Tipología y ciclo de vida de los datos

Práctica 2: Limpieza y análisis de datos

Sergio Moya Copa

2022/2023 - Semestre 1

Contents

1. Presentación del dataset **3**

1.1. Contexto . . . 3

1.2. Descripción . . . 3

1.3. Licencia . . . 3

1.4. Atributos . . . 3

1.5. Potencial del dataset . . . 4

2. Integración y selección **5**

2.1. Subselección de los datos originales . . . 5

2.2. Creación de nuevos atributos . . . 6

3. Limpieza de los datos **6**

3.1. Valores nulos . . . 6

3.2. Valores extremos . . . 9

3.3. Exportación de dataset preprocesado . . . 10

4. Análisis de los datos **10**

4.1. Selección de grupos de datos . . . 10

4.2. Comprobación de normalidad e homocedasticidad . . . 11

4.3. Aplicación de pruebas estadísticas . . . 12

4.3.1. ¿Qué variables influyen más en los consumos del vehículo? . . . 12

4.3.2. ¿Qué vehículos són más eficientes? . . . 13

4.3.4. Modelo de regresión lineal . . . 16

Colinealidad . . . 17

Diagnosis del modelo generado . . . 18

Predicción . . . 19

5. Conclusiones **20**

1. Presentación del dataset

1.1. Contexto

En el contexto actual de alarmante situación ecológica, las emisiones de gases por parte de los vehículos son un problema del que la población es cada vez más consciente, al ser uno de los principales contribuidores a la contaminación y al calentamiento global. Los vehículos emiten gases de efecto invernadero como el dióxido de carbono y el metano, así como otros contaminantes como el monóxido de carbono y los óxidos de nitrógeno, que pueden ser dañinos para la salud humana y el medio ambiente. Además, con el aumento del tráfico y la congestión de las ciudades, es cada vez un problema más grave la mala calidad del aire en las áreas urbanas.

Para abordar este problema es importante desarrollar vehículos menos contaminantes y más eficientes en términos de energía (así como promover modos de transporte más sostenibles como el transporte público y la bicicleta).

La sostenibilidad de los vehículos es un factor cada vez a tener más en cuenta por los compradores de nuevos vehículos. Para facilitar el proceso de selección, y siguiendo normativas europeas, el Gobierno de España a través del Instituto para la Diversificación y Ahorro de la Energía (IDAE) publica cada año un catálogo de todos los vehículos disponibles en el mercado (excepto algunos fabricantes de alta gama como Ferrari, Bentley o Aston Martin). En este catálogo se especifica para cada uno de los modelos todas sus características básicas (peso, medidas, capacidad) así como las características relevantes para su sostenibilidad (emisiones, consumo).

1.2. Descripción

A pesar de que los datos se pueden encontrar de forma abierta en la página web del IDEA [1], no está habilitada una opción de descargar todo el conjunto de datos al completo para su explotación. Por ese motivo se ha utilizado el dataset *car-emissions-spain-2022* [2] publicado en Kaggle por el usuario Maurici. Para obtener este dataset se ha hecho web scraping de la totalidad del catálogo, obteniendo una versión actualizada a fecha de Julio de 2022. Los datos han sido previamente limpiados y preprocesados: corrección de tipos de variables y columnas intercambiadas en algunos de los casos. Otros procesos de preparación aplicados adicionalmente serán especificados en el punto 2.

Referencias

[1] https://coches.idae.es/

[2] https://www.kaggle.com/datasets/mauriciy/car-emissions-spain-2022?resource=download

1.3. Licencia

El dataset se encuentra bajo una licencia *Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)*, la cual permite copiar y redistribuir el material en cualquier formato, así como modificarlo, transformarlo y añadir contenido para cualquier finalidad.

Las términos bajo los que se goza de tales libertades son: obligatoriedad de acreditar el origen de los datos, informar debidamente de cualquier modificación y distribución del resultado bajo la misma licencia.

1.4. Atributos

- **id** (Número entero): Identificador único para cada modelo.

-**make** (Categoría): Fabricante.

-**model** (Texto): Modelo.

-**market_segment** (Categoría): Tipo de vehículo.

-**engine_type** (Categoría): Tipo de motor.

-**consumption_min_l_100km** (Número): Consumo mínimo de carburante a los 100km (L).

-**consumption_max_l_100km** (Número): Consumo máximo de carburante a los 100km (L).

-**emissions_min_gCO2_km** (Número): Emisiones mínimas de CO2 por kilómetro (g).

-**emissions_min_gCO2_km** (Número): Emisiones máximas de CO2 por kilómetro (g).

-**transmission** (Categoría): Tipo de transmisión.

-**engine_displacement_cm3** (Número entero): Cilindrada (cm3).

-**power_cv** (Número): Potencia (caballos).

-**power_ice_kw** (Número): Potencia de refrigeración (Kw).

-**power_electric_kw** (Número): Potencia eléctrica (Kw).

battery_range_kw (Número): Rango de batería (Kw).

-**{avg_wltp_consumption_l_100km** (Número): Promedio de consumo WLTP [3] a los 100 km (L).

-**avg_wltp_emission_gCO2_100km** (Número): Promedio de emisión de CO2 WLTP [3] a los 100 km (g).

-**length_mm** (Número entero): Longitud (mm).

-**width_mm** (Número entero): Ancho (mm).

-**height_mm** (Número entero): Altura (mm).

-**gross_vehicle_weight_rating_kg** (Número entero): Peso (Kg).

-**total_seating** (Número entero): Ocupantes.

-**fuel_economy_index** (Categórico) :Índice de economía de combustible (IDAE).

-**type_hybrid** (Categórico): Indica si el vehículo es híbrido.

-**electric_consumption_kwh_100km** (Numérico): Consumo de vehículos eléctricos a los 100 km.

-**{battery_capacity_kwh**(Numérico): Capacidad de la batería en vehículos eléctricos.

Referencias

[3] https://en.wikipedia.org/wiki/Worldwide_Harmonised_Light_Vehicles_Test_Procedure

1.5. Potencial del dataset

Un dataset de estas características permitirá analizar una gran variedad de variables de los vehículos (fabricante, tamaño, tipo, cilindrada, peso, ocupantes) en base a su rendimiento según parámetros de sostenibilidad y ecologismo (consumo de combustible y emisiones de CO2).

