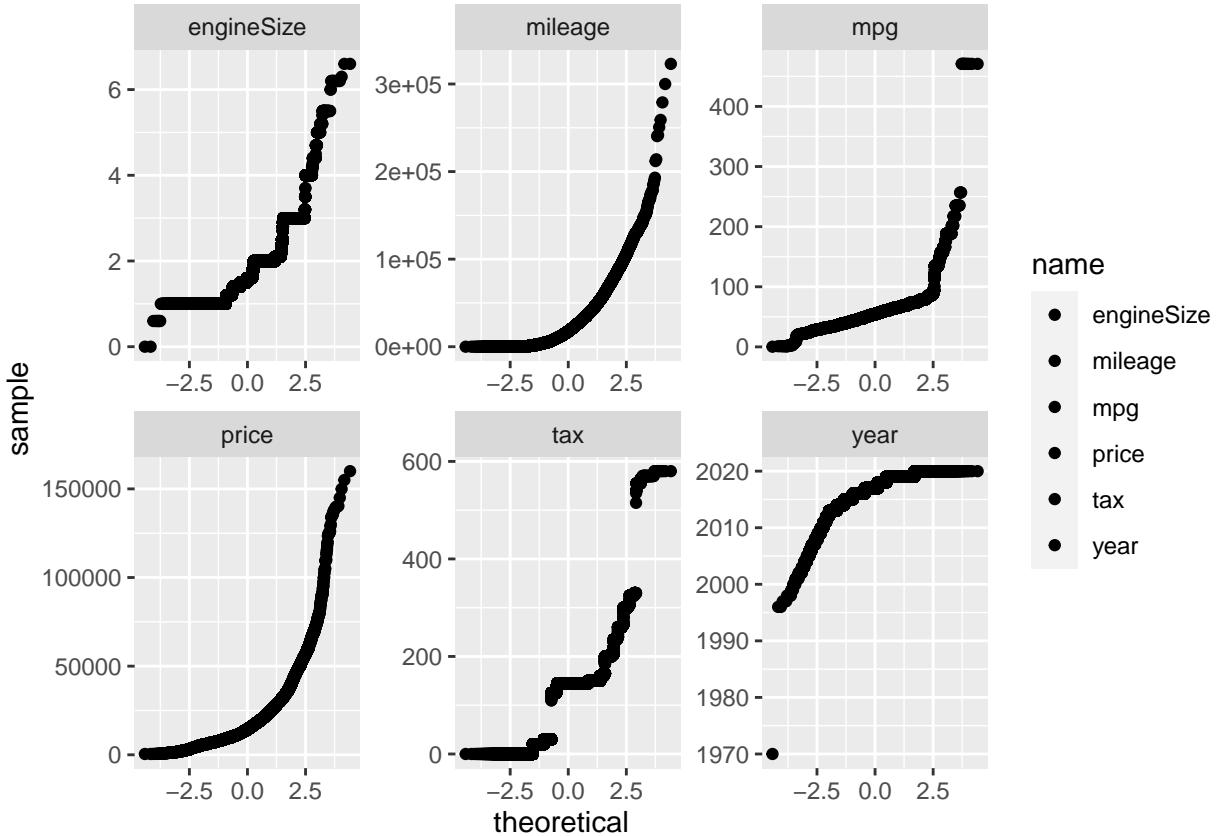
Una vez obtenida la información hacemos:

- varios contraste de hipótesis (ONE-WAY ANOVA) comparando: 1.1 precio medio de cada uno de los modelos en el año 2015 (5 años de antigüedad). 1.2 impuesto de matriculación medio 1.3 kilometraje medio

```
top_10_cars <- used_cars %>% filter((manufacturer %in% top_10_used_cars$manufacturer) & (model %in% top_10_models))
```

```
vars <- c("year", "price", "mileage", "tax", "mpg", "engineSize")
```

```
used_cars %>% select(vars) %>% pivot_longer(cols=vars, values_drop_na = TRUE) %>% ggplot(.) + facet_wrap(~name)
```



- Análisis de frecuencia de las características categóricas de cada uno de los modelos. 2.1 Tipo de transmisión 2.2 FuelType
- Seleccionamos tres de los modelos (los que más nos gusten) y estudiamos:

3.1 la evolución del precio del modelo en función del año de matriculación. De esta manera podemos obtener una estimación del precio al que podríamos vender el coche en los años posteriores a la compra. Regresión lineal.

