Portafolio A00829925

Diego Rodriguez

2023-08-25

Exploración y preparación de la base de datos

Lectura de datos y Exploración

Con base en varias encuestas de mercado, la consultora ha recopilado un gran conjunto de datos de diferentes tipos de automóviles en el mercado estadounidense que presenta en la siguiente base de datos:

```
df = read.csv("precios_autos.csv")
```

Exploración de variables numéricas

```
# Exploración de variables numéricas:
numeric_columns = sapply(df, is.numeric)
summary(df[, numeric_columns])
```

```
##    symboling         wheelbase        carlength        carwidth
##  Min.   :-2.0000   Min.   : 86.60   Min.   :141.1   Min.   :60.30
##  1st Qu.: 0.0000   1st Qu.: 94.50   1st Qu.:166.3   1st Qu.:64.10
##  Median : 1.0000   Median : 97.00   Median :173.2   Median :65.50
##  Mean   : 0.8341   Mean   : 98.76   Mean   :174.0   Mean   :65.91
##  3rd Qu.: 2.0000   3rd Qu.:102.40   3rd Qu.:183.1   3rd Qu.:66.90
##  Max.   : 3.0000   Max.   :120.90   Max.   :208.1   Max.   :72.30
##    carheight       curbweight     enginesize        stroke
##  Min.   :47.80   Min.   :1488   Min.   : 61.0   Min.   :2.070
##  1st Qu.:52.00   1st Qu.:2145   1st Qu.: 97.0   1st Qu.:3.110
##  Median :54.10   Median :2414   Median :120.0   Median :3.290
##  Mean   :53.72   Mean   :2556   Mean   :126.9   Mean   :3.255
##  3rd Qu.:55.50   3rd Qu.:2935   3rd Qu.:141.0   3rd Qu.:3.410
##  Max.   :59.80   Max.   :4066   Max.   :326.0   Max.   :4.170
##  compressionratio   horsepower       peakrpm        citympg
##  Min.   : 7.00    Min.   : 48.0   Min.   :4150   Min.   :13.00
##  1st Qu.: 8.60    1st Qu.: 70.0   1st Qu.:4800   1st Qu.:19.00
##  Median : 9.00    Median : 95.0   Median :5200   Median :24.00
##  Mean   :10.14    Mean   :104.1   Mean   :5125   Mean   :25.22
##  3rd Qu.: 9.40    3rd Qu.:116.0   3rd Qu.:5500   3rd Qu.:30.00
##  Max.   :23.00    Max.   :288.0   Max.   :6600   Max.   :49.00
##    highwaympg        price
##  Min.   :16.00   Min.   : 5118
##  1st Qu.:25.00   1st Qu.: 7788
##  Median :30.00   Median :10295
##  Mean   :30.75   Mean   :13277
##  3rd Qu.:34.00   3rd Qu.:16503
##  Max.   :54.00   Max.   :45400
```

Estas medidas estadísticas proporcionan información descriptiva sobre las características de carros y sus precios en una base de datos. Aquí está la interpretación de cada una de ellas:

symboling: Esta variable parece representar una medida de seguridad o riesgo asociada al vehículo, posiblemente según algún sistema de clasificación. Los valores varían de -2 a 3, donde valores más bajos indican vehículos más seguros o menos riesgosos, y valores más altos indican vehículos menos seguros o más riesgosos.La media de 0.8341 sugiere un promedio ligeramente positivo en términos de seguridad en la muestra.

wheelbase: La variable "wheelbase" se refiere a la distancia entre los ejes delantero y trasero del vehículo. Los valores oscilan entre 86.6 y 120.9. Esta medida puede estar relacionada con la estabilidad y el espacio interior del vehículo. Valores mayores podrían indicar más espacio interior. La mediana de 97.00 muestra que la mitad de los vehículos tienen una distancia entre ejes mayor a este valor.

carlength, *carwidth*, *carheight*: Estas variables representan las dimensiones físicas del vehículo en términos de longitud, ancho y altura, respectivamente. Estas medidas son importantes para comprender el tamaño del automóvil y su apariencia general.

curbweight: Esta variable se refiere al peso del vehículo en condiciones de operación normales, es decir, con todos los fluidos y equipos necesarios. Puede ser un indicador importante de la potencia necesaria para mover el vehículo y su eficiencia energética. La mediana de 2414 muestra que la mitad de los vehículos tienen un peso mayor a este valor. El valor máximo de 4066 indica el peso más alto en la muestra.

enginesize: Representa el tamaño del motor del vehículo en términos de su capacidad cúbica. Motores más grandes suelen tener más potencia, lo que puede influir en el rendimiento del vehículo. La mediana de 120.0 muestra que la mitad de los vehículos tienen un tamaño de motor mayor a este valor. La media de 126.9 indica el promedio de los tamaños de motor de los vehículos en el conjunto de datos.

stroke: La variable "stroke" se refiere a la longitud de la carrera del pistón en el motor. Esto puede tener efectos en el rendimiento y la eficiencia del motor. La mediana de 3.290 muestra que la mitad de los vehículos tienen una longitud de carrera mayor a este valor. La media de 3.255 indica el promedio de las longitudes de carrera de los vehículos en el conjunto de datos.

compressionratio: Indica la relación de compresión del motor, que está relacionada con la eficiencia y el rendimiento del motor. Aunque el valor máximo es de 23.00, la media de compresión del motor es apenas de 10.14.

horsepower: Representa la potencia del motor en caballos de fuerza. Esta medida está directamente relacionada con el rendimiento y la velocidad del vehículo. La mediana de 95.0 muestra que la mitad de los vehículos tienen una potencia mayor a este valo, mientras que la media es de 104.1 con un valor máximo de 288 caballos de fuerza.

peakrpm: Es el número de revoluciones por minuto (RPM) en las que el motor produce su potencia máxima. Puede estar relacionado con la respuesta y la velocidad máxima del vehículo.

citympg, *highwaympg*: Estas variables representan el consumo de combustible en millas por galón (MPG) en entornos urbanos y en carretera, respectivamente. Valores más altos indican mayor eficiencia de combustible.

price: Esta es la variable objetivo en el análisis, que representa el precio del vehículo. Los valores varían desde 5118 hasta 45400. Esta variable es esencial para cualquier análisis de precios o modelado predictivo.

Exploración de las variables categóricas

```
# Exploración de variables categóricas:

categoric_columns = sapply(df, is.character)

for (i in names(categoric_columns[categoric_columns == TRUE])[-1]){
  cat("Distribución de la variable", i, ': ')
  print(table(df[i]))
```

```
  cat("\n")
}
```

```
## Distribución de la variable fueltype : fueltype
## diesel    gas
##     20    185
##
## Distribución de la variable carbody : carbody
## convertible     hardtop   hatchback       sedan       wagon
##           6           8          70          96          25
##
## Distribución de la variable drivewheel : drivewheel
## 4wd fwd rwd
##   9 120  76
##
## Distribución de la variable enginelocation : enginelocation
## front  rear
##   202     3
##
## Distribución de la variable enginetype : enginetype
##  dohc dohcv     l   ohc  ohcf  ohcv rotor
##    12     1    12   148    15    13     4
##
## Distribución de la variable cylindernumber : cylindernumber
##  eight   five   four    six  three twelve    two
##      5     11    159     24      1      1      4
```

fueltype: La variable "fueltype" tiene dos categorías únicas: "diesel" y "gas". En la muestra, hay 20 vehículos que funcionan con diésel y 185 vehículos que funcionan con gasolina.

carbody: La variable "carbody" representa el tipo de carrocería de los vehículos. Las categorías únicas son "convertible", "hardtop", "hatchback", "sedan" y "wagon". La distribución muestra que hay 6 convertibles, 8 hardtops, 70 hatchbacks, 96 sedanes y 25 wagones en la muestra.

drivewheel: La variable "drivewheel" indica el tipo de tracción en las ruedas de los vehículos. Las categorías son "4wd" (tracción en las cuatro ruedas), "fwd" (tracción delantera) y "rwd" (tracción trasera). Hay 9 vehículos con tracción en las cuatro ruedas, 120 con tracción delantera y 76 con tracción trasera.

enginelocation: La variable "enginelocation" indica si el motor está ubicado en la parte delantera o trasera del vehículo. La distribución muestra que 202 vehículos tienen el motor en la parte delantera y solo 3 vehículos tienen el motor en la parte trasera.

enginetype: La variable "enginetype" se refiere al tipo de motor de los vehículos. Las categorías son "dohc", "dohcv", "l", "ohc", "ohcf", "ohcv" y "rotor". Hay una distribución variada de tipos de motor en la muestra, con diferentes cantidades en cada categoría.

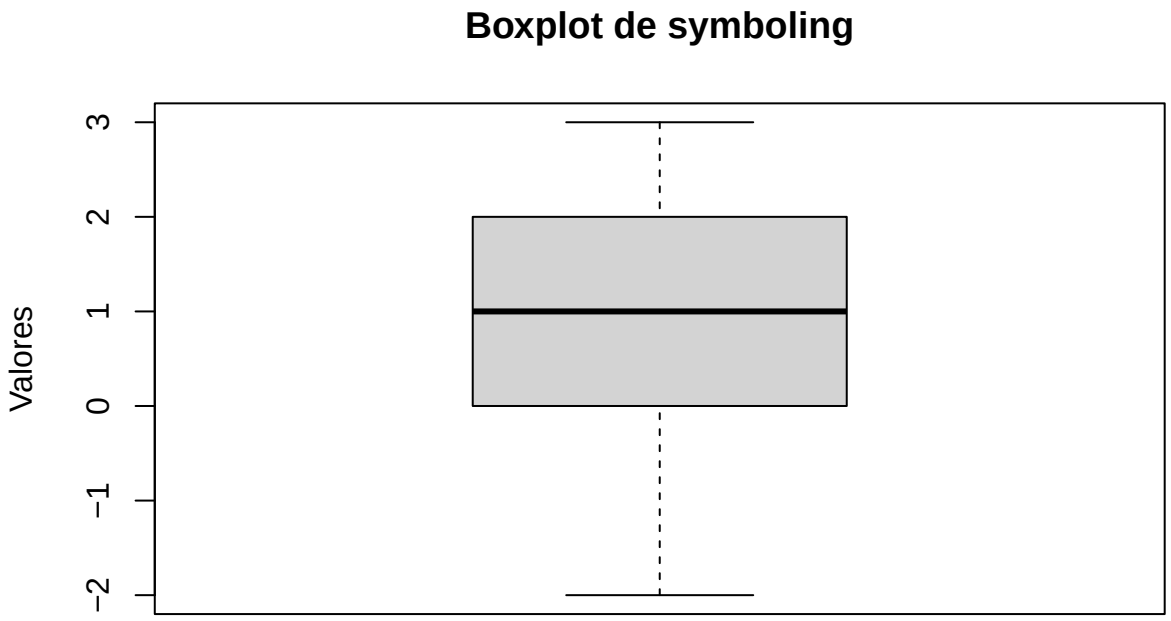
cylindernumber: La variable "cylindernumber" indica la cantidad de cilindros en el motor de los vehículos. Las categorías son "eight", "five", "four", "six", "three", "twelve" y "two". La mayoría de los vehículos tienen motores de cuatro cilindros (159), seguidos de motores de seis cilindros (24) y cinco cilindros (11), entre otros.