En general, podemos considerar que nos encontramos ante un dataset adecuado para la visualización de datos ya que:

- La fuente es una organización fiable (institución gubernamental).
- Alta densidad de datos: más de 15000 registros, más de 25 atributos sin apenas valores perdidos.

- Permite realizar diferentes preguntas relacionadas con la sostenibilidad de los vehículos.
- Se podría combinar con otras fuentes. Por ejemplo, si incluimos los precios podemos plantear un análisis de la sostenibilidad relacionada al coste.

2. Integración y selección

```
# Carga del dataset e inspección de los tipos de las variables
data_raw = read.csv("idae_emissions.csv", header=TRUE, sep=",")

str(data_raw)
```

```
## 'data.frame':    15753 obs. of  26 variables:
##  $ id                             : int  551266 464453 464454 464455 464456 464457 464458 464460 4644
##  $ make                           : chr  "AIWAYS" "ALKE" "ALKE" "ALKE" ...
##  $ model                          : chr  "AIWAYS U5 MAS861-WVTA/2WB/FL4" "ALKE ATX 310 E plomo acido"
##  $ market_segment                 : chr  "Berlinas-Familiares Medios" "Chasis-Cabina Pequeño" "Chasis
##  $ engine_type                    : chr  "Eléctricos puros" "Eléctricos puros" "Eléctricos puros" "El
##  $ consumption_min_l_100km        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ consumption_max_l_100km        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ emissions_min_gCO2_km          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ emissions_max_gCO2_km          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ transmission                   : chr  "A" "SC" "SC" "SC" ...
##  $ engine_displacement_cm3        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ power_cv                       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ power_ice_kw                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ power_electric_kw              : num  55 14 14 14 14 14 14 14 14 14 ...
##  $ battery_range_km               : num  400 75 75 75 75 75 75 80 150 90 ...
##  $ avg_wltp_consumption_l_100km   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_wltp_emissions_gCO2_km     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ length_mm                      : int  4680 3030 3850 3850 4610 3850 4610 3850 3850 3850 ...
##  $ width_mm                       : int  1865 1500 1500 1500 1500 1500 1500 1500 1500 1500 ...
##  $ height_mm                      : int  1700 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
##  $ gross_vehicle_weight_rating_kg: int  2135 1510 1510 1510 2150 2510 2510 2150 2150 2150 ...
##  $ total_seating                  : int  5 2 2 2 4 2 4 2 2 2 ...
##  $ fuel_economy_index             : chr  "Sin clasificación" "Sin clasificación" "Sin clasificación"
##  $ type_hybrid                    : chr  "" "" "" "" ...
##  $ electric_consumption_kwh_100km: num  15.8 11 11 11 11 ...
##  $ battery_capacity_kwh           : num  63 10 10 10 10 10 10 10 20 14.4 ...
```

```
# El tipo de variable 'factor' facilitará trabajar con las variables categóricas
char_cols = sapply(data_raw, is.character)
data_raw[,char_cols] = lapply(data_raw[,char_cols], as.factor)
```

2.1. Subselección de los datos originales

Vamos a prescindir de aquellos atributos que nos resulten redundantes o con poca utilidad para nuestro posterior análisis.

· **id** y **model**, ya que no estamos interesados en los modelos concretos, si no en sus características como vehículo.

· Los cuatro atributos de emisiones y consumo máximos y mínimos, ya que por simplicidad trabajaremos con los valores promedio (WLTP).

· El atributo **type_hybrid**, ya que resulta redundante: la información de si un coche es híbrido ya se encuentra dentro del atributo **engine_type**.

2.2. Creación de nuevos atributos

Ya que una parte importante del análisis va a consistir en analizar las diferencias entre los diferentes tipos de vehículos y motores, vamos a crear unas versiones simplificadas que reduzcan las posibilidades a:

- Tipos de coche: ligeros (turismos, furgonetas, motocicletas) y pesados (camiones, autobuses).
- Tipos de motor: eléctricos, híbridos y combustibles (gasolina, gasóleo y derivados).

```
vehicle_type <- as.factor(ifelse(data$market_segment %in% c("Autobús/Autocar Rígido", "Camión","Autobús/

engine = as.factor(ifelse(data$engine_type %in% c("Gasolina", "Gasóleo","Gas natural", "Gases licuados d
                ifelse(data$engine_type %in% c("Híbridos enchufables", "Híbridos de gasóleo","Híbridos d

data$vehicle_type = vehicle_type
data$engine = engine

data = data[, !(names(data) %in% c("market_segment", "engine_type"))]
```

3. Limpieza de los datos

3.1. Valores nulos

Se lleva a cabo una inspección de valores nulos para eliminar aquellos registros o atributos con exceso de datos perdidos.

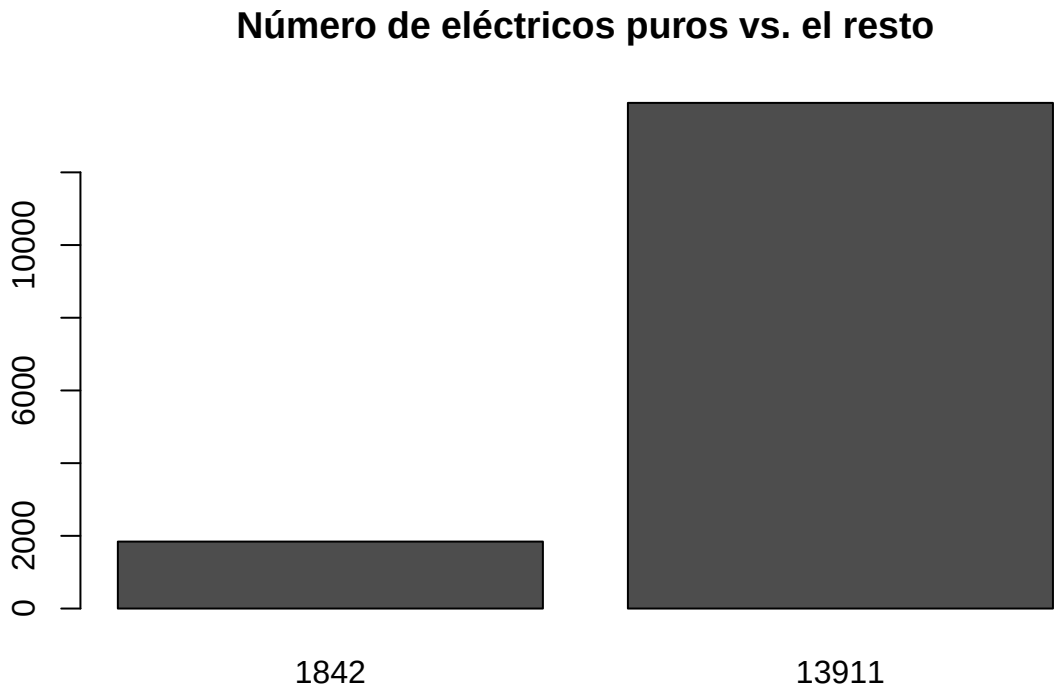
```
# Suma de valores nulos en cada atributo
colSums(is.na(data))
```

```
##                            make                    transmission
##                               0                               0
##         engine_displacement_cm3                        power_cv
##                               0                               0
##                    power_ice_kw              power_electric_kw
##                               0                               0
##                battery_range_km    avg_wltp_consumption_l_100km
##                               0                            1842
##      avg_wltp_emissions_gCO2_km                       length_mm
##                              42                               0
##                        width_mm                       height_mm
##                               0                               0
## gross_vehicle_weight_rating_kg                   total_seating
##                               0                               0
##              fuel_economy_index electric_consumption_kwh_100km
##                               0                           13911
```

```
##           battery_capacity_kwh                   vehicle_type
##                          13911                              0
##                         engine
##                              0
```