3.2 el impacto del kilometraje en el precio del coche.

- En función de los resultados anteriores seleccionamos un modelo y comparamos: 4.1. El impacto del tipo de transmisión (manual vs. automática) en el precio, kilometraje e impuesto de matriculación.

## 5 Análisis. Caso práctico compra-venta

Somos una empresa que se dedica a hacer negocio comprando coches baratos, arreglándolos y volviéndolos a vender más caros. Queremos saber coches del conjunto de datos se venden al menos un 20% debajo de su precio de mercado (“chollos”) para poder comprarlos y revenderlos.

Para ello vamos a modelar el “mercado de compraventa” mediante un modelo de regresión lineal. Para ello utilizaremos la función “lm” la cual internamente transformará todas las clases de nuestras variables cualitativas en variables dicotómicas para poder obtener el valor estimado como la combinación lineal de todas las variables.

A continuación vemos los coeficientes:

```
used_cars[] <- lapply(used_cars, function(x) if(is.factor(x)) factor(x) else x)
lm_model <- lm(price~, used_cars)
lm_model$coefficients
```

```
##              (Intercept)      manufacturerbmw
##              -2.855802e+06   -4.624875e+03
##              manufacturerford   manufacturerhyundi
##              -5.333965e+03   -5.309136e+03
##              manufacturermerc   manufacturerskoda
##              1.308790e+03   -5.451869e+03
##              manufactorytoyota   manufacturervauxhall
##              -9.187311e+03   -6.794879e+03
##              manufacturervw     model180
##              -8.293321e+03   -2.187413e+03
##              model2 Series      model200
##              5.427032e+02   -5.096533e+03
##              model220          model3 Series
##              -5.054756e+03   2.156648e+03
##              model4 Series      model5 Series
##              2.215567e+03   4.154055e+03
##              model6 Series      model7 Series
##              2.450702e+03   1.089540e+04
##              model8 Series      modelA Class
##              3.060982e+04   -2.336430e+03
##              modelA1           modelA2
##              -3.723222e+03   1.216506e+04
##              modelA3           modelA4
##              -1.957558e+03   -1.687786e+03
##              modelA5           modelA6
##              -7.621704e+01   4.926291e+02
##              modelA7           modelA8
##              6.888025e+02   4.359135e+03
##              modelAccent       modelAdam
##              1.310217e+04   -2.508401e+03
##              modelAgila        modelAmarok
##              2.811577e+03   8.090919e+03
##              modelAmica        modelAmpera
##              6.138312e+03   1.560984e+04
##              modelAntara       modelArteon
##              1.432914e+03   7.719533e+03
##              modelAstra         modelAuris
```

```

##          -5.907927e+02      9.321676e+02
##          modelAvensis        modelAygo
##          3.769856e+03      8.435165e+02
##          modelB Class       modelB-MAX
##          -4.273815e+03     -2.104151e+03
##          modelBeetle        modelC Class
##          4.765610e+03      -1.407794e+03
##          modelC-HR          modelC-MAX
##          5.021464e+03      -8.110287e+02
##          modelCaddy          modelCaddy Life
##          5.024400e+03      2.434682e+03
##          modelCaddy Maxi    modelCaddy Maxi Life
##          6.117205e+03      2.306406e+03
##          modelCalifornia     modelCamry
##          3.966785e+04      8.754282e+02
##          modelCaravelle      modelCascada
##          2.370660e+04      8.993113e+02
##          modelCC              modelCitigo
##          2.918515e+03     -2.889752e+03
##          modelCL Class       modelCLA Class
##          -2.438452e+03     -1.836727e+03
##          modelCLC Class      modelCLK
##          -3.566322e+03     -5.116048e+02
##          modelCLS Class      modelCombo Life
##          -3.475859e+02      6.113969e+02
##          modelCorolla         modelCorsa
##          3.864385e+03      -3.342383e+03
##          modelCrossland X    modelE Class
##          -1.197898e+03      1.174535e+02
##          modelEcoSport        modelEdge
##          -1.187189e+02      5.538108e+03
##          modelEos              modelEscort
##          7.783121e+03      1.963050e+04
##          modelFabia            modelFiesta
##          -2.030361e+03     -6.560201e+02
##          modelFocus             modelFox
##          9.862316e+02      1.070872e+04
##          modelFusion            modelG Class
##          5.395058e+03      6.117242e+04
##          modelGalaxy           modelGetz
##          3.847060e+03      9.393738e+03
##          modelGL Class         modelGLA Class
##          -7.749881e+02     -3.608870e+03
##          modelGLB Class        modelGLC Class
##          7.233173e+03      5.055639e+03
##          modelGLE Class        modelGLS Class
##          9.449511e+03      1.389163e+04
##          modelGolf              modelGolf SV
##          4.440464e+03      2.688395e+03
##          modelGrand C-MAX     modelGrand Tourneo Connect
##          -3.257786e+02      2.380303e+03
##          modelGrandland X     modelGT86
##          3.004024e+03      4.736637e+03
##          modelGTC              modelHilux