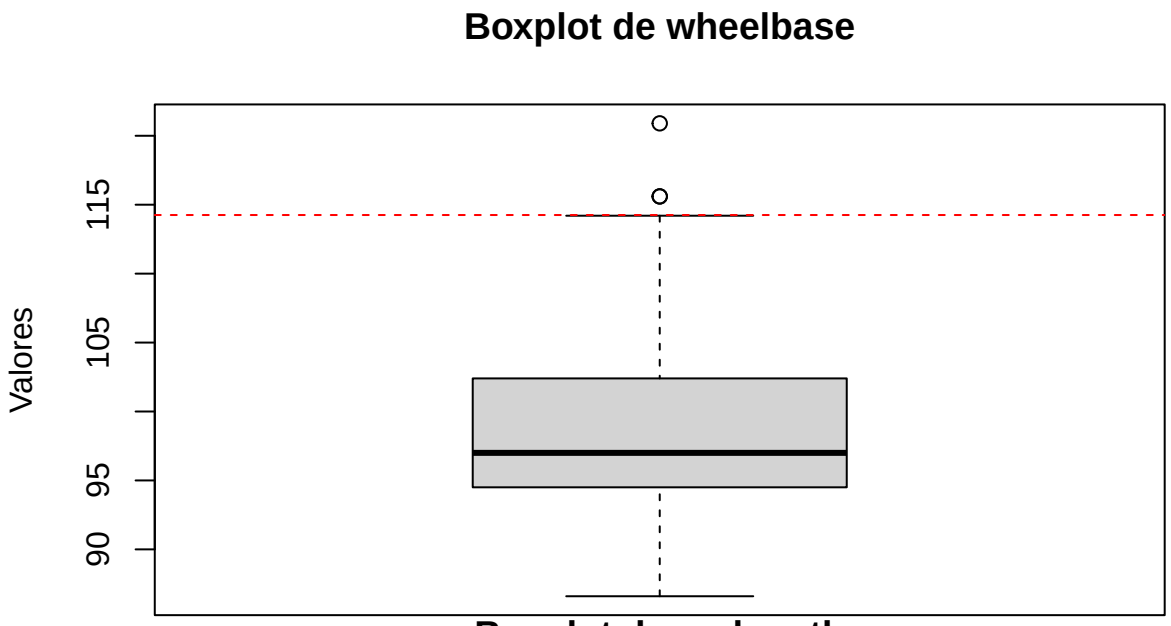
Visualización de los datos

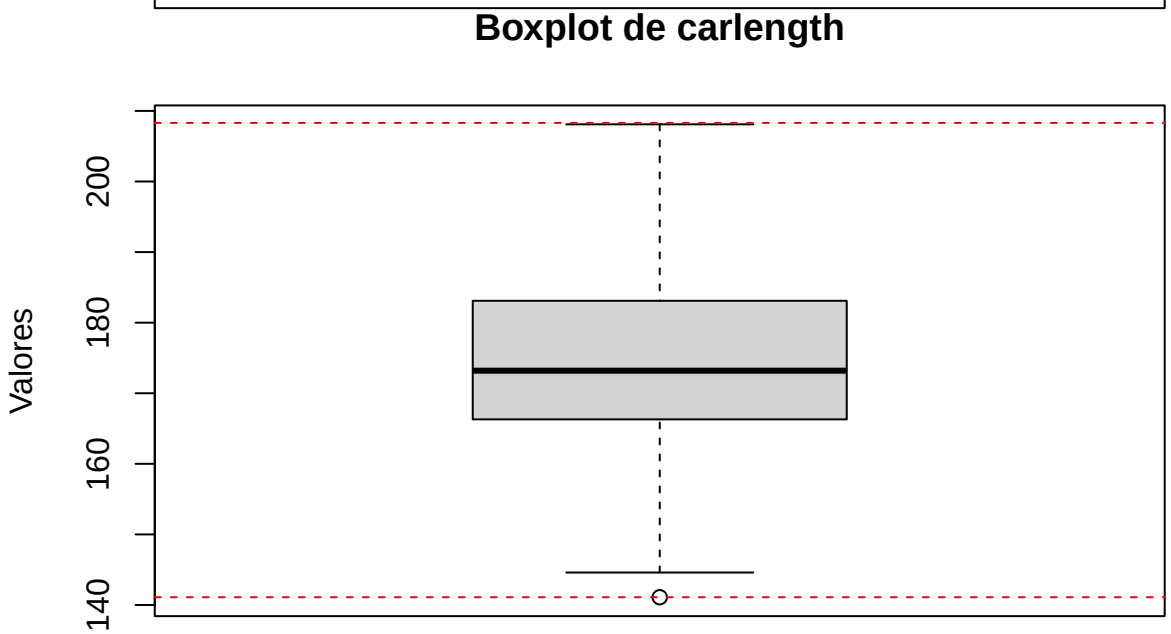
Boxplots

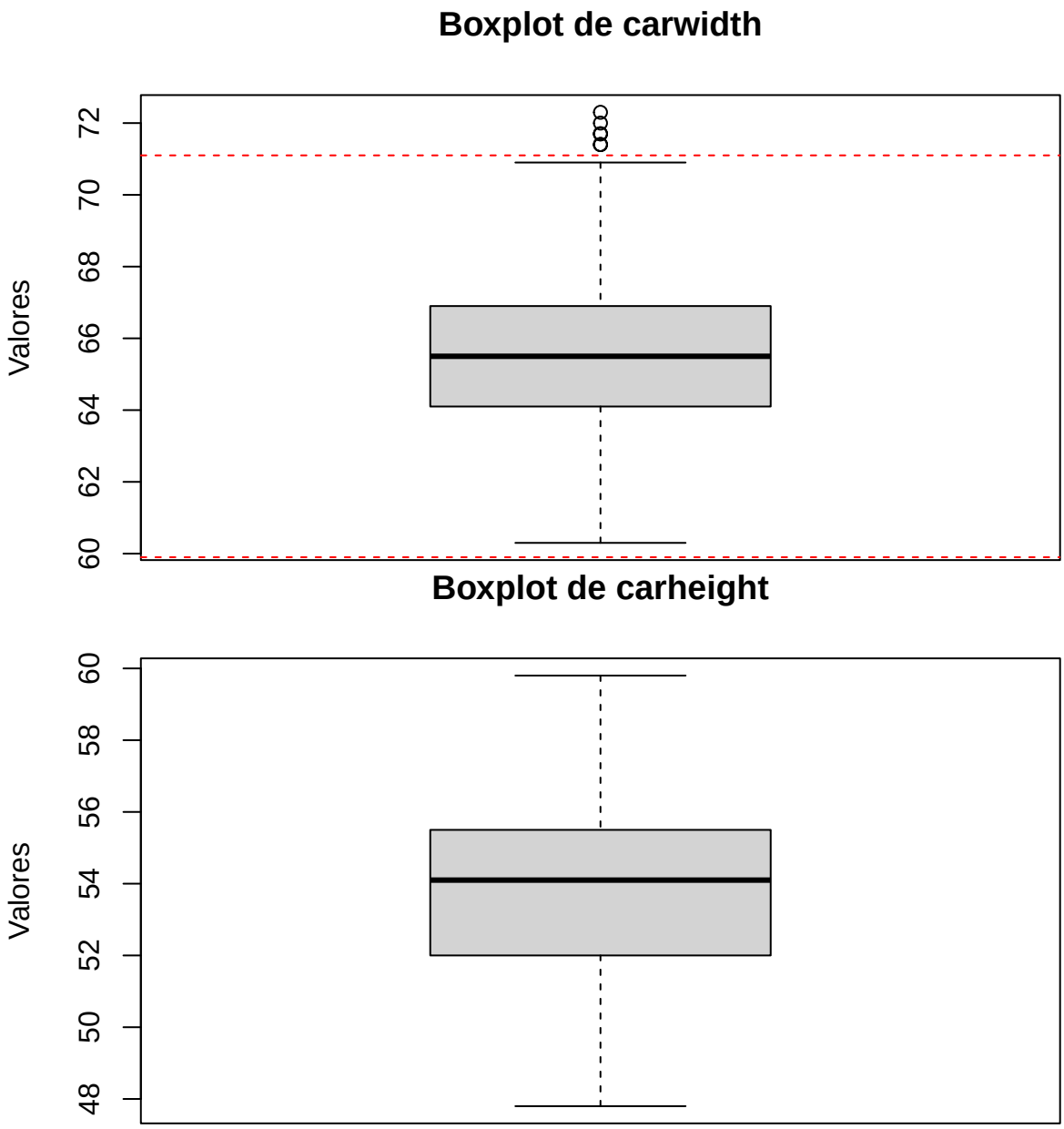
```
#Boxplots

create_boxplot <- function(data, column_name, treshold) {
```

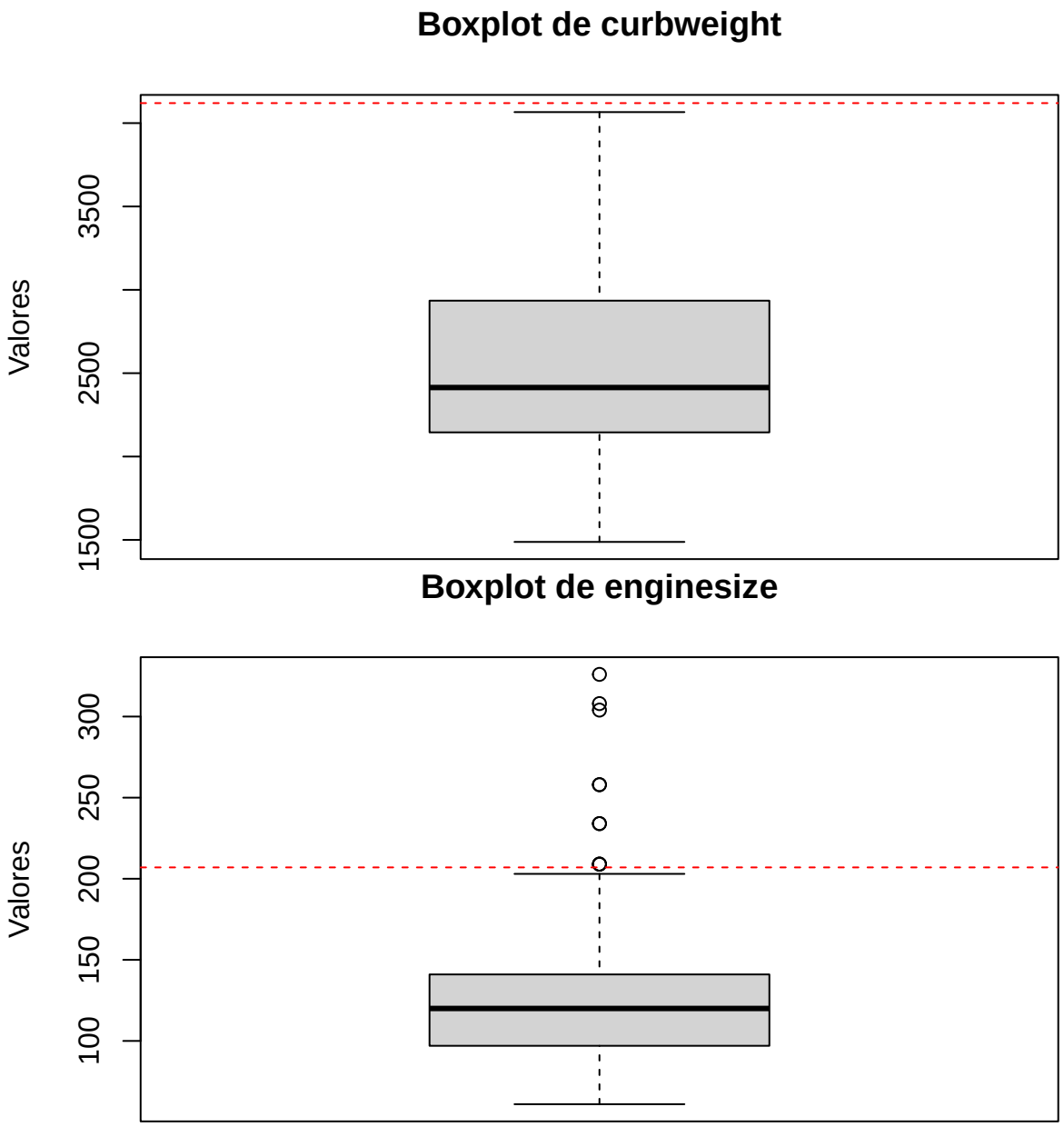
```r
  values <- data[[column_name]]
  values <- values[is.finite(values)]  # Filtrar valores finitos

  quantiles <- quantile(values, probs = c(0.25, 0.5, 0.75), na.rm = TRUE)
  outlier_threshold <- treshold * IQR(values, na.rm = TRUE)

  bp <- boxplot(values, main=paste("Boxplot de", column_name), ylab="Valores", ylim=c(min(values), max(

  # Agregar líneas del outlier threshold al boxplot
  abline(h=quantiles[3] + outlier_threshold, col="red", lty=2)
  abline(h=quantiles[1] - outlier_threshold, col="red", lty=2)

}

for (i in names(numeric_columns[numeric_columns == TRUE])){
  create_boxplot(df, i, 1.5)
}
```