Ya que muchos de estos valores pueden deberse al carácter eléctrico puro de los vehículos, comprobamos la distribución.

```
num_electr = sum(data$engine == "Eléctricos")
num_no_elec = nrow(data) - num_electr

barplot(cbind(num_electr, num_no_elec), names.arg = c(num_electr, num_no_elec), main = "Número de eléct
```



Los datos perdidos de *avg_wltp_consumption_l_100km*, *electric_consumption_kwh_100km* y *battery_capacity_kwh* son explicables por las características de los vehículos eléctricos puros (~11.7%), por lo que se mantienen sin cambios. Los datos se separarán ambos grupos para evitar que los valores nulos afecten al análisis.

El número de valores perdidos de *avg_wltp_emissions_gCO2_km* es tan pequeño (<0.5%) como para que podamos permitirnos eliminar los registros sin resentir la calidad del dataset.

```
data = data[!is.na(data$avg_wltp_emissions_gCO2_km),]
```

Adicionalmente, tras comprobar los distribución de las variables con histogramas, detectamos una presencia de valores 0 muy mayoritaria en varios de los atributos restantes.

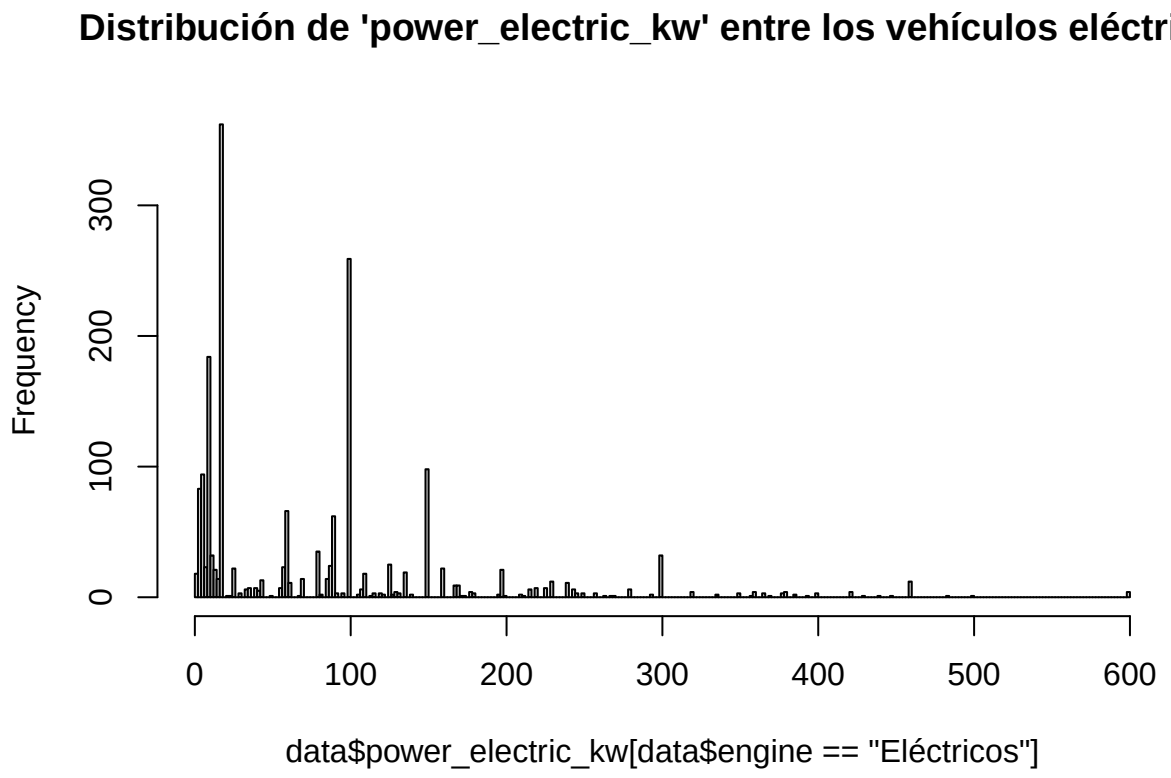
```r
numeric_data = data %>% select_if(is.numeric)

# Número de ceros que contiene cada una de las variables numéricas
colSums(numeric_data==0)
```

```
##        engine_displacement_cm3                       power_cv
##                           1818                           1812
##                  power_ice_kw             power_electric_kw
##                           1812                          12601
##               battery_range_km   avg_wltp_consumption_l_100km
##                          13384                             NA
##     avg_wltp_emissions_gCO2_km                      length_mm
##                           3428                              2
##                       width_mm                      height_mm
##                              2                              2
## gross_vehicle_weight_rating_kg                 total_seating
##                              1                              1
## electric_consumption_kwh_100km          battery_capacity_kwh
##                             NA                             NA
```

El número de ceros en las variables *engine_displacement_cm3*, *power_cv* y *power_ice_kw* puede explicarse de nuevo por la presencia de vehículos eléctricos puros, ya que para estas variables parece que se han utilizado ceros en lugar de valores NA.