```

##	-9.776344e+02	6.088339e+03
##	modelI10	modelI20
##	-3.854265e+03	-3.145850e+03
##	modeli3	modelI30
##	3.957444e+04	-9.227402e+02
##	modelI40	modeli8
##	-3.256362e+02	4.245655e+04
##	modelI800	modelInsignia
##	-4.905178e+03	2.905903e+02
##	modelIoniq	modelIQ
##	-2.246640e+03	7.091816e+03
##	modelIX20	modelIX35
##	-4.162977e+03	1.282404e+03
##	modelJetta	modelKA
##	1.640201e+03	-1.923813e+03
##	modelKa+	modelKadjar
##	-5.375647e+03	-3.888678e+02
##	modelKamiq	modelKaroq
##	1.398888e+03	3.499697e+03
##	modelKodiaq	modelKona
##	6.104293e+03	1.733894e+02
##	modelKuga	modelLand Cruiser
##	8.662033e+02	1.941492e+04
##	modelM Class	modelM2
##	-3.989109e+02	1.120039e+04
##	modelM3	modelM4
##	1.058093e+04	1.352692e+04
##	modelM5	modelM6
##	2.012980e+04	4.017600e+03
##	modelMeriva	modelMokka
##	-1.472467e+03	1.574487e+02
##	modelMokka X	modelMondeo
##	-1.785662e+03	1.836390e+02
##	modelMustang	modelOctavia
##	-3.493391e+03	-7.105724e+02
##	modelPassat	modelPolo
##	4.886297e+03	2.231534e+03
##	modelPrius	modelPROACE VERSO
##	6.365848e+03	1.073972e+04
##	modelPuma	modelQ2
##	5.940509e+03	-4.667672e+02
##	modelQ3	modelQ5
##	6.029353e+02	5.258900e+03
##	modelQ7	modelQ8
##	1.280519e+04	2.487790e+04
##	modelR Class	modelR8
##	-2.855628e+03	5.294189e+04
##	modelRanger	modelRapid
##	1.972737e+03	-1.532494e+03
##	modelRAV4	modelRoomster
##	3.590174e+03	1.345387e+03
##	modelRS3	modelRS4
##	6.479719e+03	1.697953e+04
##	modelRS5	modelRS6

```

##          1.670469e+04          2.011961e+04
##          modelRS7          models Class
##          7.073882e+03          1.285551e+04
##          models-MAX          models3
##          3.389038e+03          9.863654e+02
##          models4          models5
##          -3.983319e+01          -5.099362e+03
##          models8          modelSanta Fe
##          -3.926587e+02          5.481470e+03
##          modelScala          modelScirocco
##          -5.227877e+02          3.151741e+03
##          modelSharan          modelShuttle
##          6.368578e+03          7.990656e+03
##          modelSL CLASS          modelSLK
##          1.334623e+03          -2.817712e+03
##          modelsQ5          modelsQ7
##          2.982305e+03          1.441970e+04
##          modelStreetka          modelSuperb
##          8.265812e+03          1.807928e+03
##          modelSupra          modelT-Cross
##          2.309555e+04          6.139556e+03
##          modelT-Roc          modelTerracan
##          6.910312e+03          8.120669e+03
##          modelTigra          modelTiguan
##          6.720798e+03          7.361910e+03
##          modelTiguan Allspace          modelTouareg
##          1.005867e+04          1.087880e+04
##          modelTouran          modelTourneo Connect
##          6.852343e+03          1.590197e+03
##          modelTourneo Custom          modelTransit Tourneo
##          3.684761e+03          NA
##          modelTT          modelTucson
##          NA          -8.167014e+01
##          modelUp          modelUrban Cruiser
##          NA          6.355830e+03
##          modelV Class          modelVectra
##          6.141778e+03          7.815243e+03
##          modelVeloster          modelVerso
##          NA          2.530089e+03
##          modelVerso-S          modelViva
##          6.238150e+03          -3.558480e+03
##          modelVivaro          modelX-CLASS
##          5.432494e+03          NA
##          modelX1          modelX2
##          2.895605e+03          5.218400e+03
##          modelX3          modelX4
##          8.310073e+03          9.370188e+03
##          modelX5          modelX6
##          1.476602e+04          1.703289e+04
##          modelX7          modelYaris
##          3.899810e+04          NA
##          modelYeti          modelYeti Outdoor
##          5.504001e+02          NA
##          modelZ3          modelZ4