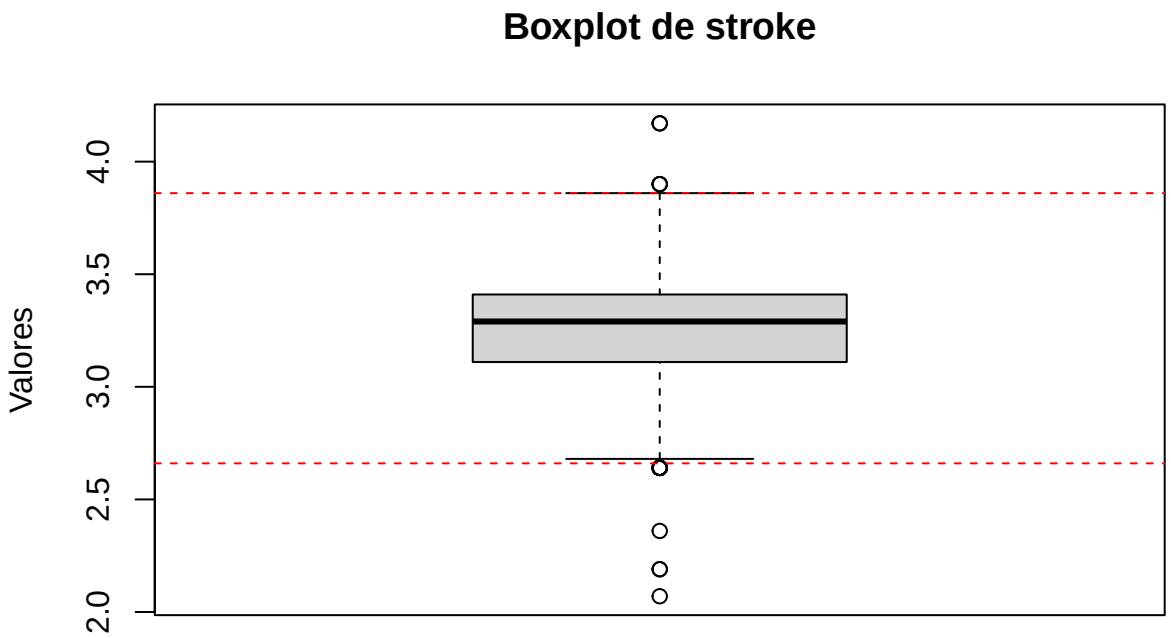


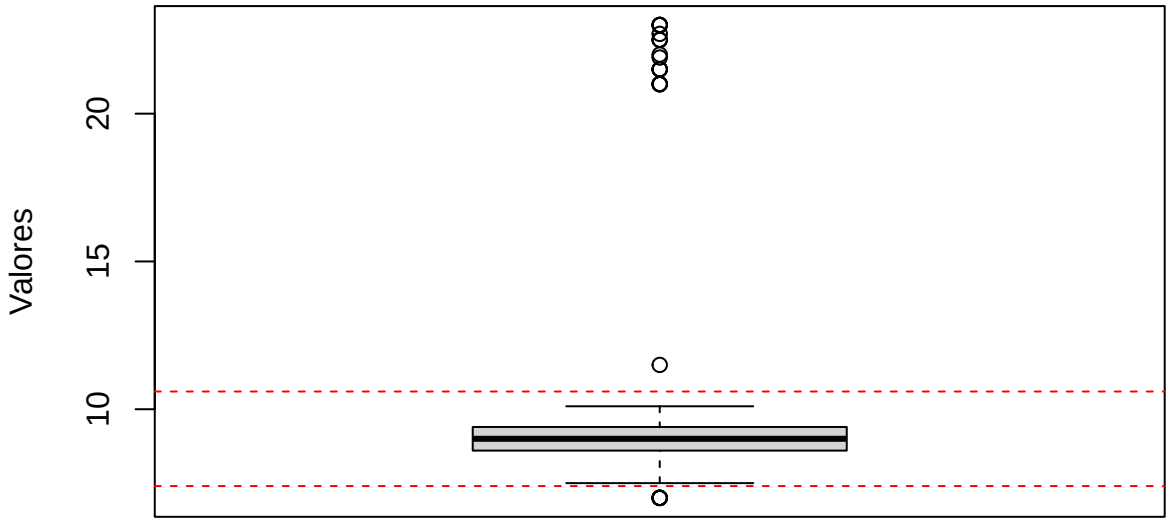

Boxplot de wheelbase
Valores
90
95
105
115

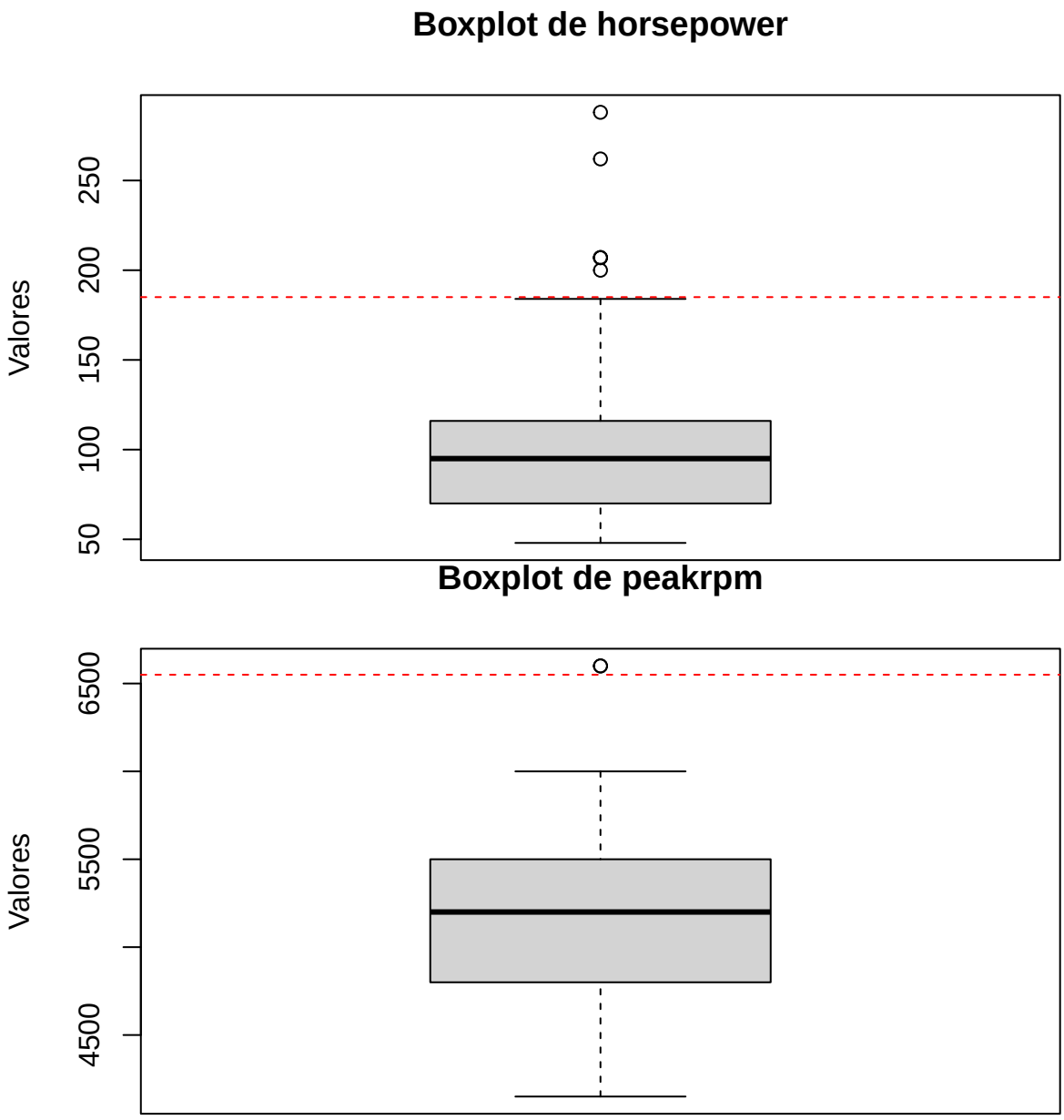

Boxplot de carlength
Valores
140
160
180
200

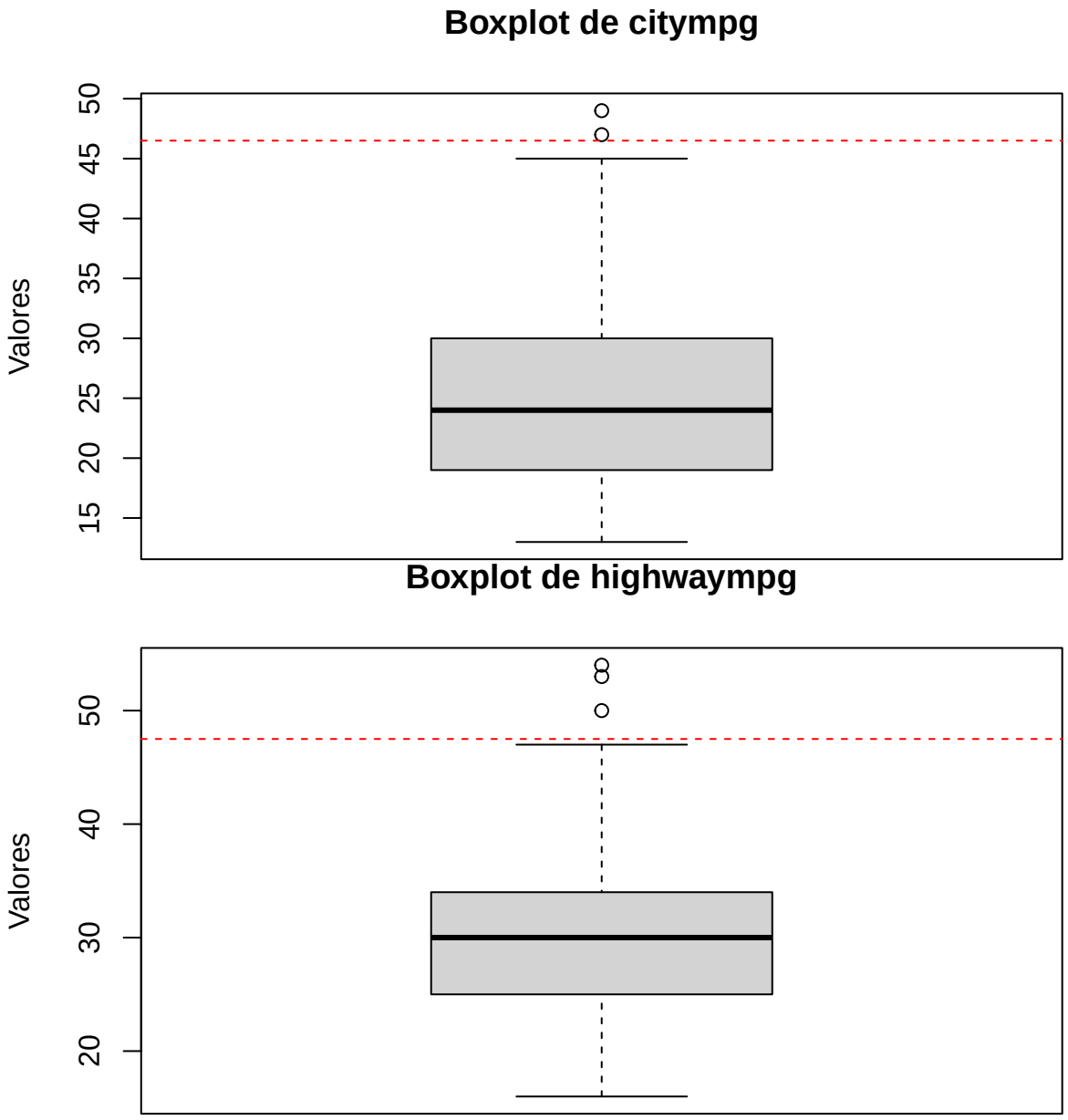

Boxplot de carwidth
Valores
60
62
64
66
68
70
72
Boxplot de carheight
Valores
48
50
52
54
56
58
60

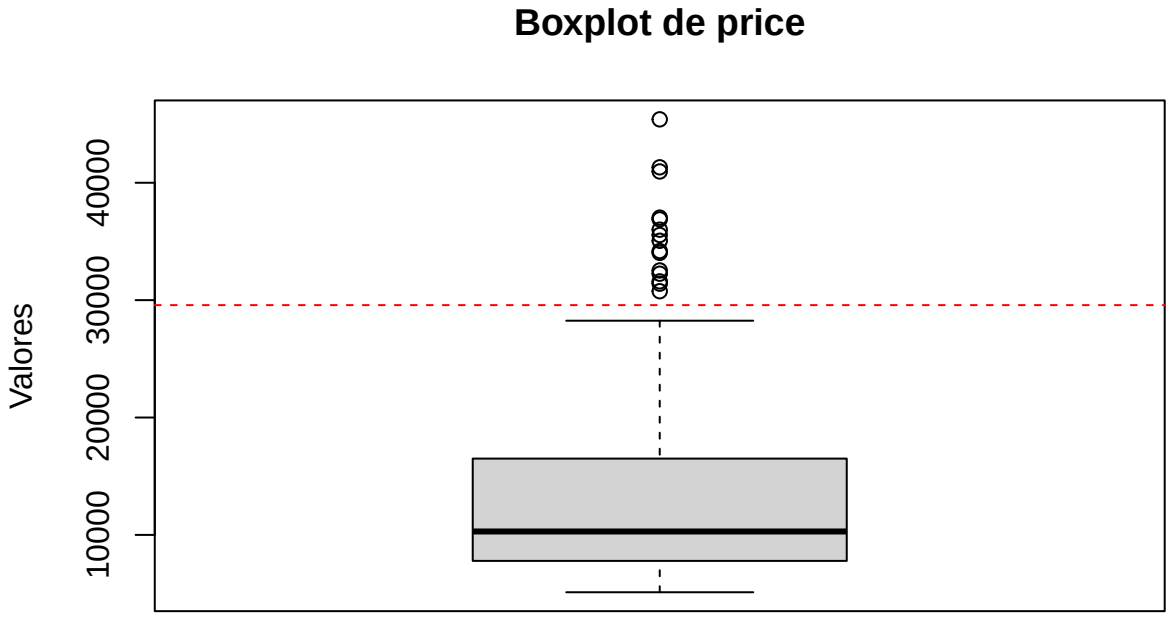

Boxplot de curbweight
Valores
1500
2500
3500
Boxplot de enginesize
Valores
100
150
200
250
300


Boxplot de stroke
Valores
2.0
2.5
3.0
3.5
4.0


Boxplot de compressionratio
Valores
10
15
20


Boxplot de horsepower
Valores
50
100
150
200
250
Boxplot de peakrpm
Valores
4500
5500
6500


Boxplot de citympg
Valores
15
20
25
30
35
40
45
50
Boxplot de highwaympg
Valores
20
30
40
50



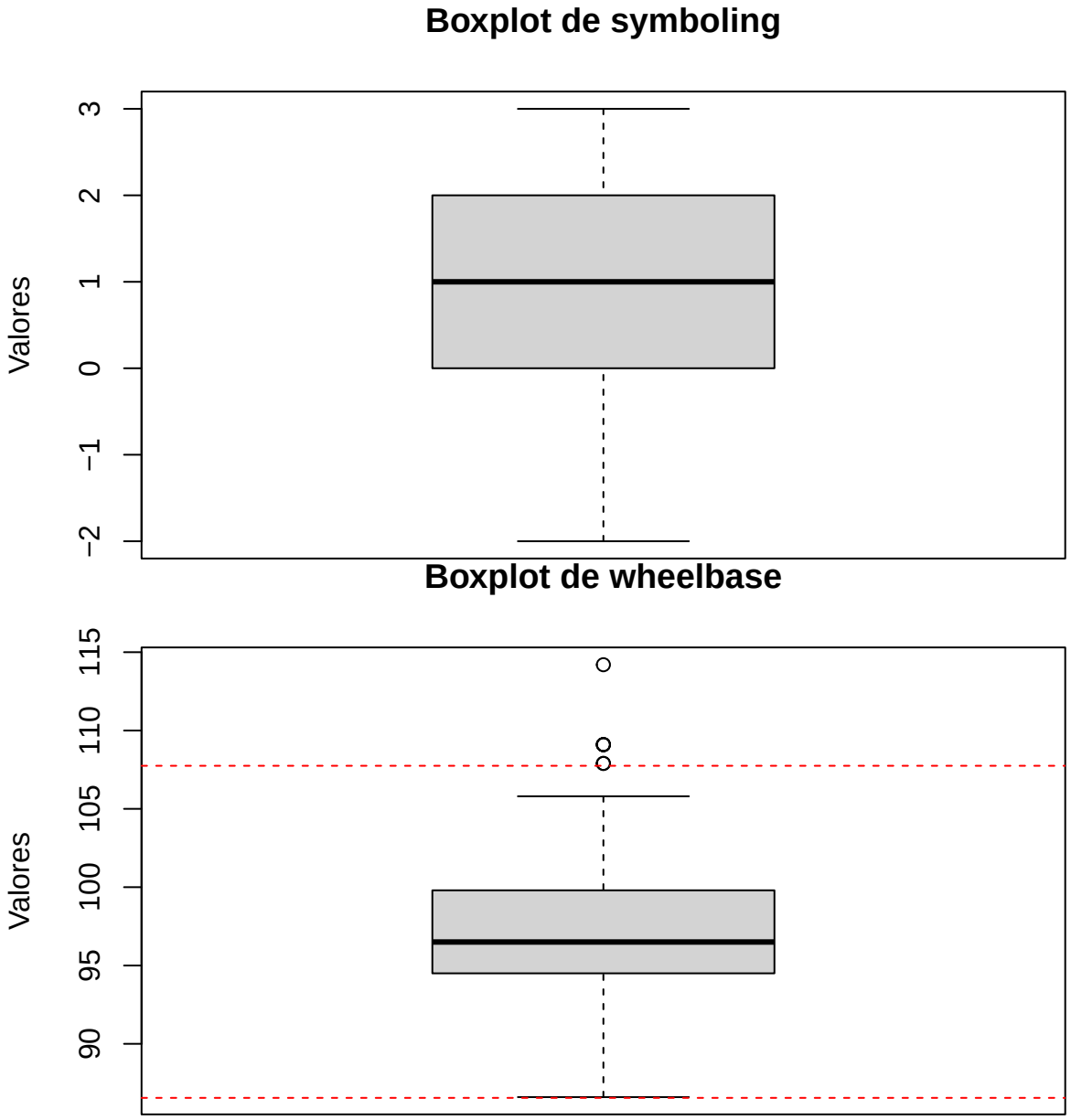
Gracias a las gráficas de boxplot de las variables numéricas podemos identificar la distribución de cuartiles, la media y la existencia de valores outliers en los datos. Por ejemplo identificamos que *wheelbase*, *carwith*, *enginesize*, *stroke*, *compressionratio*, *horsepower* y *price* contienen datos atípicos.

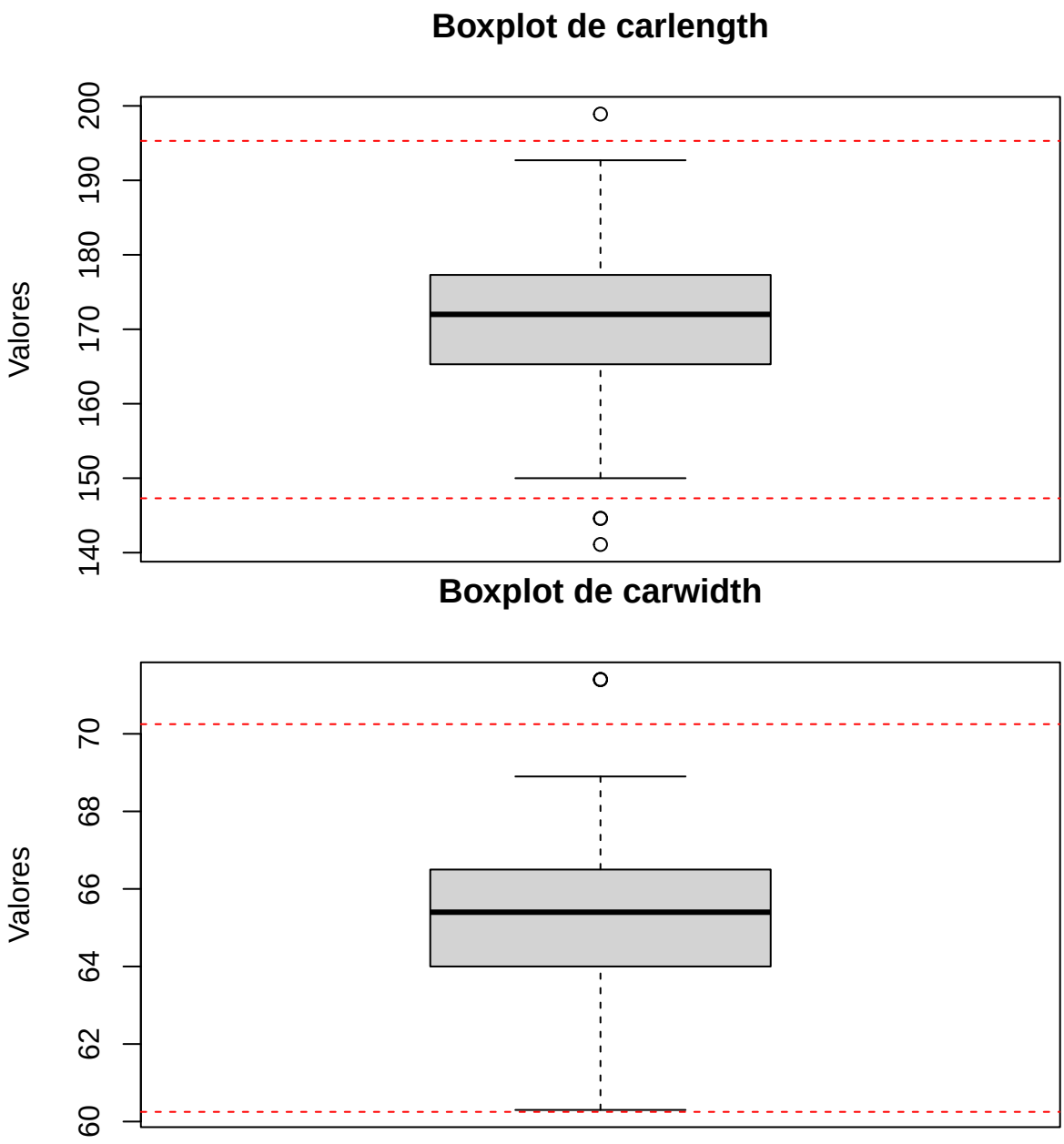
Removiendo outliers

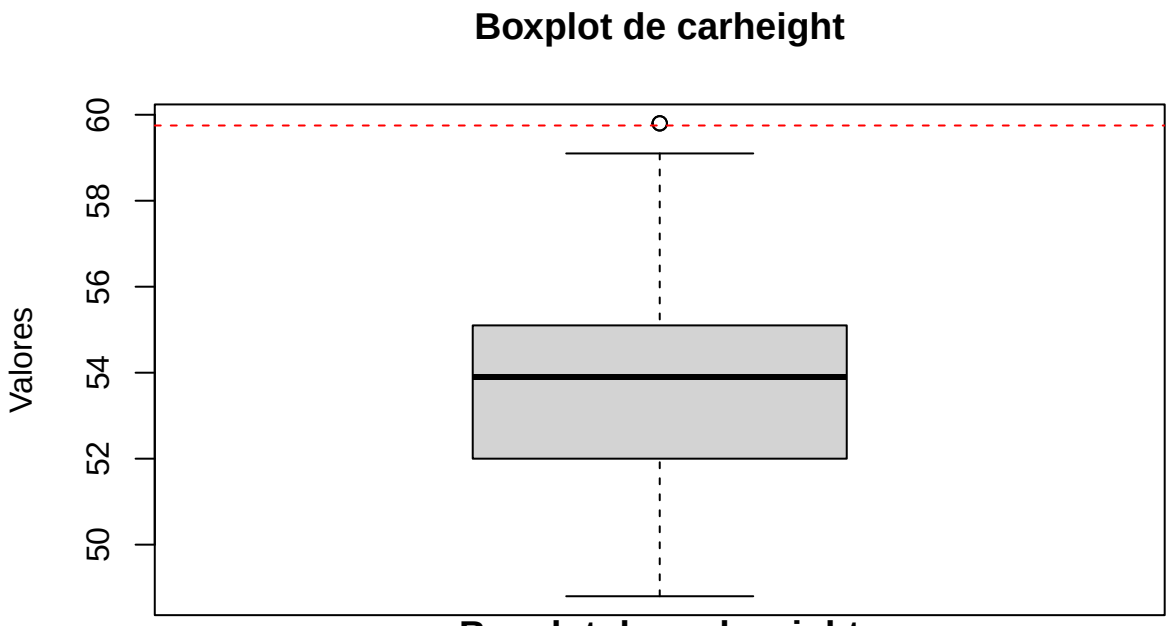
Para este análisis se considera un valor atípico aquellos datos que se alejan más de 1.5 veces el rango intercuartil de los q1 y q3, sin embargo, debido al contexto del problema, se considera que la mayoría de estos valores son posibles y reales, es decir, no errores de captura. Aun así, para que no sesguen ni generen ruido en el análisis se decidió que se eliminarán los valores que se alejen hasta 2 veces el rango intercuartil, con el objetivo de no reducir en gran cantidad la base de datos, pero tampoco mantener datos muy alejados a la media.