En el caso de *battery_range_km* se decide eliminar la variable debido a su redundancia con *battery_capacity_kwh*, mientras que *power_electric_kw* se elimina por su alta cantidad de ceros incluso entre los vehículos eléctricos.

```r
hist(data$power_electric_kw[data$engine=="Eléctricos"], breaks=400, main = "Distribución de 'power_elec
```




```
data = data[, !(names(data) %in% c("power_electric_kw", "battery_range_km"))]
```

3.2. Valores extremos

Los valores extremos o outliers son aquellos que, debido a su desviación del resto, parecen no ser congruentes por alguna razón. Podemos detectarlos gráficamente al representar los datos en *boxplots*, o directamente usando la función integrada *boxplot.stats*, que usaremos para nuestras variables numéricas.

```
numeric_data = numeric_data[, !(names(numeric_data) %in% c("power_electric_kw", "battery_range_km"))]

# Aplicamos boxplot.stats a cada una de las variables numéricas
outliers <- lapply(numeric_data, function(x) boxplot.stats(x)$out)

# Contamos el número de outliers de cada variable
sapply(outliers, function(x) length(x))
```

```
##        engine_displacement_cm3                      power_cv
##                           3143                          3653
##                   power_ice_kw  avg_wltp_consumption_l_100km
##                           3653                          2404
##    avg_wltp_emissions_gCO2_km                     length_mm
##                             28                          2150
##                       width_mm                     height_mm
##                           2466                          1612
```

```
## gross_vehicle_weight_rating_kg                  total_seating
##                            1537                           3629
## electric_consumption_kwh_100km           battery_capacity_kwh
##                            284                             63
```

A pesar de que debemos descontar el número de ceros en las tres primeras variables (al restar los aproximadamente 1800 ouliers quedarían alrededor de 1300 outliers), siguen siendo números bastante altos en la mayoría de variables, por lo que vamos a estudiar su rango para entender su origen.

```
# Obtenemos el rango de los outliers de cada variable
sapply(outliers, function(x) range(x))
```

```
##      engine_displacement_cm3 power_cv power_ice_kw avg_wltp_consumption_l_100km
## [1,]                       0     0.00            0                         0.00
## [2,]                   19894   729.62          537                        33.45
##      avg_wltp_emissions_gCO2_km length_mm width_mm height_mm
## [1,]                        323         0        0         0
## [2,]                        790     24500     2600      4000
##      gross_vehicle_weight_rating_kg total_seating
## [1,]                              0             0
## [2,]                          36000           180
##      electric_consumption_kwh_100km battery_capacity_kwh
## [1,]                             48                 140
## [2,]                           2700                4800
```

Podemos observar que los únicos valores inverosímiles son los ceros en los atributos físicos de los vehículos (dimensiones y peso).

```
data = subset(data,length_mm !=0)
data = subset(data,width_mm !=0)
data = subset(data,height_mm !=0)
data = subset(data,gross_vehicle_weight_rating_kg  !=0)
```

Por tanto concluímos que la alta presencia de valores extremos viene dada por el hecho de combinar en un mismo dataset vehículos de características tan diferentes. Por ejemplo, al incluír el peso de un camión en un dataset en el que hace media con una mayoría de coches y motocicletas, se interpreta como una desviación atípica a pesar de ser un dato real. Se decide no eliminar ningún outlier.

3.3. Exportación de dataset preprocesado

```
write.csv(data, "idae_emissions_clean.csv")
```

4. Análisis de los datos

4.1. Selección de grupos de datos

```r
# 1) Separamos en eléctricos puros y vehículos que utilizan combustible
electricos = data[data$engine == "Eléctricos",]
fuel = data[data$engine != "Eléctricos",]

# Eliminamos los datos que no corresponden a cada grupo
electricos = electricos[, !(names(electricos) %in% c("avg_wltp_consumption_l_100km", "avg_wltp_emissions
fuel = fuel[, !(names(fuel) %in% c("electric_consumption_kwh_100km", "battery_capacity_kwh"))]
```

Comprobamos que, como anticipábamos, al realizar la separación entre tipos de motores, eliminamos los valores perdidos y reducimos los ceros a valores razonables.

```r
sum(is.na(fuel))
```

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

```r
sum(fuel==0)
```

```
## [1] 3223
```

4.2. Comprobación de normalidad e homocedasticidad

Comprobaremos la normalidad y homocedasticidad de las variables numéricas de nuestros grupos de coches eléctricos y no-eléctricos. La utilización de otros subconjuntos requerirá nuevas comprobaciones, ya que la selección de un subconjunto, cuando no es realizada aleatoriamente, puede introducir sesgos.

Para comprobar la normalidad de los atributos aplicaremos la *prueba de Lilliefors* a cada una de las variables.

```r
fuel_num = fuel %>% select_if(is.numeric)
elec_num = electricos %>% select_if(is.numeric)
```

```r
# Fuel
for(i in names(fuel_num)) {
    result =lillie.test(fuel_num[,i])
    if (result$p.value > 0.05){
      print(paste0("No sigue una distribución normal: ",i))
    }
}

# Electrico
for(i in names(elec_num)) {
    result =lillie.test(elec_num[,i])
    if (result$p.value > 0.05){
      print(paste0("No sigue una distribución normal: ",i))
    }
}
```

Podemos observar que para ambos casos, todos los atributos son incapaces de rechazar la hipótesis nula y por lo tanto siguen una distribución normal.

A continuación, a modo de ejemplo del estudio de la homocedasticidad entre diferentes grupos, vamos a estudiar la homogeneidad de varianzas entre los consumos de los diferentes tipos de vehículos que forman nuestra cohorte de vehículos a combustible. Si es necesario se realizarán durante el análisis comprobaciones equivalentes para otros grupos y variables.

```r
leveneTest(avg_wltp_consumption_l_100km ~ vehicle_type, data = fuel)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  911.76 < 2.2e-16 ***
##       13866
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vemos que en este caso la varianza del consumo de combustible no es homogénea entre los 2 tipos de vehículos (ligeros/pesados).

4.3. Aplicación de pruebas estadísticas

4.3.1. ¿Qué variables influyen más en los consumos del vehículo?

Para comprobar qué variables son las que tienen mayor peso a la hora de determinar los consumos de combustible o electricidad en los vehículos, vamos a realizar un análisis de correlación entre todos los atributos y el atributo objetivo en cada ocasión. Ya que hemos comprobado que nos encontramos ante datos normalmente distribuídos, podremos calcular el coeficiente más común para este propósito: la correlación de **Pearson**.