```

```

##          1.602423e+04          4.888320e+03
##      modelZafira      modelZafira Tourer
##          3.257250e+02                  NA
##      year      transmissionManual
##          1.423958e+03          -1.097409e+03
## transmissionOther      transmissionSemi-Auto
##          -3.766877e+02          7.560395e+02
##      mileage      fuelTypeElectric
##          -8.910463e-02          2.268662e+03
## fuelTypeHybrid      fuelTypeOther
##          5.469959e+03          3.485747e+03
## fuelTypePetrol      tax
##          1.682551e+03          -1.189618e+01
##      mpg      engineSize
##          -7.537524e+01          6.612009e+03

```

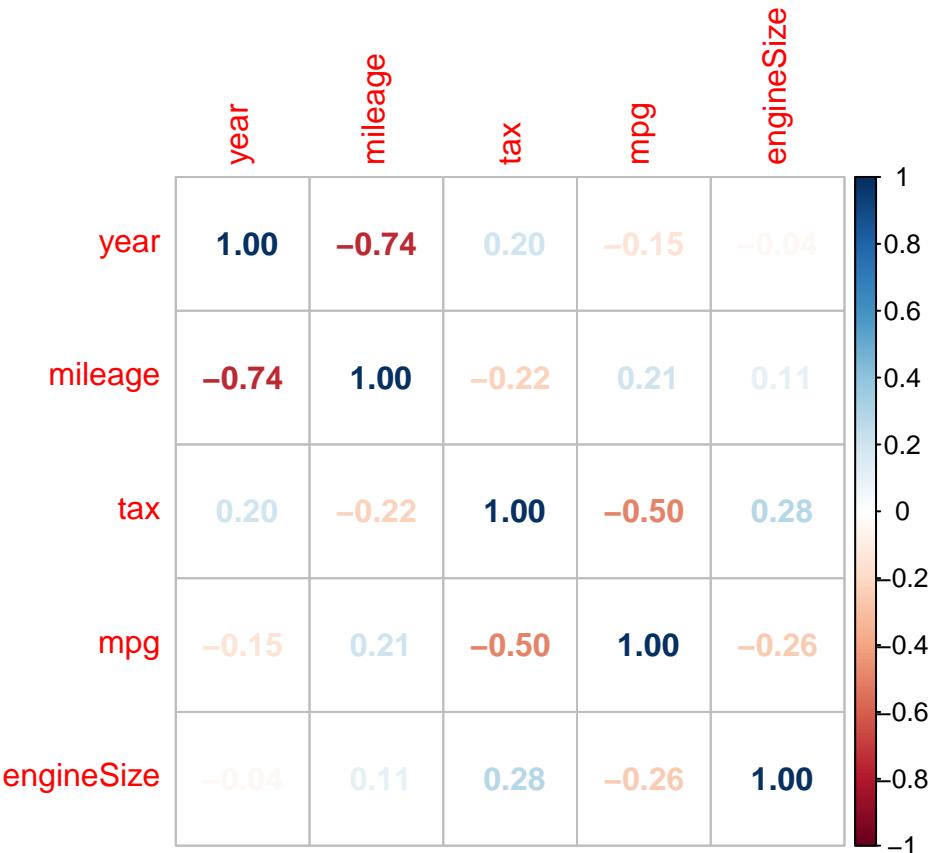
- Como podemos analizar por manufacturer los coches Mercedes “manufacturermerc” son más caros (“añaden” 1309€ a la estimación final) y los Toyota son más baratos (“restan” 9187€ a la estimación final).
- En cuanto a modelos por ejemplo Audi RS6 altera al alza el precio sumando 20120€ al precio.
- En cuanto a la transmision vemos que los Manuales son más baratos por lo general que los Semi-Auto.
- Y los coches híbridos son más caros que los de gasolina y diesel.
- Al ser más nuevo, el precio es mayor.
- Mientras más mayor es son las tasas el coche suele ser más barato (coches antiguos, contaminan más, mayor tasa)
- Al tener mayor “kilometraje” el precio es menor.
- El tamaño del motor determina positivamente el precio del coche.

Veamos también la correlación de las variables numéricas por si tuvieramos que eliminar alguna del modelo por ser redundante:

```

matriz_correl <- cor(used_cars %>% select(c("year", "mileage", "tax", "mpg", "engineSize")))
corrplot::corrplot(matriz_correl, method="number")

```



Vemos que “mileage” y “year” están correlacionadas inversamente, mientras más nuevo el coche, menor es el kilometraje. Igual sucede con el consumo y las tasas, mientras mayor distancia recorre con menos combustible, menor es la tasa (menor consumo, menor tasa). A pesar de todo no hablamos de correlaciones muy altas. Por lo que podemos aceptar el modelo.

Veamos la calidad de nuestro modelo mediante el RMSE (raíz de suma de mínimos cuadrados) para tenerlo el error en unidades de precio:

```
sqrt(mean(lm_model$residuals^2))
```

```
## [1] 3631.171
```

Vemos que no es un error despreciable y tendriamos que refinar un poco más nuestro modelo. Con el presente modelo haremos responderemos a la pregunta anterior ¿Cuáles son los chollos del conjunto, coches el 20% por debajo del precio de mercado?

```
mean((lm_model$fitted.values-(used_cars$price*1.2)) > 0)
```

```
## [1] 0.1402922
```