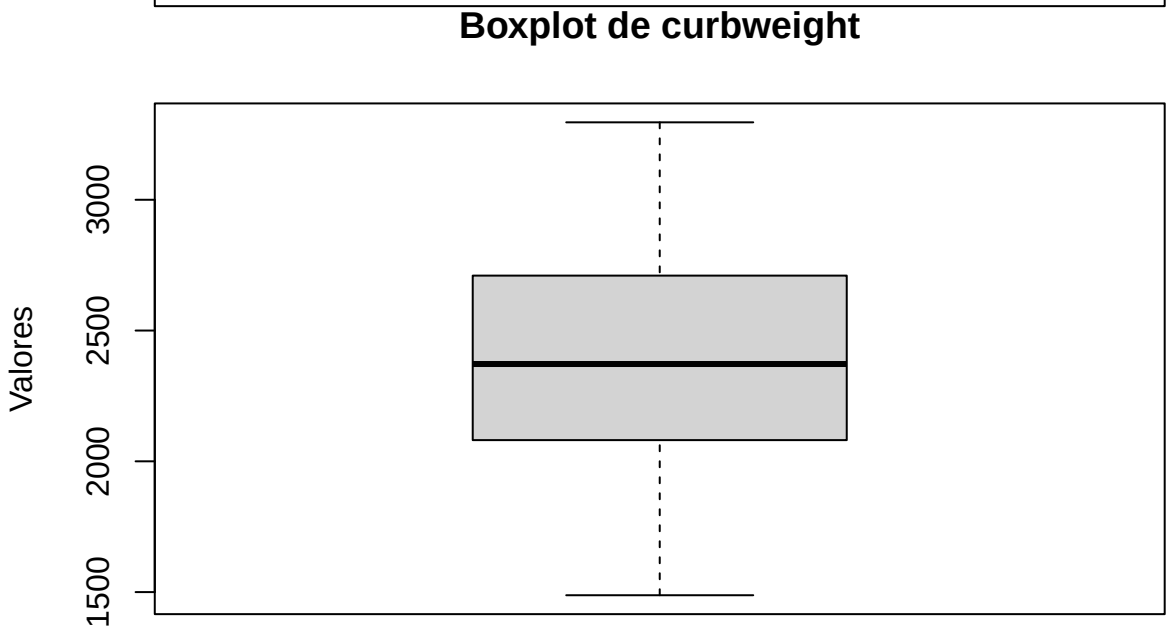
```
remove_outliers_iqr <- function(data, column_name, multiplier) {
  column_values <- data[[column_name]]

  q1 <- quantile(column_values, 0.25, na.rm = TRUE)
  q3 <- quantile(column_values, 0.75, na.rm = TRUE)
  iqr <- q3 - q1

  lower_bound <- q1 - multiplier * iqr
  upper_bound <- q3 + multiplier * iqr


  data_clean <- data[column_values >= lower_bound & column_values <= upper_bound, ]
  return(data_clean)
}

df_clean = df

for (i in c("price", "wheelbase", "carwidth", "enginesize", "stroke", "compressionratio", "horsepower")){
   df_clean = remove_outliers_iqr(df_clean, i, 2)
}

for (i in names(numeric_columns[numeric_columns == TRUE])){
```

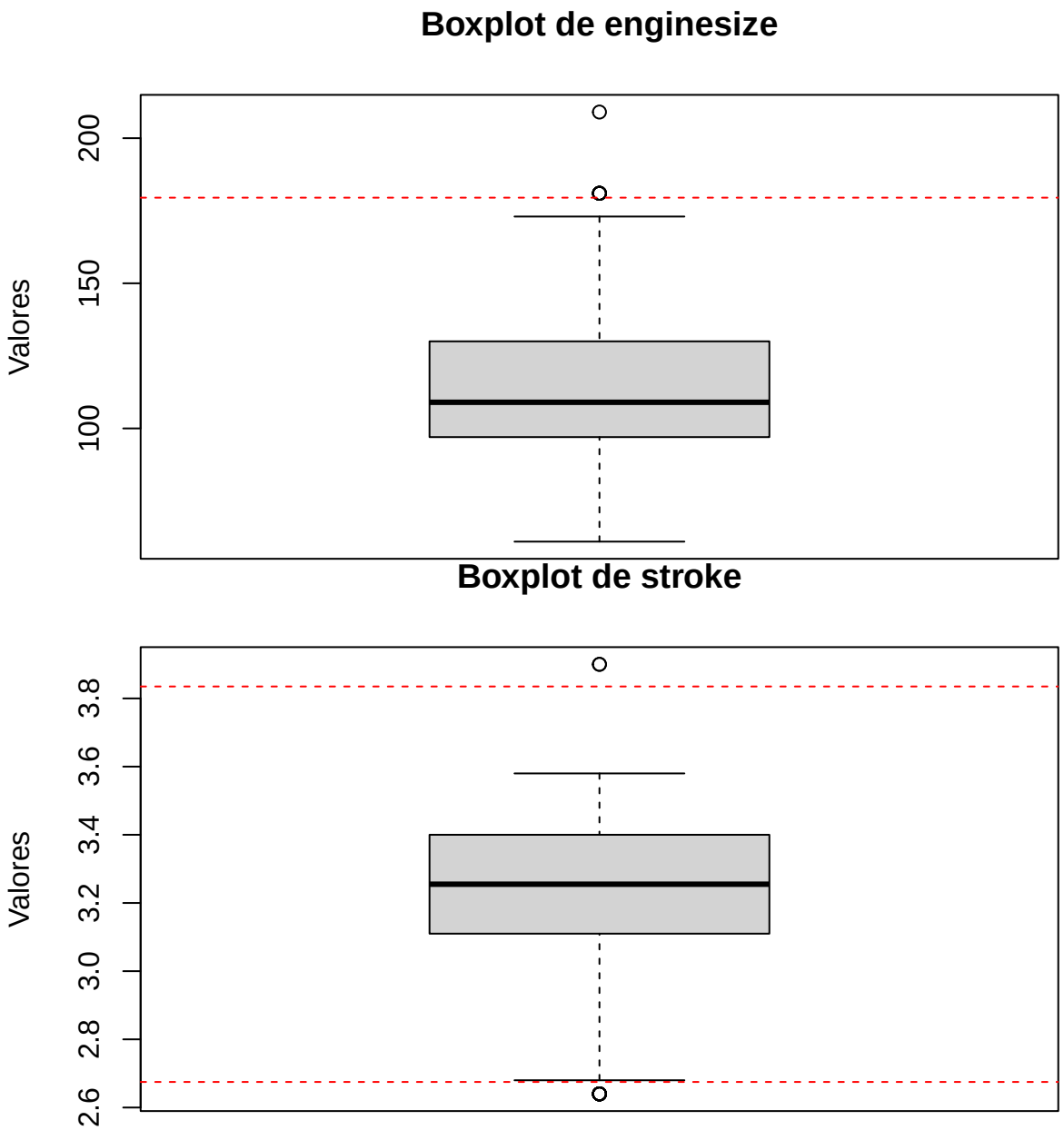
```
  create_boxplot(df_clean, i, 1.5)
}
```

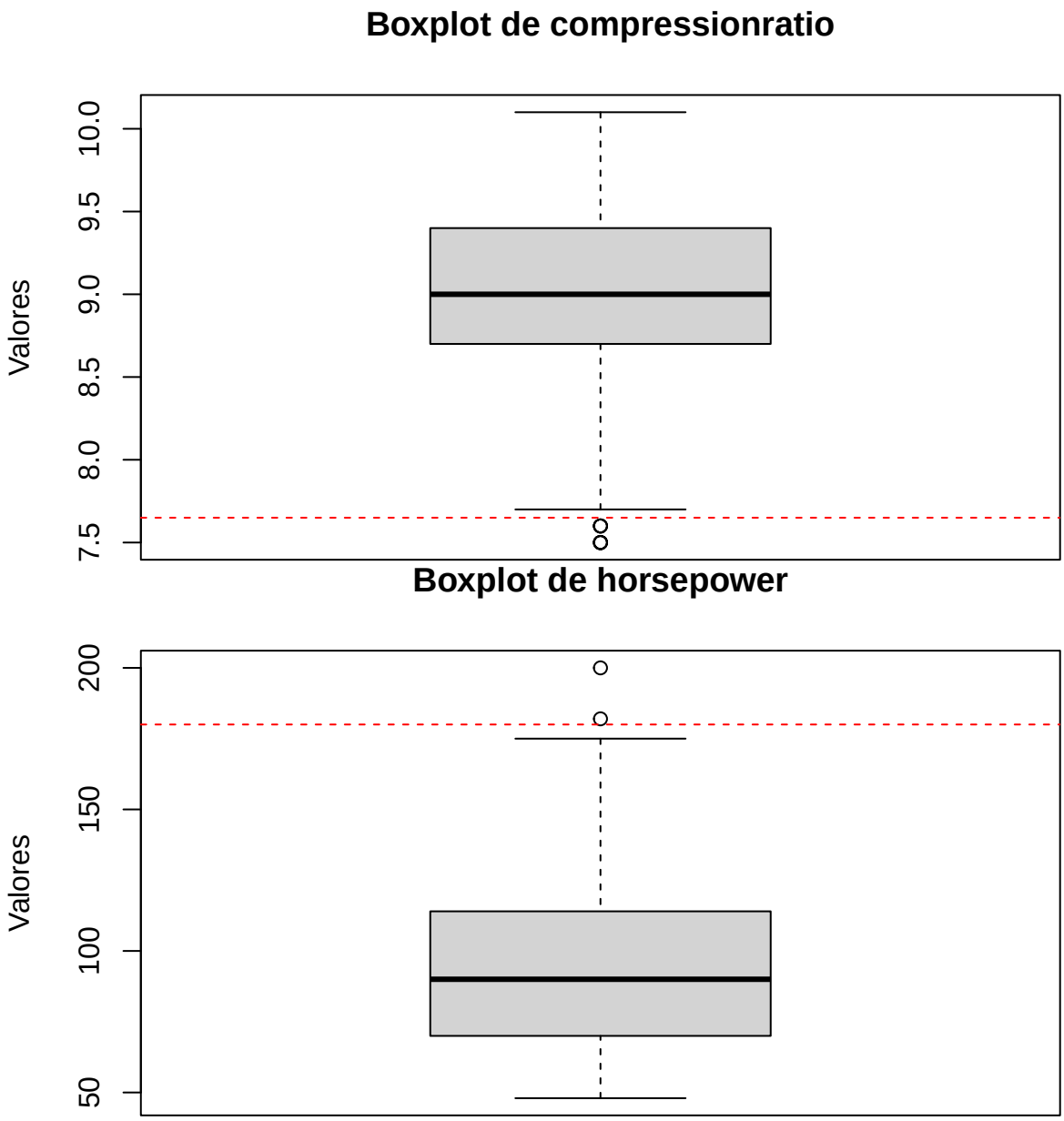


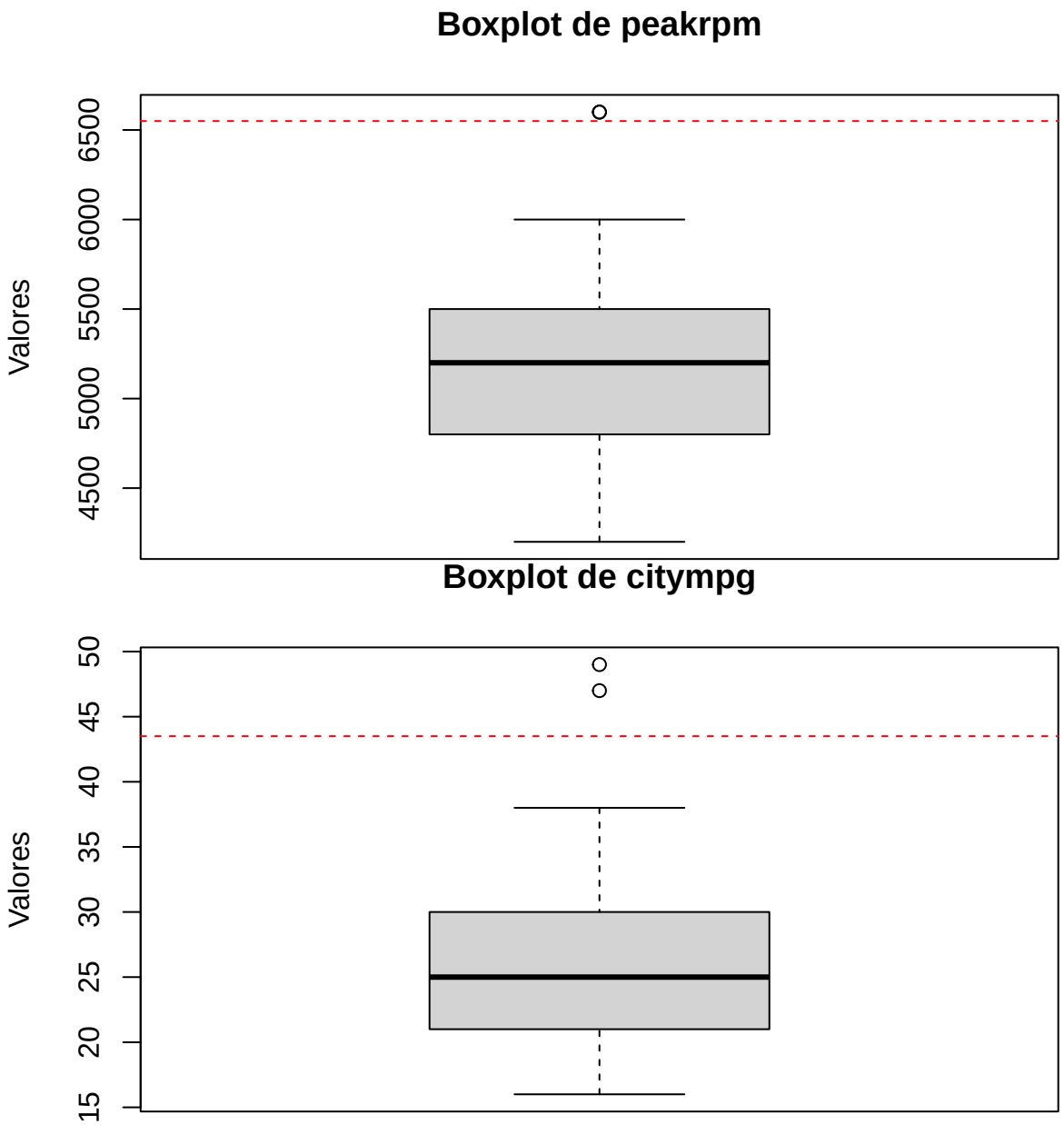

Boxplot de carlength
Valores
200
190
180
170
160
150
140
Boxplot de carwidth
Valores
70
68
66
64
62
60