```r
# Consumo de combustible
for(i in names(fuel_num)) {
    if (i != 'avg_wltp_consumption_l_100km'){
      corr = cor(fuel_num[, i], fuel_num$avg_wltp_consumption_l_100km, method = 'pearson')
      pval = cor.test(fuel_num[, i], fuel_num$avg_wltp_consumption_l_100km, method = 'pearson')
       print(paste(i, ":", corr, ", pval:", pval$p.value))

    }
}
```

```
## [1] "engine_displacement_cm3 : -0.519564434123786 , pval: 0"
## [1] "power_cv : -0.0809657882705965 , pval: 1.30257670358622e-21"
## [1] "power_ice_kw : -0.0809678687329397 , pval: 1.29948509434486e-21"
## [1] "avg_wltp_emissions_gCO2_km : 0.962500996178463 , pval: 0"
## [1] "length_mm : -0.56268984656224 , pval: 0"
## [1] "width_mm : -0.471721795608431 , pval: 0"
## [1] "height_mm : -0.530117898050641 , pval: 0"
## [1] "gross_vehicle_weight_rating_kg : -0.589390683282426 , pval: 0"
## [1] "total_seating : -0.0358259212807951 , pval: 2.44385376456571e-05"
```

Parece lógico que la mayor correlación se obtenga con las emisiones de CO2 (relación estrecha consumo-emisiones).

Mucho más sorprendente es la correlación negativa (aunque más moderada) con el resto de atributos, lo que parece llevar a conclusiones contraintuitivas como que una mayor cilindrada equivale a menor consumo.

```r
# Consumo eléctrico
for(i in names(elec_num)) {
    if (i != 'electric_consumption_kwh_100km'){
```

```
      corr = cor(elec_num[, i], elec_num$electric_consumption_kwh_100km, method = 'pearson')
       pval = cor.test(elec_num[, i], elec_num$electric_consumption_kwh_100km, method = 'pearson')
       print(paste(i, ":", corr, ", pval:", pval$p.value))

    }
}
```

```
## [1] "engine_displacement_cm3 : -0.0255561939049782 , pval: 0.273091608008274"
## [1] "power_cv : -0.0151197807695343 , pval: 0.516765717873188"
## [1] "power_ice_kw : -0.0151201579014164 , pval: 0.516755259577568"
## [1] "length_mm : 0.127940498481035 , pval: 3.62007252797992e-08"
## [1] "width_mm : 0.120149843717733 , pval: 2.33476776816432e-07"
## [1] "height_mm : 0.155775630222327 , pval: 1.81264378509695e-11"
## [1] "gross_vehicle_weight_rating_kg : 0.0718001430208853 , pval: 0.00205196284483321"
## [1] "total_seating : 0.0122101556091833 , pval: 0.600582299484915"
## [1] "battery_capacity_kwh : 0.00283997341454953 , pval: 0.903079692598189"
```

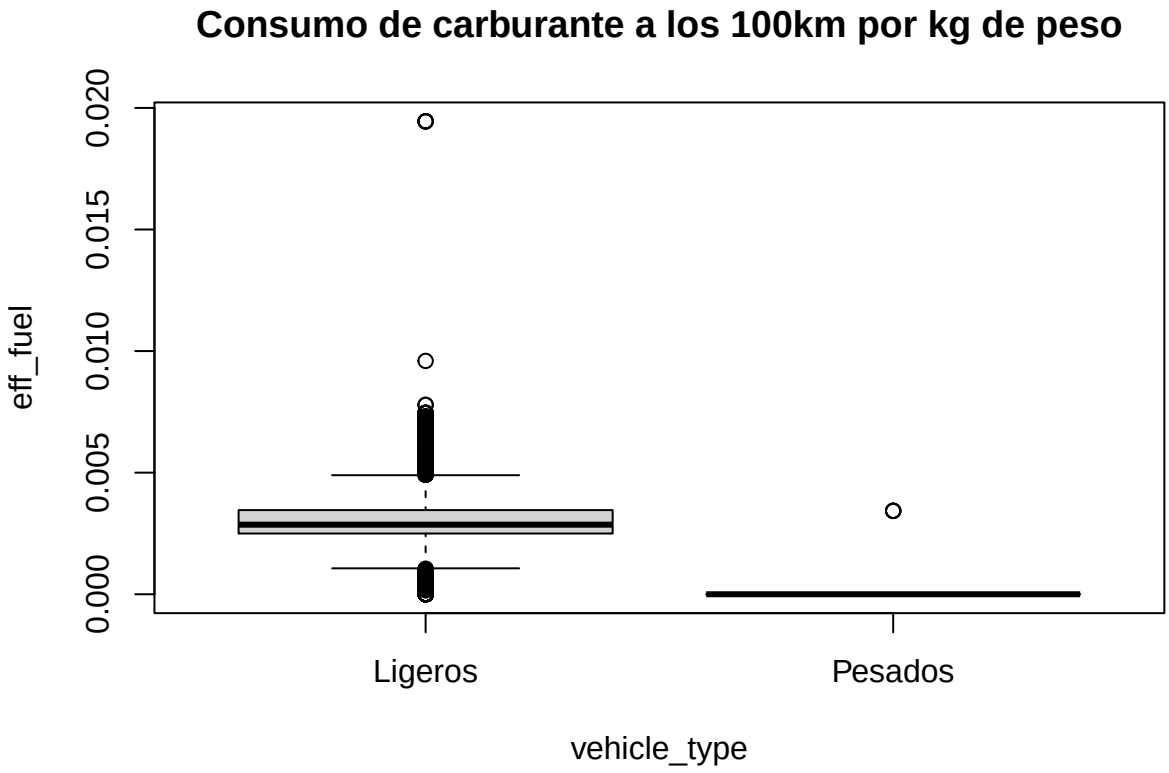
En el caso de los vehículos eléctricos, nos encontramos con que las únicas correlaciones con el consumo eléctrico estadísticamente significativas son las de los atributos de tamaño y el peso del vehículo. En este caso, y a pesar de ser correlaciones muy moderadas, el signo positivo de los cocientes resulta más intuitivo ya que relaciona un mayor tamaña y peso con un mayor consumo.

4.3.2. ¿Qué vehículos són más eficientes?

Como hemos podido observar en el análisis anterior, el peso es uno de los factores que más influye en el consumo de combustible de un vehículo (el que más influye si excluímos la relación consumo-emisiones). Por ese motivo vamos a hacer el ejercicio de estudiar qué grupo de vehículos (ligeros/pesados) es más eficiente, onteniendo un parámetro de consumo de combustible/electricidad cada 100 km **por cada kilogramo de peso**.