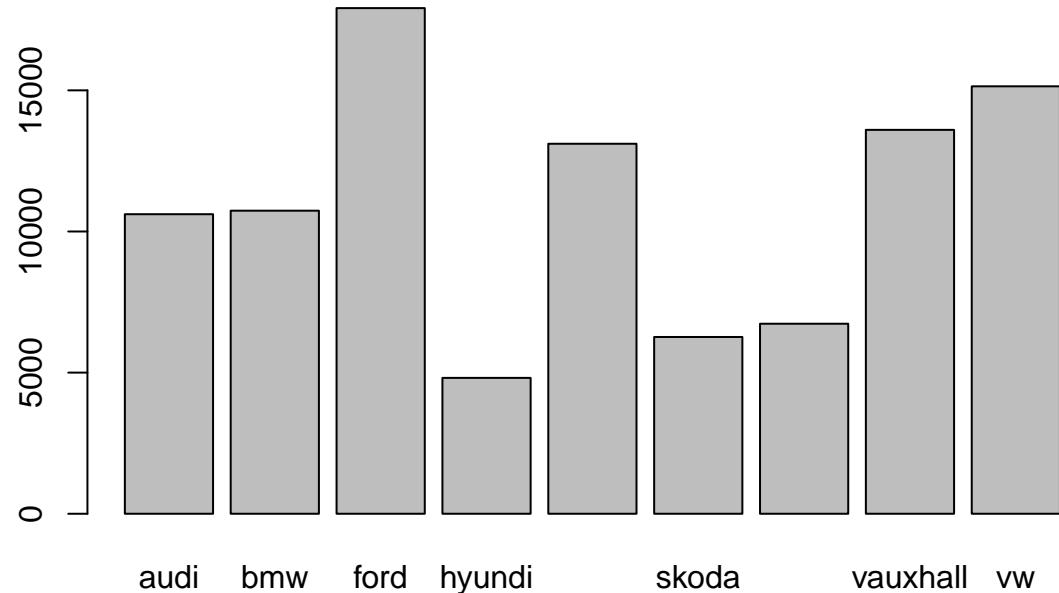
Tenemos que el ~14% está al menos un 20% por debajo del mercado. Por ejemplo (un sampleo de 10):

```
used_cars[(lm_model$fitted.values-(used_cars$price*1.2)) > 0,] %>% sample_n(., 10) %>% knitr::kable()
```

manufacturer	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
toyota	Prius	2019	22000	Automatic	4874	Hybrid	135	61.4	1.8
ford	Focus	2019	15970	Automatic	404	Petrol	145	47.9	1.5
merc	M Class	2012	14400	Automatic	61898	Diesel	300	39.2	3.0
bmw	3 Series	2016	15000	Automatic	53745	Diesel	145	54.3	3.0
merc	S Class	2016	24880	Automatic	75000	Hybrid	190	42.4	3.5
bmw	3 Series	2014	16882	Semi-Auto	41083	Diesel	125	57.6	3.0
vw	Touareg	2017	24998	Semi-Auto	31611	Diesel	235	42.8	3.0
vw	Golf	2017	14498	Semi-Auto	14662	Petrol	30	53.3	1.4
vauxhall	Mokka	2016	9863	Manual	4680	Petrol	200	40.9	1.6
vw	Passat	2017	14995	Semi-Auto	38418	Diesel	30	62.8	2.0

- 2. Modelo de regresión logistica para predecir el precio de un coche en función de sus características.