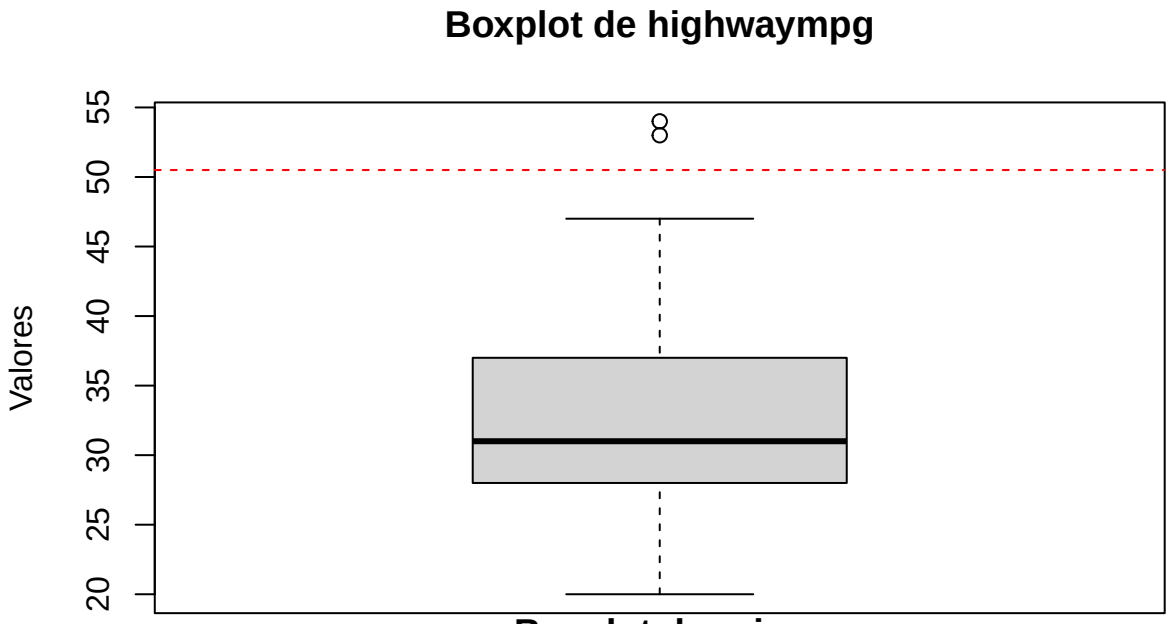

Boxplot de carheight
Valores
50
52
54
56
58
60

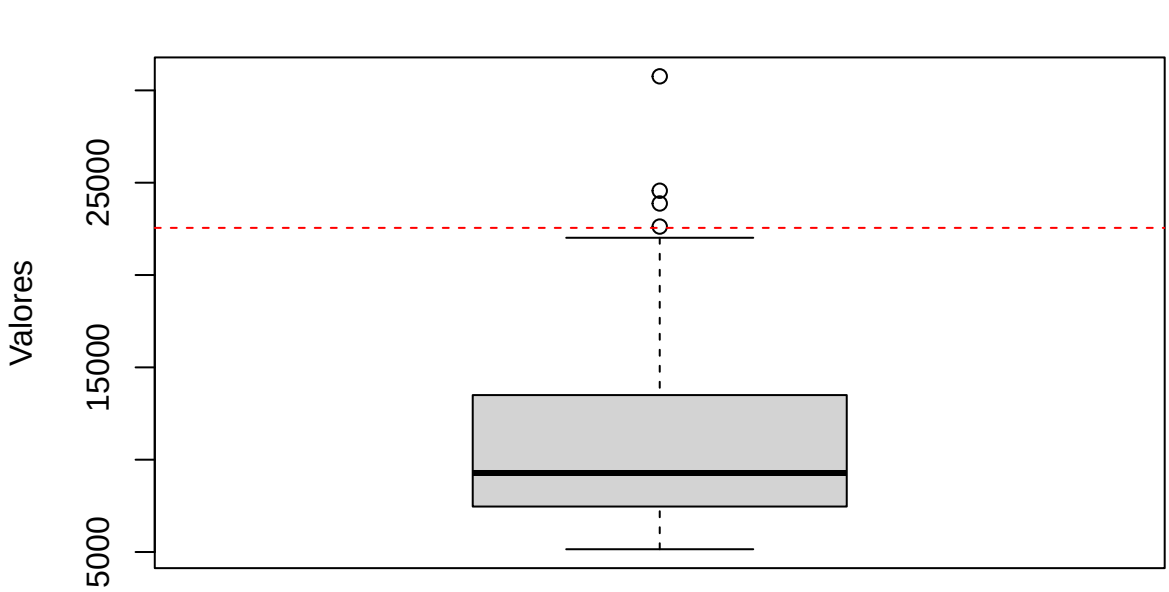

Boxplot de curbweight
Valores
1500
2000
2500
3000


Boxplot de enginesize
Valores
100
150
200
Boxplot de stroke
Valores
2.6
2.8
3.0
3.2
3.4
3.6
3.8


Boxplot de compressionratio
Valores
10.0
9.5
9.0
8.5
8.0
7.5
Boxplot de horsepower
Valores
200
150
100
50

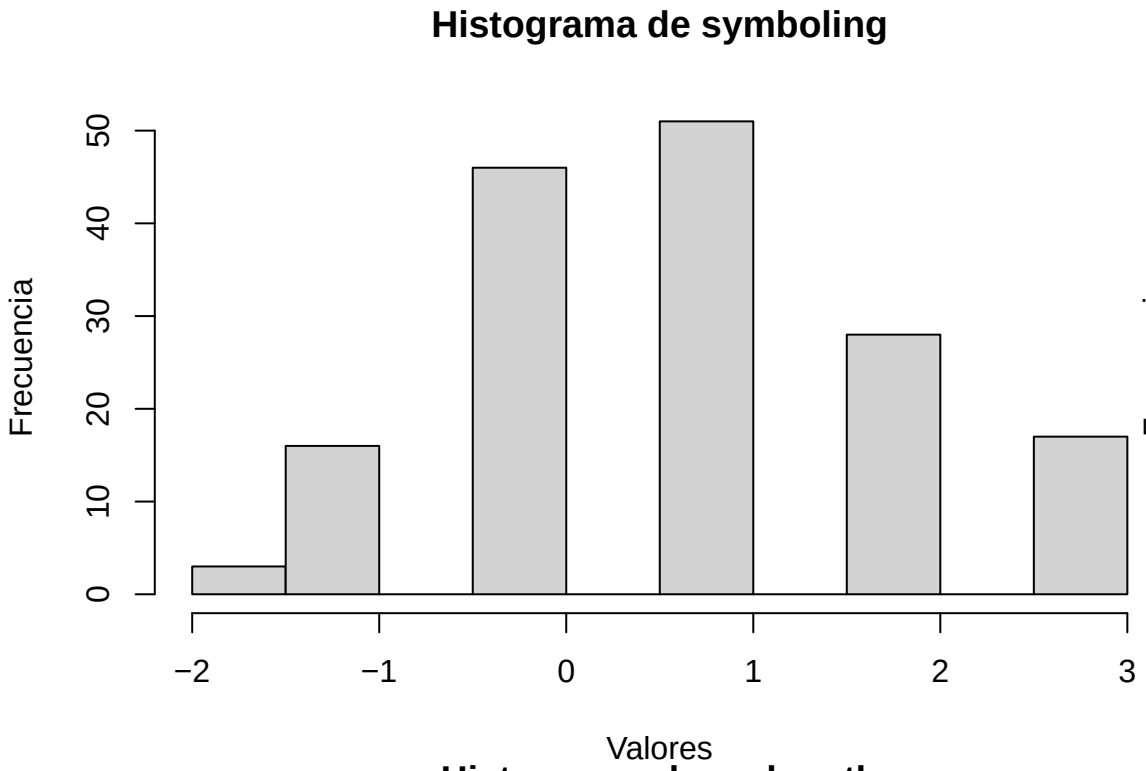

Boxplot de peakrpm
Valores
4500
5000
5500
6000
6500
Boxplot de citympg
Valores
15
20
25
30
35
40
45
50

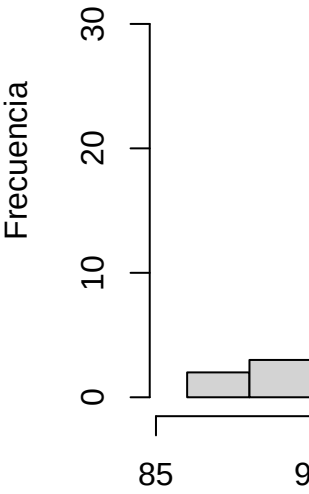


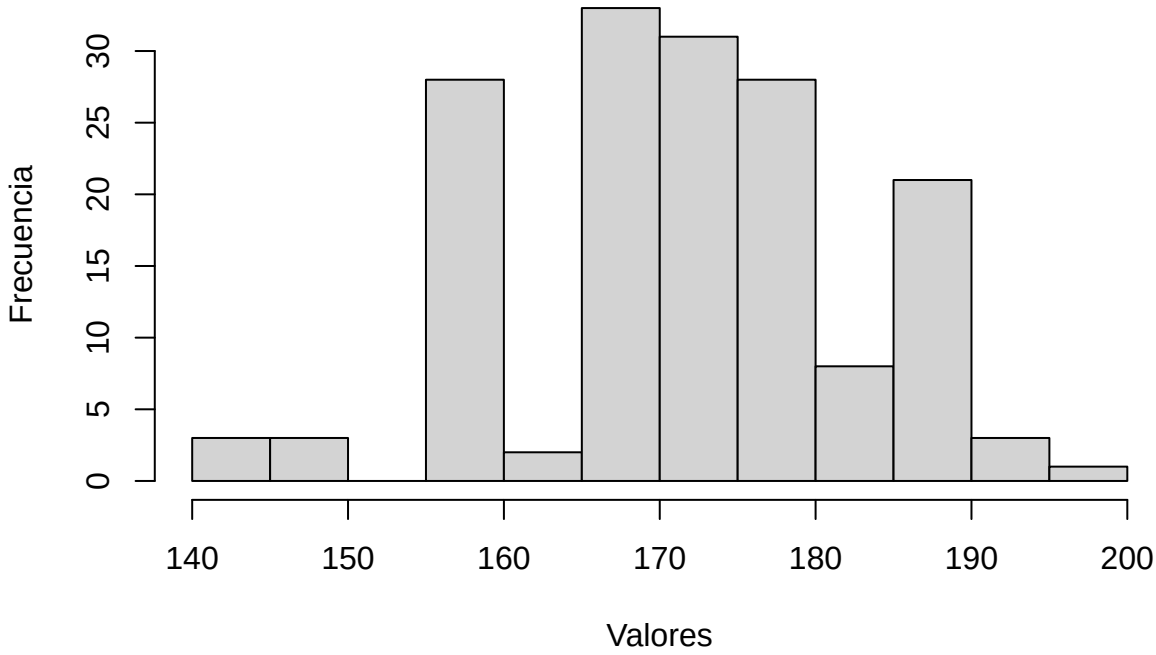


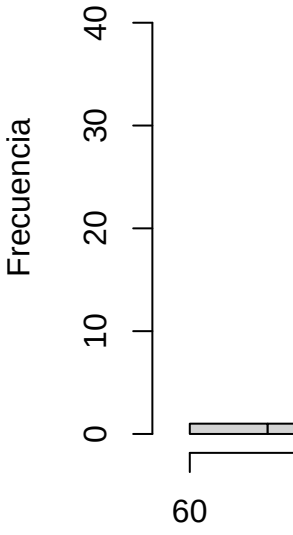
Histogramas

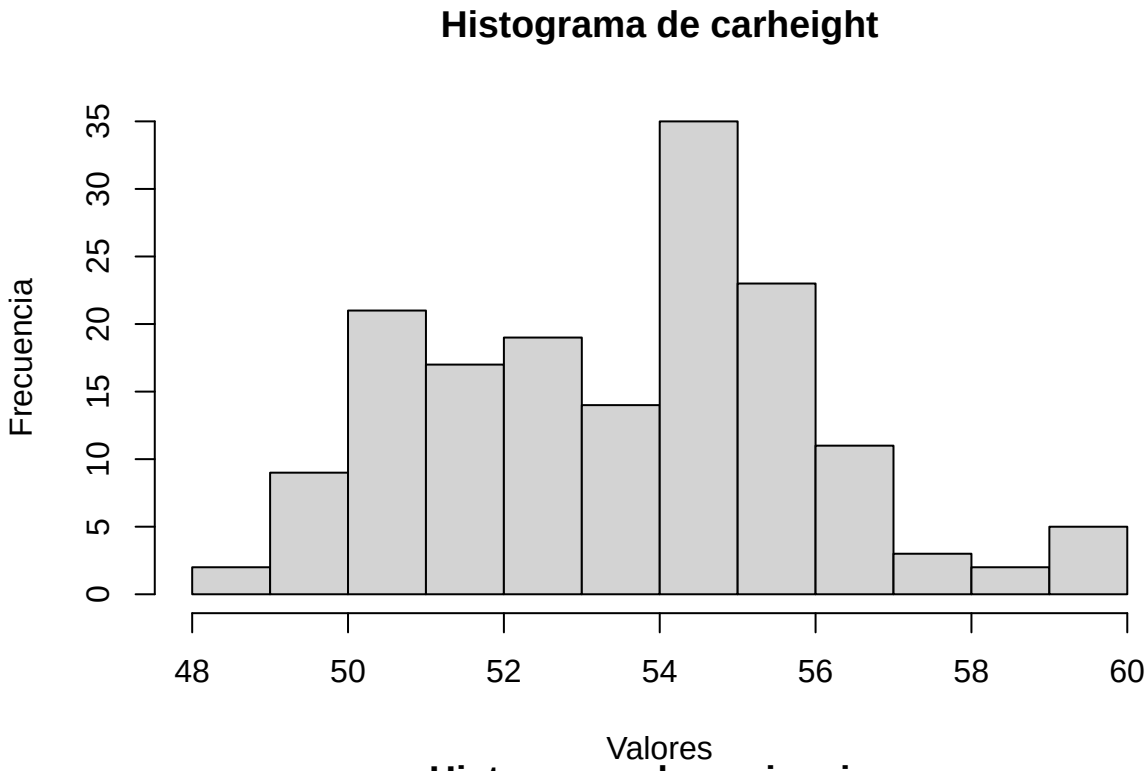
```
#Histogramas

create_histogram <- function(data, column_name) {
  hist(data[[column_name]], main=paste("Histograma de",column_name), xlab="Valores", ylab="Frecuencia",
}

for (i in names(numeric_columns[numeric_columns == TRUE])){
  create_histogram(df_clean, i)
}
```

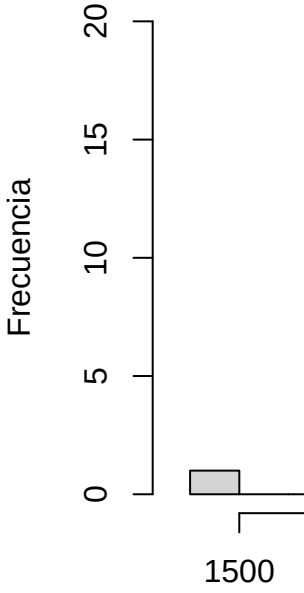

Histograma de symboling
Frecuencia
0
10
20
30
40
50
−2
−1
0
1
2
3
Valores


Frecuencia
0
10
20
30
85

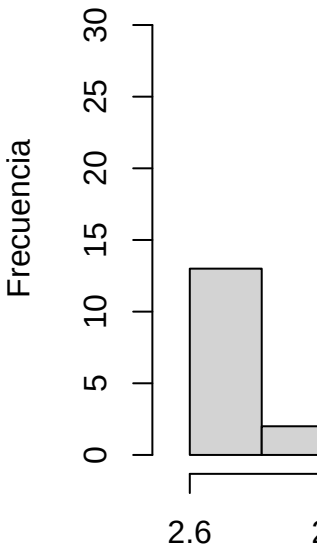

Histograma de carlength
Frecuencia
0
5
10
15
20
25
30
140
150
160
170
180
190
200
Valores