```
# Obtenemos la nueva variable
fuel$eff_fuel = fuel$avg_wltp_consumption_l_100km / fuel$gross_vehicle_weight_rating_kg
electricos$eff_elec = electricos$electric_consumption_kwh_100km / electricos$gross_vehicle_weight_ratin



eff_elec_ligeros = electricos$eff_elec[electricos$vehicle_type == 'Ligeros']
eff_fuel_ligeros = fuel$eff_fuel[fuel$vehicle_type == 'Ligeros']
eff_elec_pesados = electricos$eff_elec[electricos$vehicle_type == 'Pesados']
eff_fuel_pesados = fuel$eff_fuel[fuel$vehicle_type == 'Pesados']
```

```
# Exploración visual previa al análisis

boxplot( eff_fuel ~ vehicle_type, data=fuel, main="Consumo de carburante a los 100km por kg de peso")
```



Una exploración gráfica preliminar parece apuntar a que los vehículos pesados consumen menos combustible por kilogramo. Para llevar a cabo este análisis adecuadamente, vamos a realizar un contraste de hipótesis sobre dos muestras. Por lo tanto:

· *Hipótesis nula*: no hay una diferencia significativa entre la eficiencia de los dos grupos. · *Hipótesis alternativa*: el consumo de los vehículos ligeros es mayor que la de los vehículos pesados.

Queremos estudiar si la eficiencia de los coches ligeros es mayor, por lo que estamos ante un **test unilateral por la derecha**.

Podríamos volver aplicar algún test de normalidad, pero al tratarse de un dataset con tamaño de muestra n>30, el procedimiento de cálculo de un intervalo de confianza será aplicable a pesar de no seguir una distribución normal (teorema del límite central).

```
var.test(eff_fuel_ligeros, eff_fuel_pesados)
```

```
##
##  F test to compare two variances
##
## data:  eff_fuel_ligeros and eff_fuel_pesados
## F = 29.541, num df = 12662, denom df = 1204, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  27.13131 32.07064
## sample estimates:
## ratio of variances
##           29.54099
```

```
var.test(eff_elec_ligeros, eff_elec_pesados)
```

```
##
##  F test to compare two variances
##
## data:  eff_elec_ligeros and eff_elec_pesados
## F = 395.78, num df = 1759, denom df = 80, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  281.1031 531.7928
## sample estimates:
## ratio of variances
##           395.7825
```

En ambos casos vemos que las varianzas entre muestras no son homogéneas. Ya estamos preparados para aplicar el test adecuado: *t-student* para muestras independientes.

```
t.test(eff_fuel_ligeros, eff_fuel_pesados, var.equal =FALSE, alternative = 'greater')
```

```
##
##  Welch Two Sample t-test
##
## data:  eff_fuel_ligeros and eff_fuel_pesados
## t = 261.31, df = 9984.8, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.002881077         Inf
## sample estimates:
##    mean of x    mean of y
## 2.910710e-03 1.138115e-05
```

```
t.test(eff_elec_ligeros, eff_elec_pesados, var.equal = FALSE, alternative = 'greater')
```

```
##
##  Welch Two Sample t-test
##
## data:  eff_elec_ligeros and eff_elec_pesados
## t = 3.1632, df = 1835.8, p-value = 0.000793
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.01265039        Inf
## sample estimates:
##   mean of x   mean of y
## 0.034581187 0.008212084
```

En ambos casos, p-valores inferiores al nivel establecido de 0.05 nos permiten rechazar las hipótesis nulas, y confirmar las hipótesis establecidas en el análisis gráfico preliminar: los vehículos ligeros (motocicletas, coches, furgonetas) consumen más combustible y electricidad por kilogramo de peso que los vehículos pesados (camiones y autobuses).

4.3.4. Modelo de regresión lineal

Llegados a este punto, se considera que un modelo de regresión lineal que sea capaz de predecir las emisiones de CO2 de los vehículos a combustible a partir de sus características básicas puede ser una herrmaiento útil para los consumidores, especialmente cuando encontrar información específica de ciertos modelos puede ser una tarea difícil (muchas veces los fabricantes se limitan a aportar el índice de economía y no los valores concretos).

Vamos a excluír la variable del fabricante simplemente por inteligelibilidad del modelo, ya que es una variable categórica con más de 150 valores que tomarían coeficientes independientes. Consideraremos añadirla en caso de necesitar mejorar el ajuste.

```
# Construímos en modelo lineal con todos los atributos restantes después de la selección y limpieza
model = lm(avg_wltp_emissions_gCO2_km ~ . -make, data=fuel)

summary(model)
```

```
##
## Call:
## lm(formula = avg_wltp_emissions_gCO2_km ~ . - make, data = fuel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -504.30   -4.91    0.80    6.21  645.82
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       4.024e+00  3.070e+00   1.311  0.18990
## transmissionM                     2.766e+00  2.946e-01   9.387  < 2e-16 ***
## transmissionSC                    3.531e+00  2.832e+00   1.247  0.21257
## engine_displacement_cm3           2.529e-03  2.597e-04   9.735  < 2e-16 ***
## power_cv                          9.530e+01  4.070e+01   2.341  0.01923 *
## power_ice_kw                     -1.295e+02  5.530e+01  -2.343  0.01916 *
## avg_wltp_consumption_l_100km      2.621e+01  1.134e-01 231.163  < 2e-16 ***
## length_mm                        -1.890e-03  2.822e-04  -6.696 2.22e-11 ***
## width_mm                          1.658e-02  1.794e-03   9.240  < 2e-16 ***
## height_mm                        -5.245e-03  6.470e-04  -8.107 5.64e-16 ***
## gross_vehicle_weight_rating_kg   -1.801e-04  9.304e-05  -1.936  0.05294 .
## total_seating                     8.457e-02  2.897e-02   2.920  0.00351 **
## fuel_economy_indexB               9.763e+00  6.675e-01  14.627  < 2e-16 ***
## fuel_economy_indexC               1.280e+01  6.579e-01  19.459  < 2e-16 ***
## fuel_economy_indexD               1.648e+01  6.985e-01  23.589  < 2e-16 ***
## fuel_economy_indexE               1.945e+01  7.819e-01  24.880  < 2e-16 ***
## fuel_economy_indexF               2.143e+01  9.384e-01  22.836  < 2e-16 ***
## fuel_economy_indexG               2.413e+01  1.071e+00  22.529  < 2e-16 ***
## fuel_economy_indexSin clasificación  9.723e+00  7.464e-01  13.026  < 2e-16 ***
## vehicle_typePesados              -2.140e+01  1.395e+00 -15.343  < 2e-16 ***
## engineHíbridos                   -3.089e+00  4.465e-01  -6.918 4.79e-12 ***
## eff_fuel                         -1.369e+04  2.364e+02 -57.938  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.89 on 13846 degrees of freedom
## Multiple R-squared:  0.9494, Adjusted R-squared:  0.9493
```