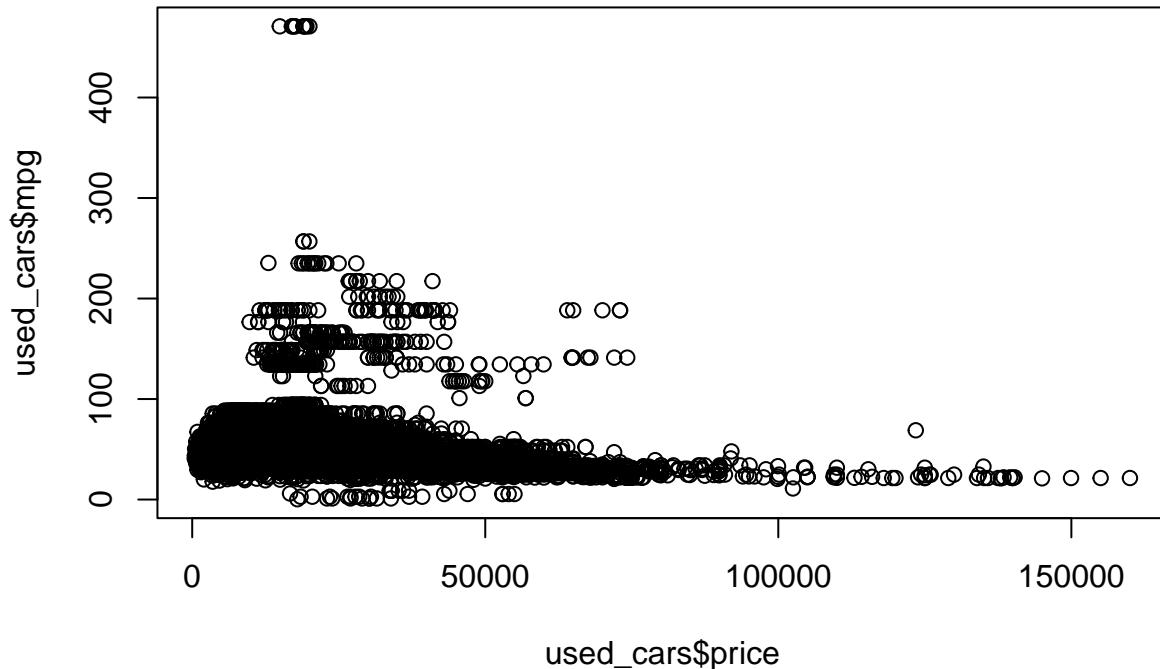
```
# Diagrama de barras con el número total de coches por fabricante
barplot(table(used_cars$manufacturer))
```



```
# Crear una función que nos devuelva el