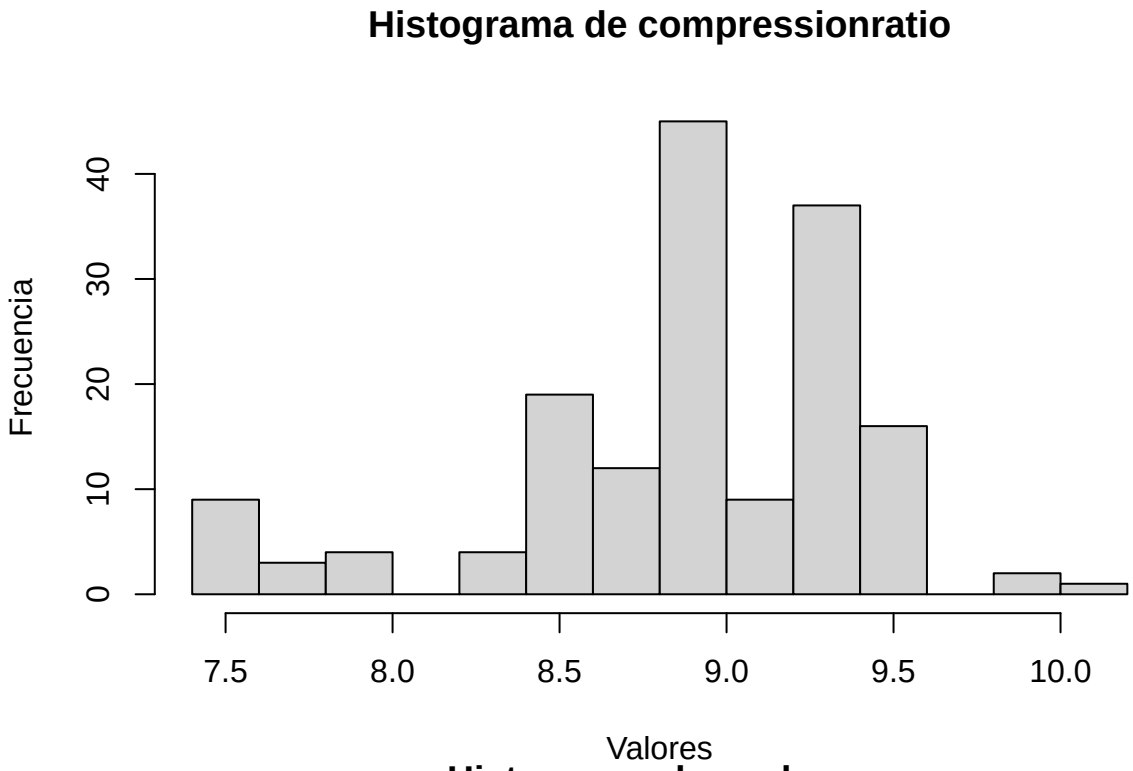

Frecuencia
0
10
20
30
40
60

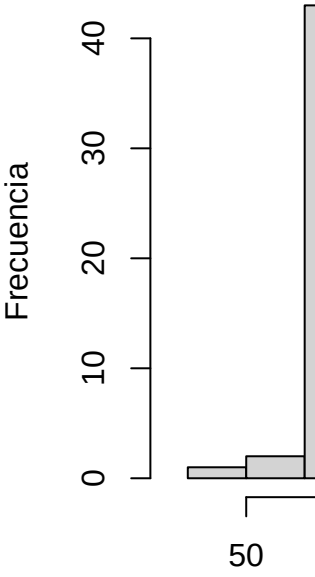

Histograma de carheight
Frecuencia
0
5
10
15
20
25
30
35
48
50
52
54
56
58
60
Valores

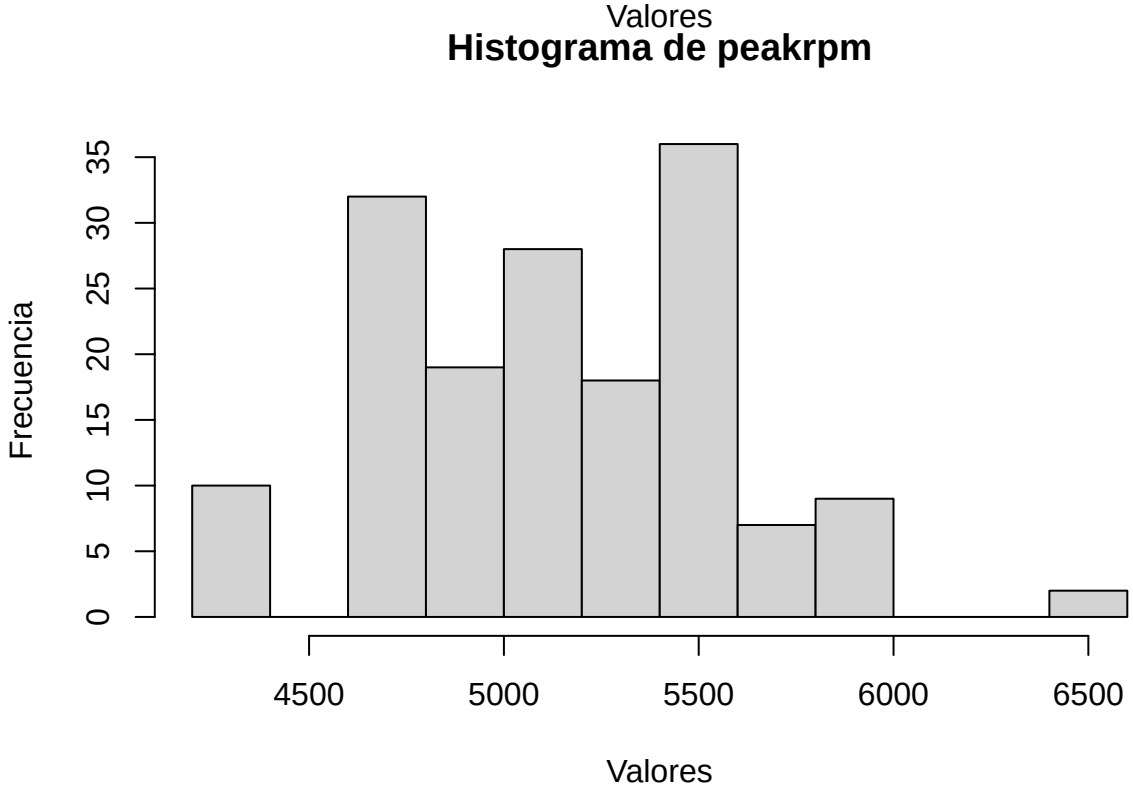

Frecuencia
0
5
10
15
20
1500

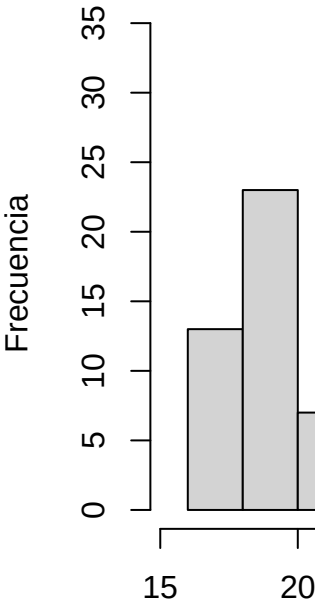

Histograma de enginesize
Frecuencia
0
10
20
30
40
100
150
200
Valores

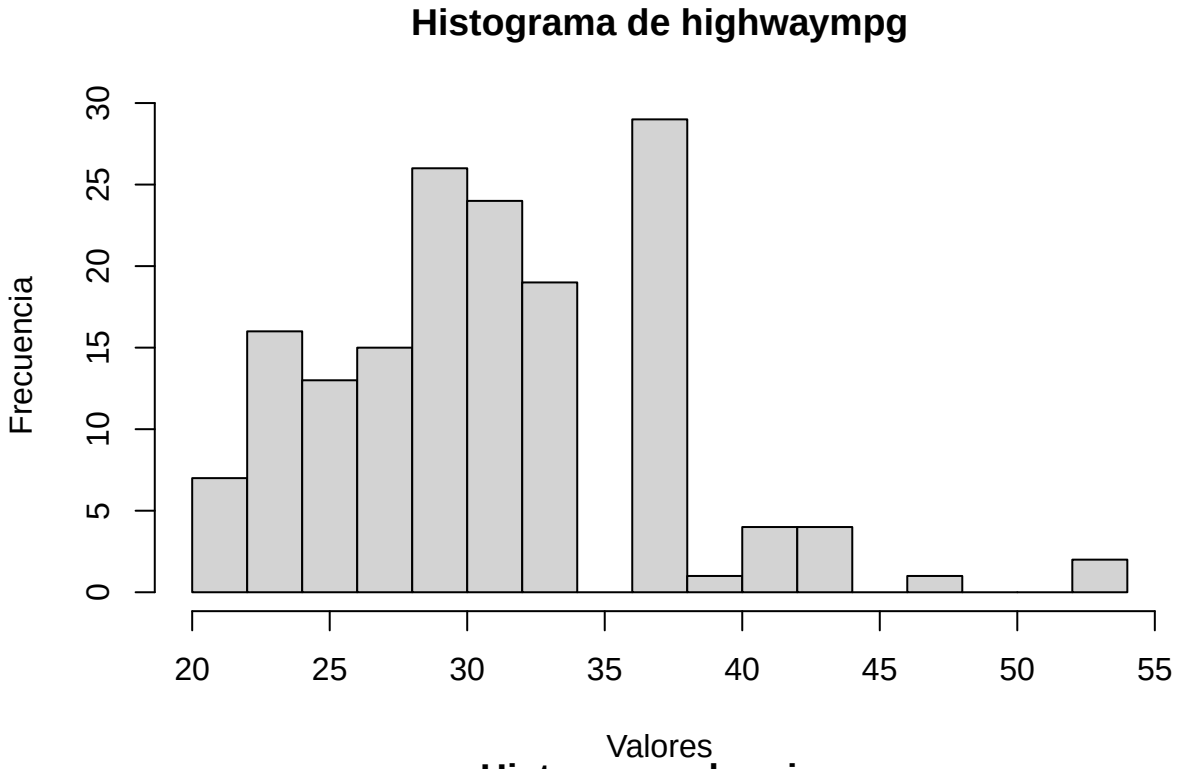

Frecuencia
0
5
10
15
20
25
30
2.6

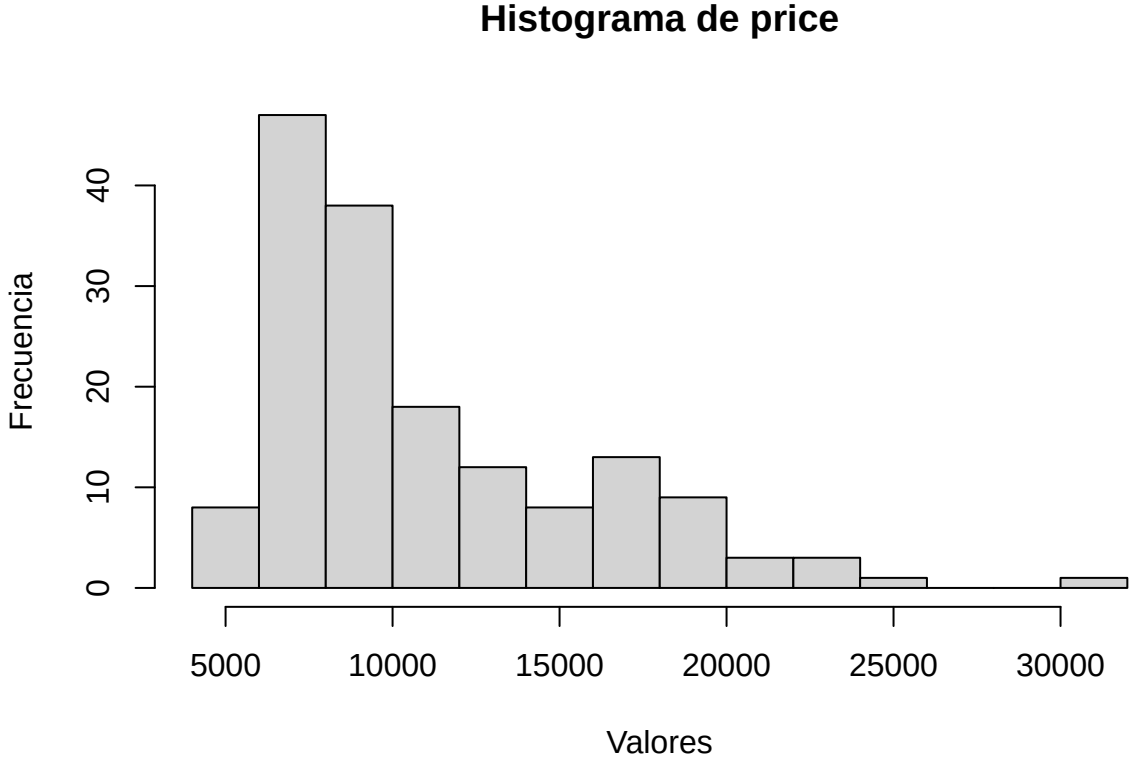

Histograma de compressionratio
Frecuencia
0
10
20
30
40
7.5
8.0
8.5
9.0
9.5
10.0
Valores


Frecuencia
0
10
20
30
40
50


Histograma de peakrpm
Frecuencia
0
5
10
15
20
25
30
35
4500
5000
5500
6000
6500
Valores


Frecuencia
0
5
10
15
20
25
30
35
15
20





En los histogramas se puede observar la distribución de las variables numéricas, en las variables *wheelbase*, *carwidth*, *curbweight*, *enginesize*, *horsepower*, *citympg* y *price* de puede notar una distribución asimétrica sesgada a la izquierda. Por otra parte, *symboling*, *carlength*, *stroke* y *peakrpm* se asemejan más a una distribución simética. Por último las variables *carheight*, *compressionratio* y *highwaympg* muestran una distribución más asimética sesgada a la derecha.

Distribución de variables categóricas

```
library(ggplot2)

plot_pie <- function(data, column_name) {
  # Calcular las frecuencias de cada categoría
  freq_table <- table(data[[column_name]])

  # Crear un gráfico de pay
  pie_chart <- ggplot(data = data.frame(freq_table),
                      aes(x = "", y = Freq, fill = factor(freq_table))) +
    geom_bar(stat = "identity", width = 1) +
    coord_polar(theta = "y") +
    labs(title = paste("Gráfico de Pay para", column_name)) +
    scale_fill_discrete(labels = names(freq_table)) +  # Usar nombres de categorías en la leyenda
    theme_void() +
    theme(legend.position = "right")

  return(pie_chart)
}

for (i in names(categoric_columns[categoric_columns == TRUE])[-1]){
  print(plot_pie(df_clean, i))
}
```

Gráfico de Pay para fueltype

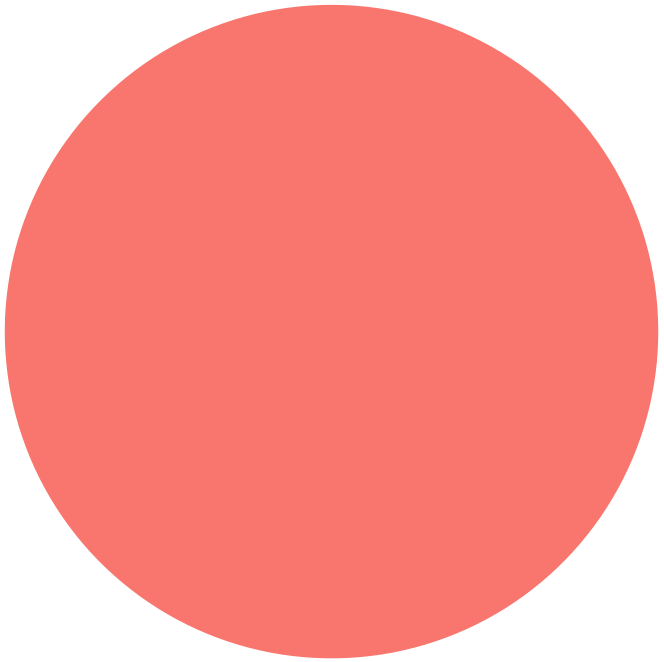
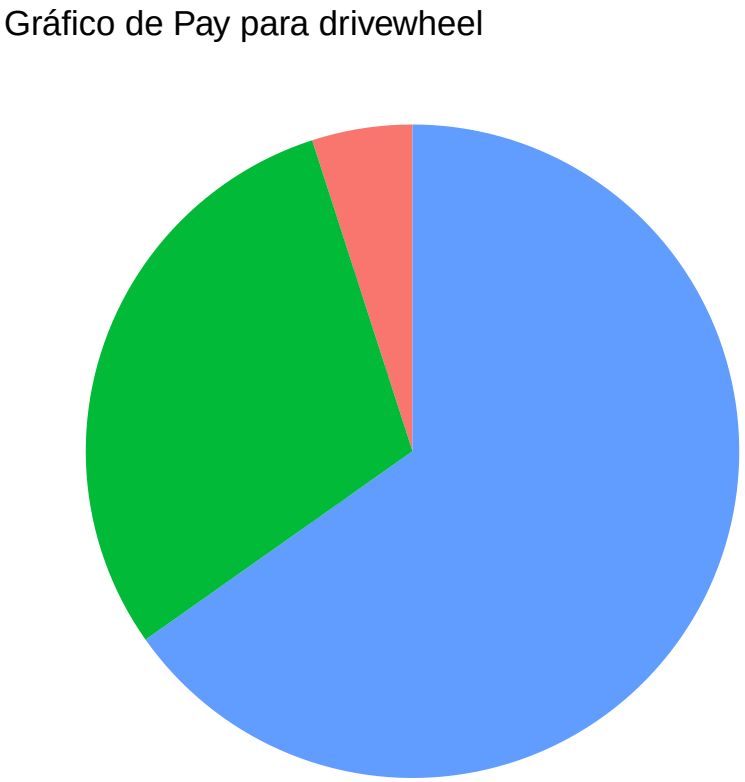
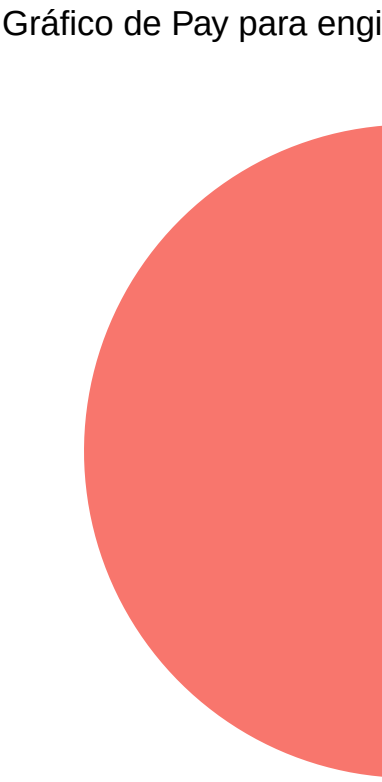
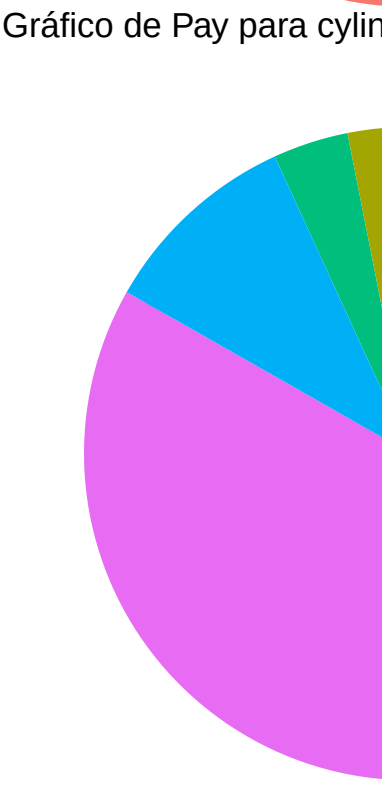