```
## F-statistic: 1.237e+04 on 21 and 13846 DF,  p-value: < 2.2e-16
```

Podemos observar como, para los atributos categóricos, se han considerado los valores que no aparecen (por ejemplo la clase A de eficiencia energética) como las referencias integradas en el intercepto, y los cocientes generados en nuestro summary representan la diferencia promedio de esos valores referencia con los valores mostrado. Por ejemplo, observando el valor de los coefientes podemos ver que cuánto peor es la clase (E, F, G), mayor es su contribución a la emisión de CO2 **en comparación** con la clase A.

Los valores de Pr(>|t|) menores que 0.05 nos indican que los atributos significativos para las emisiones. Sorprendentemente, el peso parece ser un factor no significiativo.

Si nos fijamos en el valor absoluto de los coefientes, parece que el valor de eficiencia (consumo por 100km por kg de peso) es claramente el factor más influyente a la hora de calcular las emisiones del vehículo. Su signo negativo indica que a mayor eficiencia, menores emisiones.

Colinealidad

Para comprobar la colinealidad en el modelo (presencia de atributos redundantes ya que están correlacionados entre sí) utilizaremos los valores del factor de inflación de la varianza (FIV o VIF en inglés), que mide la fuerza de las correlaciones entre las variables independientes. El estándar a la hora de interpretar los valores obtenidos es el siguiente:

- Menos de 1: Sin correlación • 1-5: Correlación moderada • Más de 5: Correlación severa

Fuente: https://www.projectpro.io/recipes/check-multicollinearity-r

```
vif(model)
```

```
##                                       GVIF Df GVIF^(1/(2*Df))
## transmission                  1.461804e+00  2        1.099568
## engine_displacement_cm3       1.815235e+01  1        4.260557
## power_cv                      8.610765e+08  1    29344.104431
## power_ice_kw                  8.610745e+08  1    29344.071119
## avg_wltp_consumption_l_100km  6.010287e+00  1        2.451589
## length_mm                     1.343939e+01  1        3.665978
## width_mm                      8.881643e+00  1        2.980209
## height_mm                     1.058161e+01  1        3.252939
## gross_vehicle_weight_rating_kg 1.822585e+01  1        4.269174
## total_seating                 1.541263e+00  1        1.241476
## fuel_economy_index            6.191034e+00  7        1.139081
## vehicle_type                  9.644759e+00  1        3.105601
## engine                        1.543334e+00  1        1.242310
## eff_fuel                      6.007242e+00  1        2.450968
```

Cómo podía ser previsible, los atributos de potencia (general y del sistema de refrigeración) están extremadamente correlacionados. Se decide mantener el valor que tiene un FIV ligeramente menor.

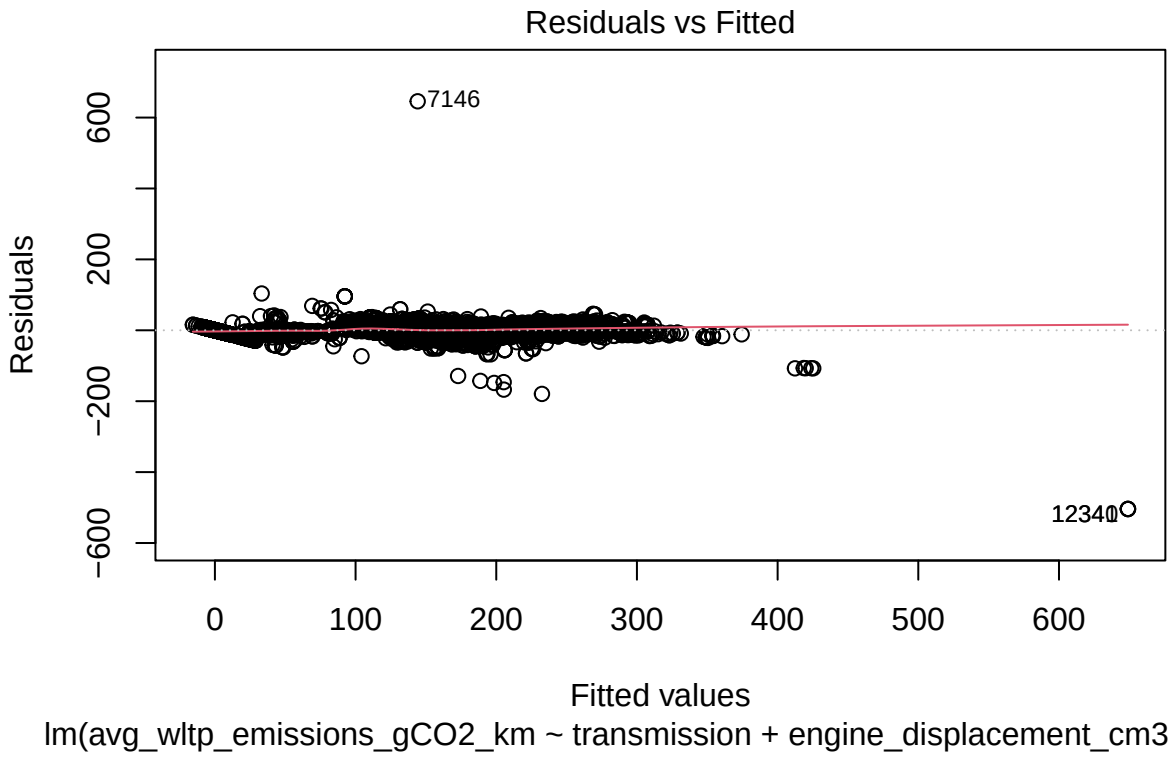
```
# Generamos el modelo final solo con las variables significativas y no redundantes
model = update(model, . ~ . -make -gross_vehicle_weight_rating_kg -power_cv)