modelo más frecuente por fabricante de coche
```

- 2. Evolución del precio medio de los coches de segunda mano

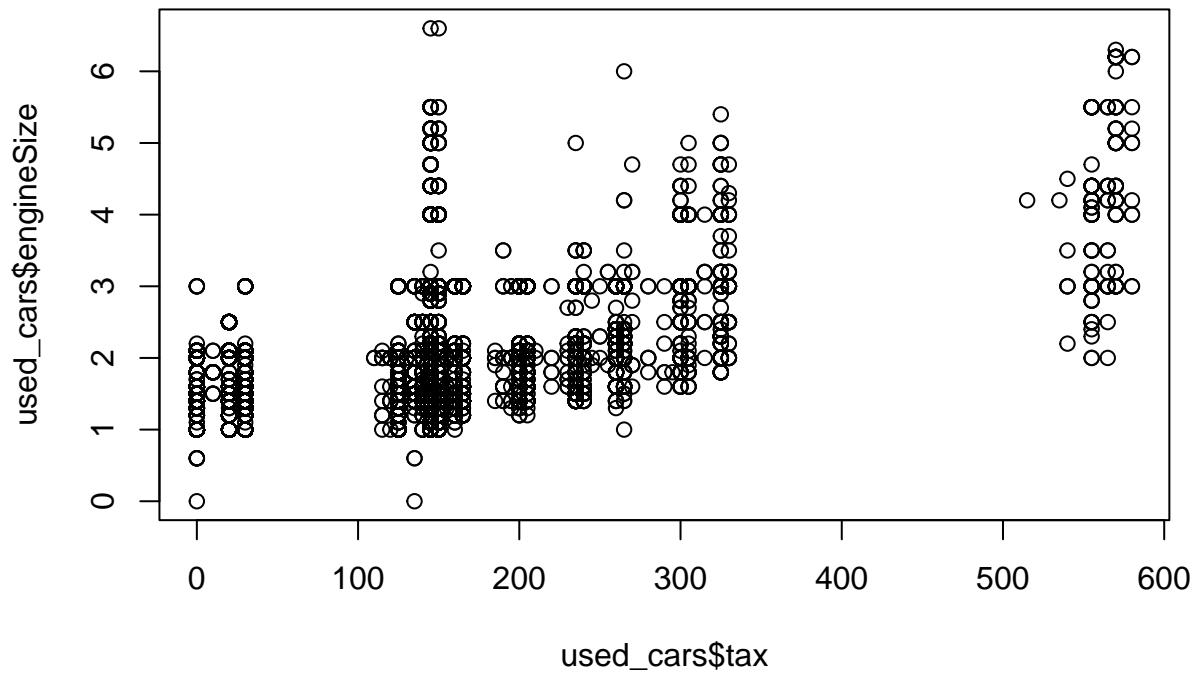
```
plot(used_cars$price, used_cars$mpg)
```