Gráfico de Pay para carb







Al analizar los gráficos de pay nos podemos percatar de varios insights, primeramente las variables *fueltype* y *enginelocation* perdieron valores posibles al momento de eliminar valores atípicos, significa que estos carros que en el anterior análisis se veía que usaban diesel o que tenian rear enginelocation eran carros atípicos en cuanto a variables numéricas. Por otra parte vemos que más del 75% de los carros tienen un carbody tipo sedan o hatchback. Más del 50% de los carros tienen un drivewheel (tipo de tracción) trasera (rwd). Más del

75% de los carros mantienen un tipo de motor ohcv. Por último, también más del 75% de los carros tienen 2 cilindros.

Seleccion de variables

Revisando valores faltantes

```
sum(is.na(df_clean))
```

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

Dados los bloxplots, histogramas, gráficos de pay y que no existen valores faltantes, se procede a la seleccion de variables importantes para el análisis de precio:

- price
- cylindernumber
- carbody
- wheelbase
- horsepower
- carlength
- enginesize

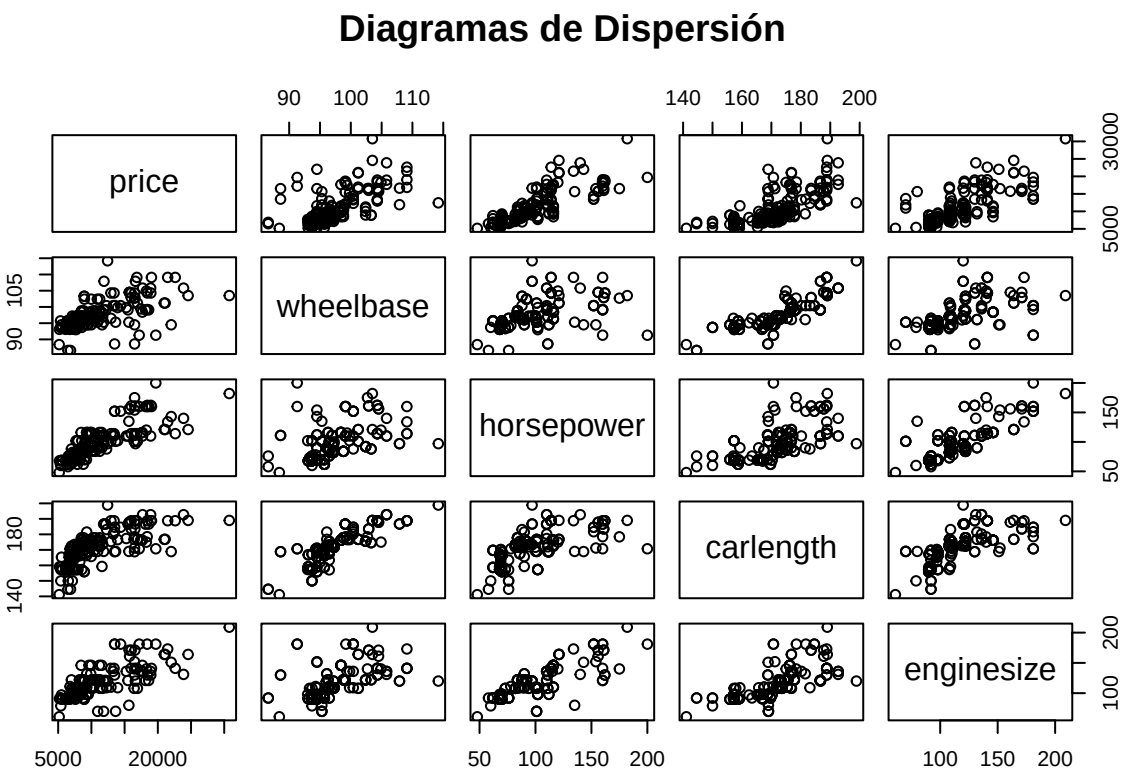
Se seleccionaron estas variables bajo la idea que son las más influyentes en la estética y potencia del carro, por lo que se tiene la hipótesis que tendrán buena relación con el precio.

```
# Análisis de colinealidad

cor_matrix <- cor(df_clean[c('price', 'wheelbase', 'horsepower', 'carlength', 'enginesize')])
print(cor_matrix)
```

```
##                price wheelbase horsepower carlength enginesize
## price      1.0000000 0.6204543  0.8179148 0.7120735  0.7453268
## wheelbase  0.6204543 1.0000000  0.4916286 0.8431678  0.5620357
## horsepower 0.8179148 0.4916286  1.0000000 0.6493830  0.8034007
## carlength  0.7120735 0.8431678  0.6493830 1.0000000  0.6926108
## enginesize 0.7453268 0.5620357  0.8034007 0.6926108  1.0000000
```

```
pairs(df_clean[c('price', 'wheelbase', 'horsepower', 'carlength', 'enginesize')], main="Diagramas de Di
```

	price	wheelbase	horsepower	carlength	enginesize
price	1.0000000	0.6204543	0.8179148 0.7120735	0.7120735	0.7453268



Si observamos las gáficas como los valores de la matriz de colinealidad, la variable precio parece tener una correlación muy alta con las demás variables numéricas seleccionadas, arriba de 0.7 a excepción de wheelbase.

Transformación de datos

```
df_seleccionado = df_clean[c('price', 'cylindernumber', 'carbody', 'wheelbase', 'horsepower', 'carlengt
head(df_seleccionado)
```

```
##   price cylindernumber     carbody wheelbase horsepower carlength enginesize
## 1 13495           four convertible      88.6        111     168.8        130
## 2 16500           four convertible      88.6        111     168.8        130
## 3 16500            six   hatchback      94.5        154     171.2        152
## 4 13950           four       sedan      99.8        102     176.6        109
## 5 17450           five       sedan      99.4        115     176.6        136
## 6 15250           five       sedan      99.8        110     177.3        136
```

Ahora que tenemos nuestra seleccion de variables en un df sin valores atípicos ni faltantes, realizaremos transformación de datos de acuerdo a lo necesario. El primer paso será trabajar con las variables categóricas:

- *cylindernumber*: Para esta variable nos podemos percatar que en realidad es una variable numérica escrita en texto, ya que en este caso los valores sí tienen relación de linealidad. Es decir, *four* cylinders sí es más que *two* cylinder. Por ello, se decide convertir la variable en variable numérica de acuerdo a un diccionario.

```
dictionary <- list(
  "two" = 2,
  "three" = 3,
```

```
  "four" = 4,
  "five" = 5,
  "six" = 6
)

map_values <- function(value, dictionary) {
  if (value %in% names(dictionary)) {
    return(dictionary[[value]])
  } else {
    return(value)
  }
}

df_seleccionado$cylindernumber <- sapply(df_seleccionado$cylindernumber, map_values, dictionary)
```

- *carbody*: Para esta variable se aplicará un one-hot encoding, construyendo una variable dummy para cada valor del tipo de carro.

```
convert_to_dummies <- function(data, column_name) {
  categorical_column <- data[[column_name]]
  unique_values <- unique(categorical_column)

  dummies <- sapply(unique_values, function(value) {
    dummy_name <- paste(column_name, value, sep = "_")
    as.numeric(categorical_column == value)
  })

  dummy_df <- as.data.frame(dummies)

  data <- cbind(data, dummy_df)

  data <- data[, -which(names(data) == column_name)]

  return(data)
}

df_seleccionado <- convert_to_dummies(df_seleccionado, "carbody")
```

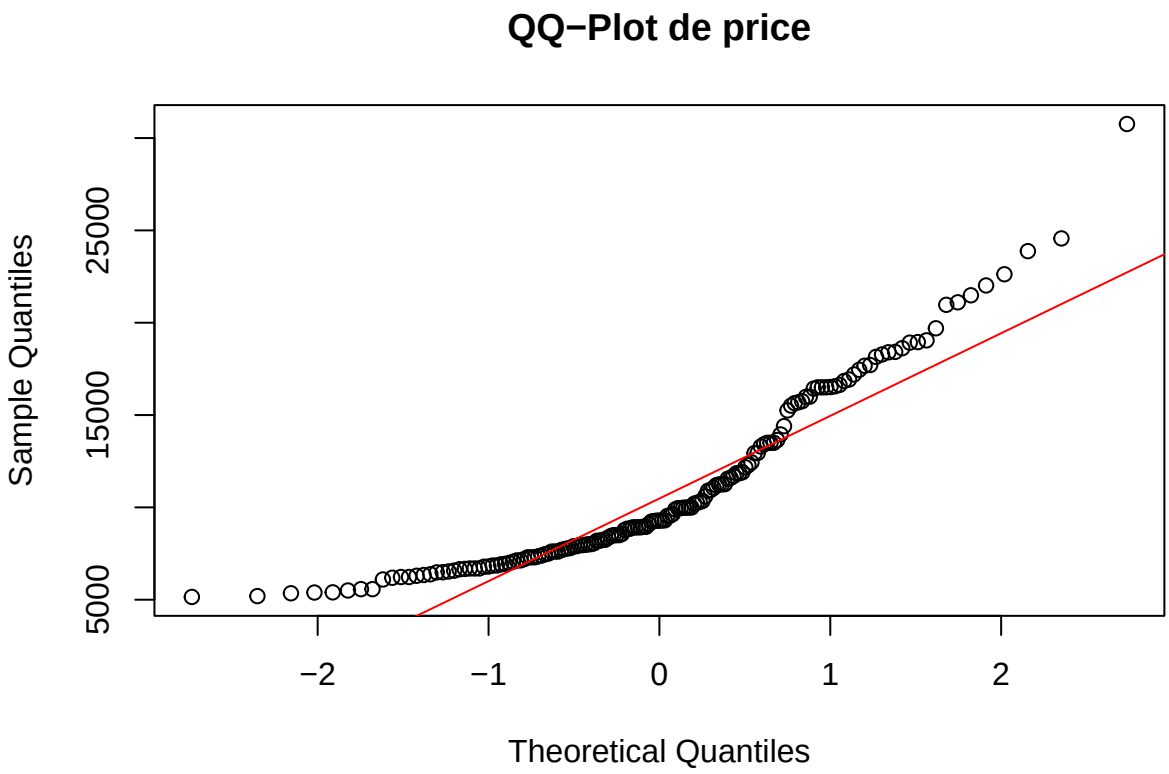
```
head(df_seleccionado)
```

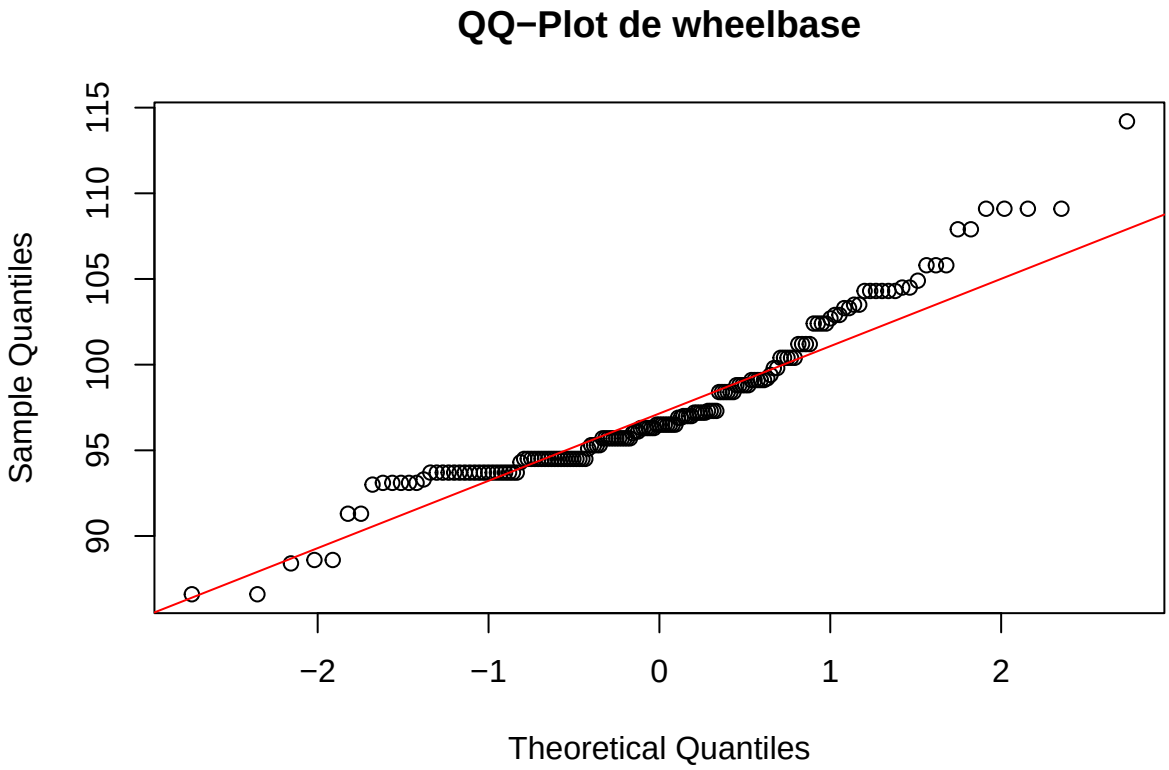
```
##   price cylindernumber wheelbase horsepower carlength enginesize convertible
## 1 13495              4      88.6        111     168.8        130           1
## 2 16500              4      88.6        111     168.8        130           1
## 3 16500              6      94.5        154     171.2        152           0
## 4 13950              4      99.8        102     176.6        109           0
## 5 17450              5      99.4        115     176.6        136           0
## 6 15250              5      99.8        110     177.3        136           0
##   hatchback sedan wagon hardtop
## 1         0     0     0       0
## 2         0     0     0       0
## 3         1     0     0       0
## 4         0     1     0       0
## 5         0     1     0       0
## 6         0     1     0       0
```