summary(model)$r.squared
```

```
## [1] 0.9493429
```

El coeficiente de determinación de nuestro modelo final (R2) presenta un valor muy positivo (0.95).

Diagnosis del modelo generado

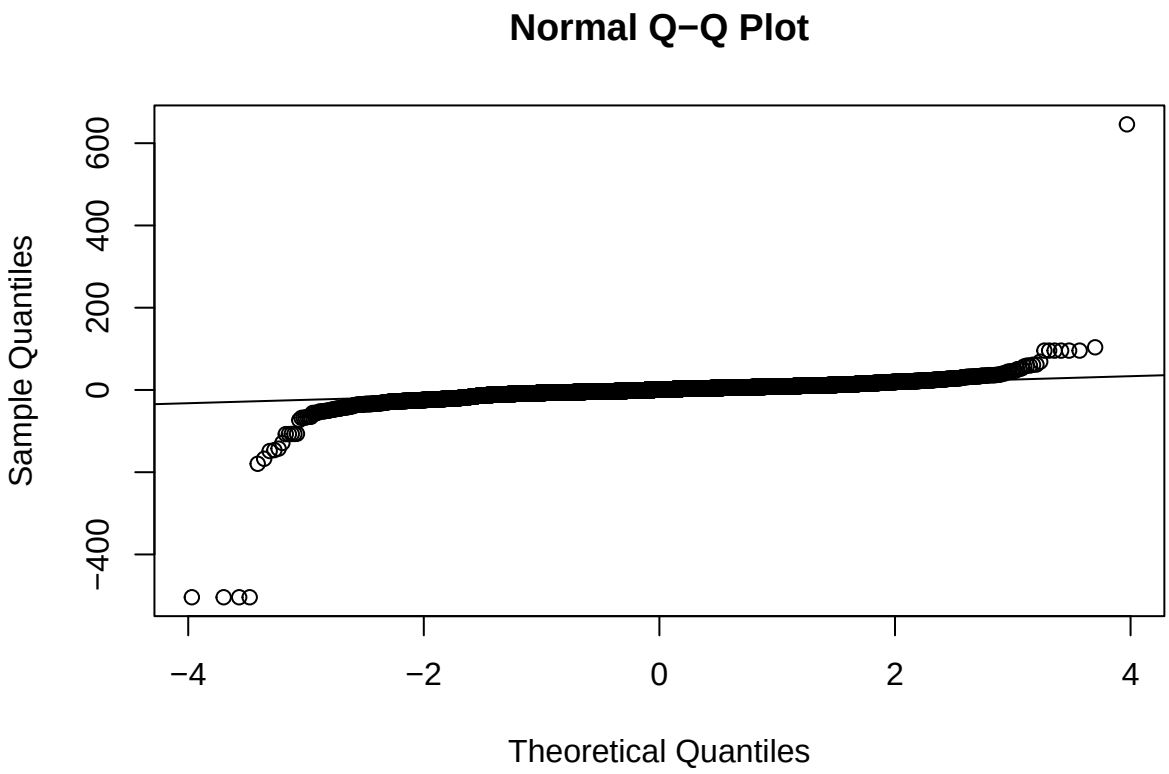
```
# Valores ajustados vs residuos
plot(model, which=1)
```



Un gráfico de valores ajustados contra residuos es una herramienta útil para evaluar el rendimiento de un modelo de regresión. Los valores ajustados son los valores pronosticados por el modelo para cada punto de datos, mientras los residuos son la diferencia entre los valores observados y los fitted values.

Si el modelo está funcionando bien, esperaríamos que los residuals se distribuyan de manera aleatoria alrededor de cero, lo que indicaría que el modelo está haciendo predicciones precisas. Si por el contrario, como en nuestro caso, se detecta una (ligera) tendencia, significa que el modelo está sobreestimando o subestimando los fitted values. En este caso la tendencia es levemente positiva así que el modelo está sobreestimando. Sería podría añadir términos adicionales para tratar de corregirlo.

```
# Gráfico cuantil-cuantil
qqnorm(model$residuals)
qqline(model$residuals)
```



Un Q-Q plot es una herramienta útil para evaluar si los residuales del modelo siguen una distribución normal. Se plotean los cuantiles de los residuos observados en el eje X contra los cuantiles teóricos de una distribución normal en el eje Y. Por lo tanto, si los residuos siguen una distribución normal el gráfico debería mostrar una línea recta.

Si por el contrario, el gráfico tiene una curvatura se interpreta que los residuales no siguen una distribución normal, lo que podría afectar a la precisión y confiabilidad de los resultados.

Nuestro gráfico se acerca mucho a una situación ideal, aunque las ligeras curvaturas en los extremos indican que los datos están un poco sesgados en esa zona (distribución con colas más pesadas).

Predicción

Testeamos el modelo generado con un vehículo elegido al azar de entre nuestra cohorte.

```
# Cogemos un vehículo al azar y guardamos sus emisiones
muestra = fuel[765,]
real_emis = muestra['avg_wltp_emissions_gCO2_km']

muestra = muestra[, !(names(muestra) %in% c("avg_wltp_emissions_gCO2_km", "make", "power_cv", "gross_vel

# Predicción
pred = predict(model, newdata=muestra)

print(paste0("Valor real: ", real_emis))
```

```
## [1] "Valor real: 41"
```

```r
print(paste0("Predicción: ", pred))
```

```
## [1] "Predicción: 47.882162433506"
```

Vemos que, aunque no conseguimos una predicción perfecta, nuestro modelo lineal se aproxima al valor real.

5. Conclusiones

En esta práctica se ha trabajado un un dataset que reúne las emisiones de C02 y los consumos de combustible y consumos eléctricos de la mayoría de los modelos de automóviles a la venta en España en el año 2022. A este dataset, previa limpieza, se le han aplicado tres análisis estadísticos que han permitido obtener información valiosa.

Con un análisis de correlaciones se han determinado los atributos con una mayor influencia en los consumos eléctricos y de carburante. En el caso de consumo de carburante, las emisiones son un factor decisivo, mientras que en los coches eléctricos lo son el peso y las dimensiones.

Por último, se ha generado un modelo de regresión lineal para calcular las emisiones de CO2 a partir de las características de los vehículos, obteniendo un coficiente de determinación del 94%.