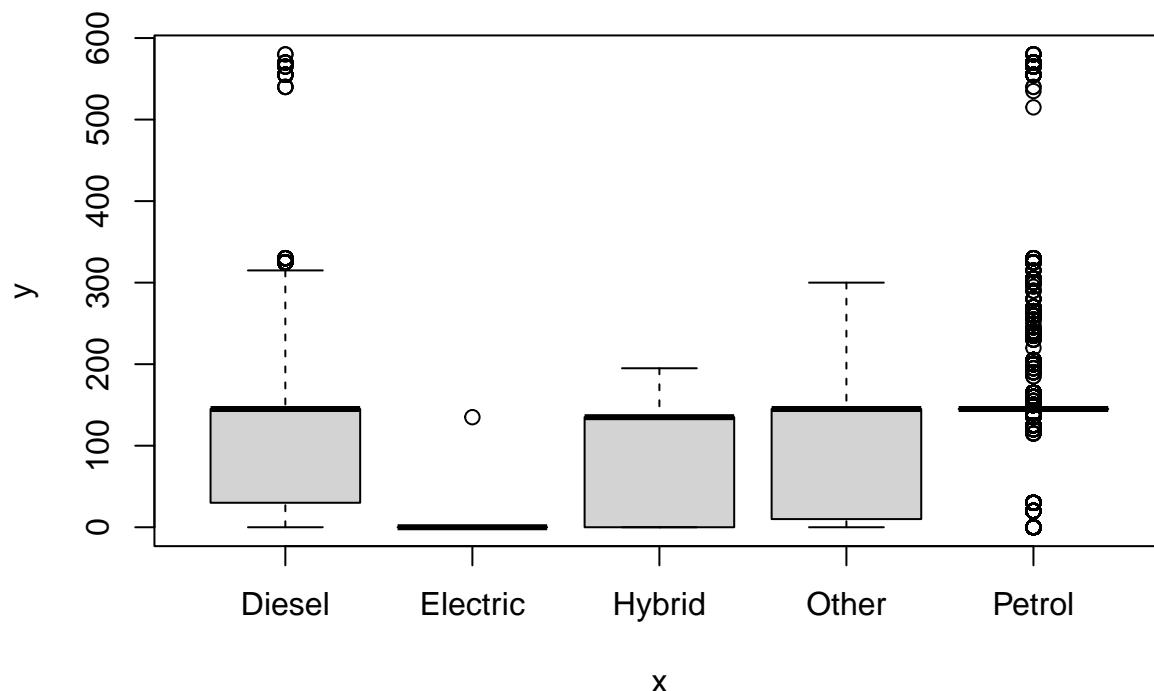
- 2. Clasificar los coches en función de su tamaño de motor
- 3. Predecir el precio de un coche en función de sus características
- 4. Comparar el precio medio de distintos modelos de coches con características similares de cuatro fabricantes distintos.
- 5. Comparar el precio de un mismo modelo matriculado con una diferencia de 5-10-15 años.
- 6. Estudiar el impacto del tamaño del motor en el impuesto de matriculación. No tenemos en cuenta los coches eléctricos (engineSize=0) y acotamos el análisis a una frangua de años de matriculación: 2015-2020.

```
# Subsetting
# Coches matriculados entre el 2015-2020
df <- used_cars
# No coches eléctricos
```

```
# Diagrama de puntos mostrando el impacto del tamaño del motor en el impuesto de matriculación
plot(used_cars$tax, used_cars$engineSize)
```



```
# Impuesto de matriculación en del combustible  
plot(used_cars$fuelType, used_cars$tax)
```



-### Comprobación de la normalidad y homogeneidad de la varianza.

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