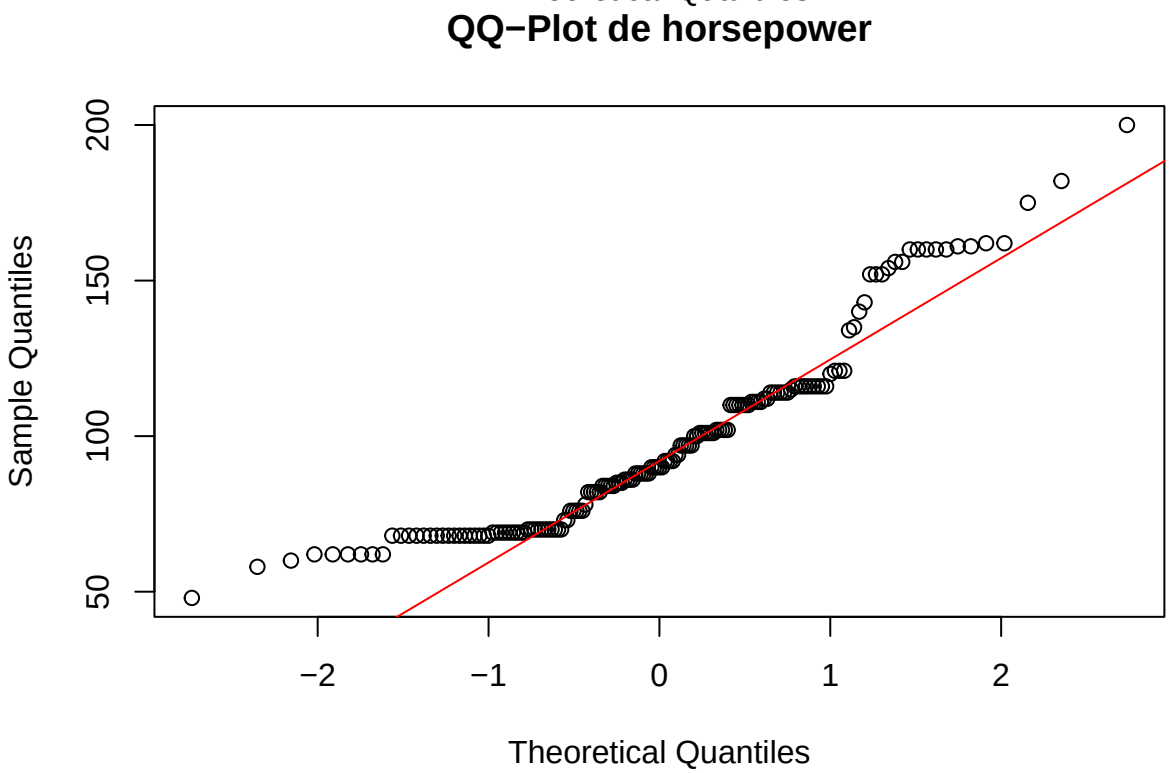
Análisis de normalidad

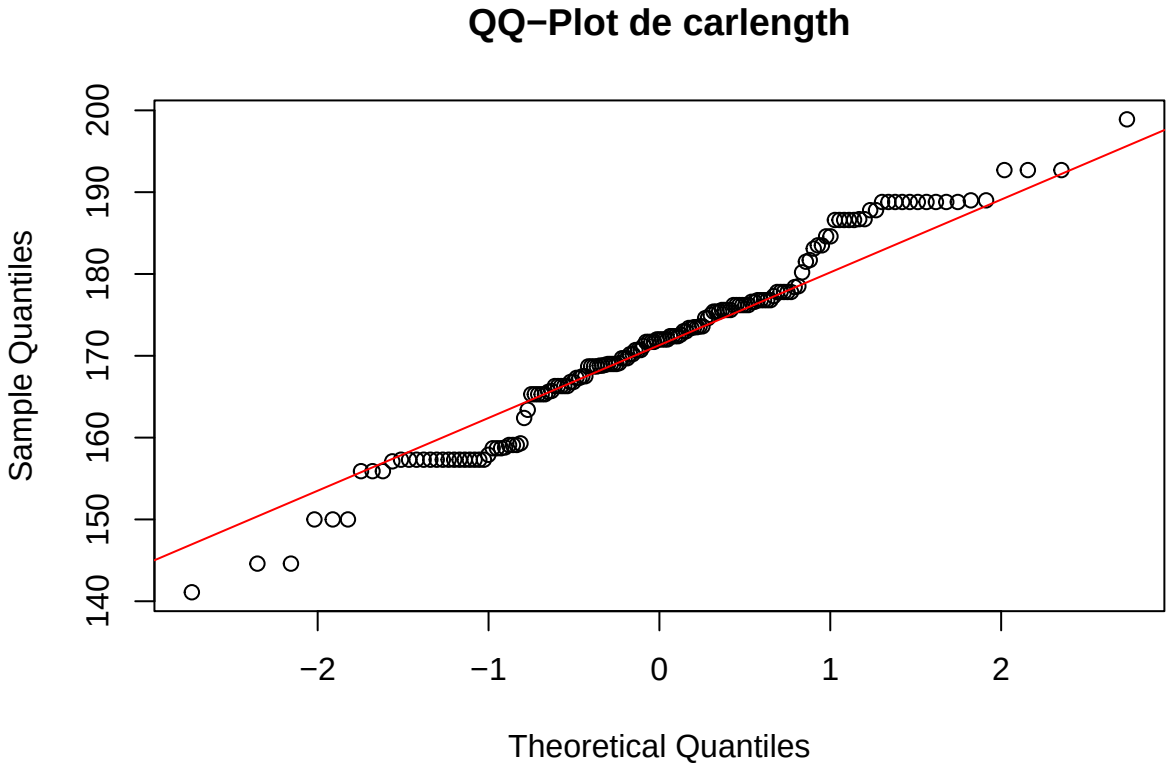
El primer análisis de normalidad en las variables numéricas se realizará mediante QQ-plots.

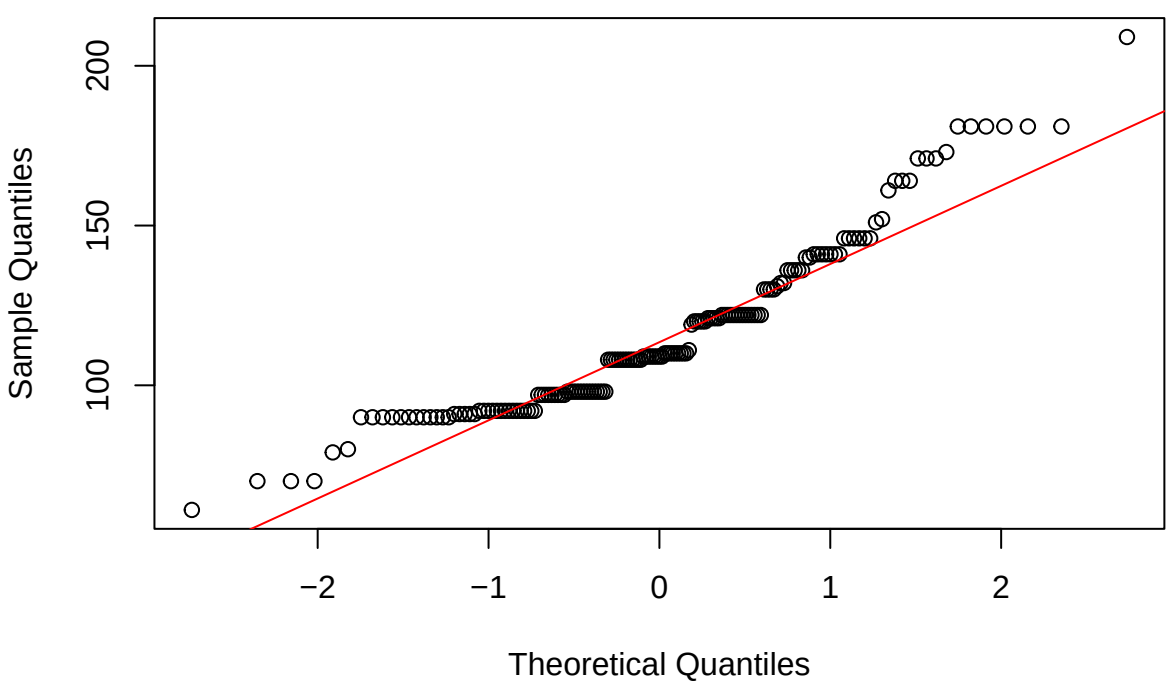
```
plot_qqplot <- function(data, column_name) {
  column_data <- data[[column_name]]
  qqnorm(column_data, main = paste("QQ-Plot de", column_name))
  qqline(column_data, col = "red")
}


for (i in c('price', 'wheelbase', 'horsepower', 'carlength', 'enginesize')){
  plot_qqplot(df_seleccionado, i)
}
```




QQ−Plot de wheelbase
Sample Quantiles
90
95
100
105
110
115
−2
−1
0
1
2
Theoretical Quantiles


QQ−Plot de horsepower
Sample Quantiles
50
100
150
200
−2
−1
0
1
2
Theoretical Quantiles





Las gráficas QQplot muestran que los datos parecen ajustarse a la normalidad, sin embargo, necesitamos confirmarlo estadísticamente. Para ello haremos una prueba de .

```
library(nortest)

# Realizar la prueba de normalidad de Anderson-Darling para 'price'
```

```r
ad_test_price <- ad.test(df_seleccionado$price)
print("Anderson-Darling test for 'price':")
```

```
## [1] "Anderson-Darling test for 'price':"
```

```r
print(ad_test_price)
```

```
##
##  Anderson-Darling normality test
##
## data:  df_seleccionado$price
## A = 6.3209, p-value = 1.385e-15
```

```r
# Realizar la prueba de normalidad de Anderson-Darling para 'wheelbase'
ad_test_wheelbase <- ad.test(df_seleccionado$wheelbase)
print("Anderson-Darling test for 'wheelbase':")
```

```
## [1] "Anderson-Darling test for 'wheelbase':"
```

```r
print(ad_test_wheelbase)
```

```
##
##  Anderson-Darling normality test
##
## data:  df_seleccionado$wheelbase
## A = 4.1518, p-value = 2.289e-10
```

```r
# Realizar la prueba de normalidad de Anderson-Darling para 'horsepower'
ad_test_horsepower <- ad.test(df_seleccionado$horsepower)
print("Anderson-Darling test for 'horsepower':")
```

```
## [1] "Anderson-Darling test for 'horsepower':"
```

```r
print(ad_test_horsepower)
```

```
##
##  Anderson-Darling normality test
##
## data:  df_seleccionado$horsepower
## A = 4.8539, p-value = 4.595e-12
```

```r
# Realizar la prueba de normalidad de Anderson-Darling para 'carlength'
ad_test_carlength <- ad.test(df_seleccionado$carlength)
print("Anderson-Darling test for 'carlength':")
```

```
## [1] "Anderson-Darling test for 'carlength':"
```

```r
print(ad_test_carlength)
```

```
##
##  Anderson-Darling normality test
##
## data:  df_seleccionado$carlength
## A = 1.2891, p-value = 0.002313
```

```r
# Realizar la prueba de normalidad de Anderson-Darling para 'enginesize'
ad_test_enginesize <- ad.test(df_seleccionado$enginesize)
print("Anderson-Darling test for 'enginesize':")
```

```
## [1] "Anderson-Darling test for 'enginesize':"
```

```r
print(ad_test_enginesize)
```

```
##
##  Anderson-Darling normality test
##
## data:  df_seleccionado$enginesize
## A = 4.2065, p-value = 1.687e-10
```

Los resultados de la prueba de normalidad de Anderson-Darling para las diferentes variables son los siguientes:

1. **'price'**:
   - Estadístico de Anderson-Darling (A) = 6.3209
   - Valor p (p-value) = 1.385e-15 (muy cercano a cero) El valor p extremadamente bajo (cercano a cero) sugiere que los datos en la variable 'price' no siguen una distribución normal. Esto significa que la variable 'price' no se ajusta bien a una distribución normal y es estadísticamente significativamente diferente de una distribución normal.
2. **'wheelbase'**:
   - Estadístico de Anderson-Darling (A) = 4.1518
   - Valor p (p-value) = 2.289e-10 (muy cercano a cero) El valor p extremadamente bajo (cercano a cero) sugiere que los datos en la variable 'wheelbase' no siguen una distribución normal. Al igual que en el caso anterior, esto indica que la variable 'wheelbase' no se ajusta bien a una distribución normal y es estadísticamente significativamente diferente de una distribución normal.
3. **'horsepower'**:
   - Estadístico de Anderson-Darling (A) = 4.8539
   - Valor p (p-value) = 4.595e-12 (muy cercano a cero) El valor p extremadamente bajo (cercano a cero) sugiere que los datos en la variable 'horsepower' no siguen una distribución normal. Una vez más, esto indica que la variable 'horsepower' no se ajusta bien a una distribución normal y es estadísticamente significativamente diferente de una distribución normal.
4. **'carlength'**:
   - Estadístico de Anderson-Darling (A) = 1.2891
   - Valor p (p-value) = 0.002313 (pequeño, pero no extremadamente bajo) En este caso, el valor p es pequeño pero no extremadamente bajo. Esto sugiere que los datos en la variable 'carlength' no siguen una distribución normal, pero la desviación de la normalidad podría ser menos pronunciada en comparación con las otras variables.
5. **'enginesize'**:
   - Estadístico de Anderson-Darling (A) = 4.2065
   - Valor p (p-value) = 1.687e-10 (muy cercano a cero) Al igual que en los casos anteriores, el valor p extremadamente bajo (cercano a cero) sugiere que los datos en la variable 'enginesize' no siguen una distribución normal y son estadísticamente significativamente diferentes de una distribución normal.

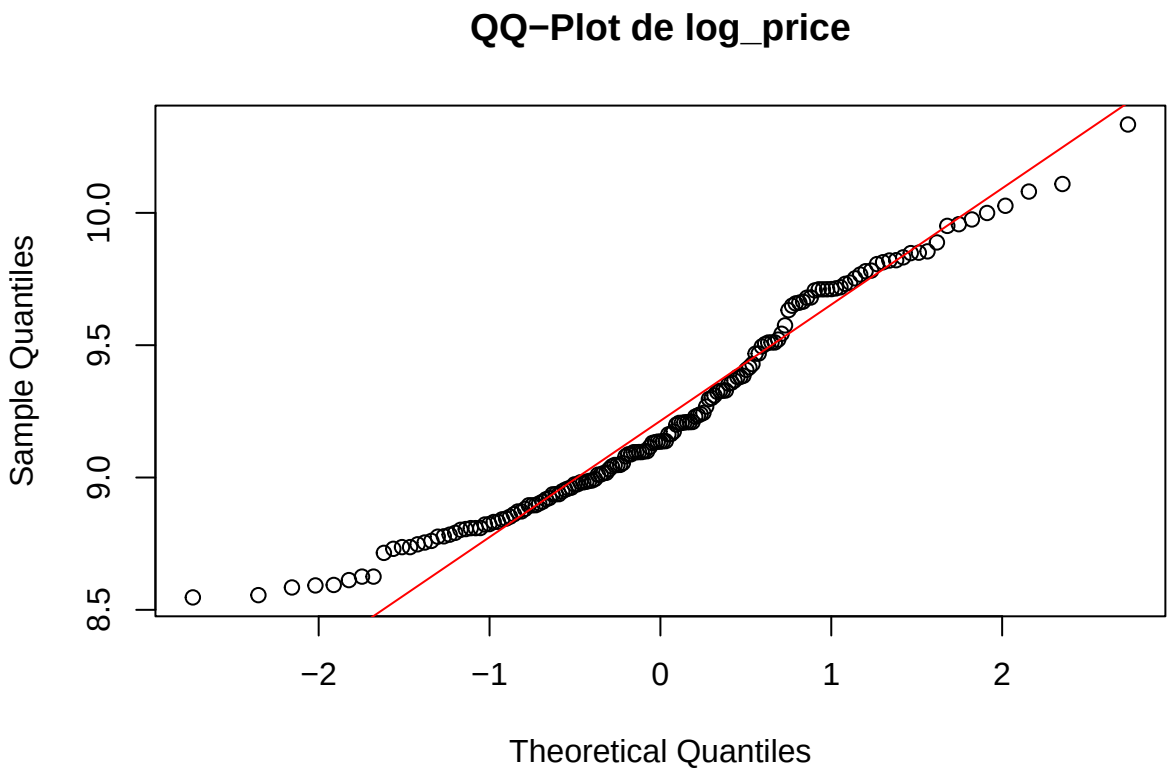
En resumen, los resultados de las pruebas de Anderson-Darling indican que las variables 'price', 'wheelbase', 'horsepower' y 'enginesize' no siguen una distribución normal. La variable 'carlength' también muestra desviaciones de la normalidad, aunque menos pronunciadas en comparación con las otras variables. Esto tiene implicaciones importantes para el análisis estadístico, ya que muchos métodos paramétricos asumen normalidad en los datos. Puede ser necesario considerar enfoques no paramétricos o transformaciones de datos para abordar estas desviaciones de la normalidad, dependiendo de los objetivos de tu análisis.

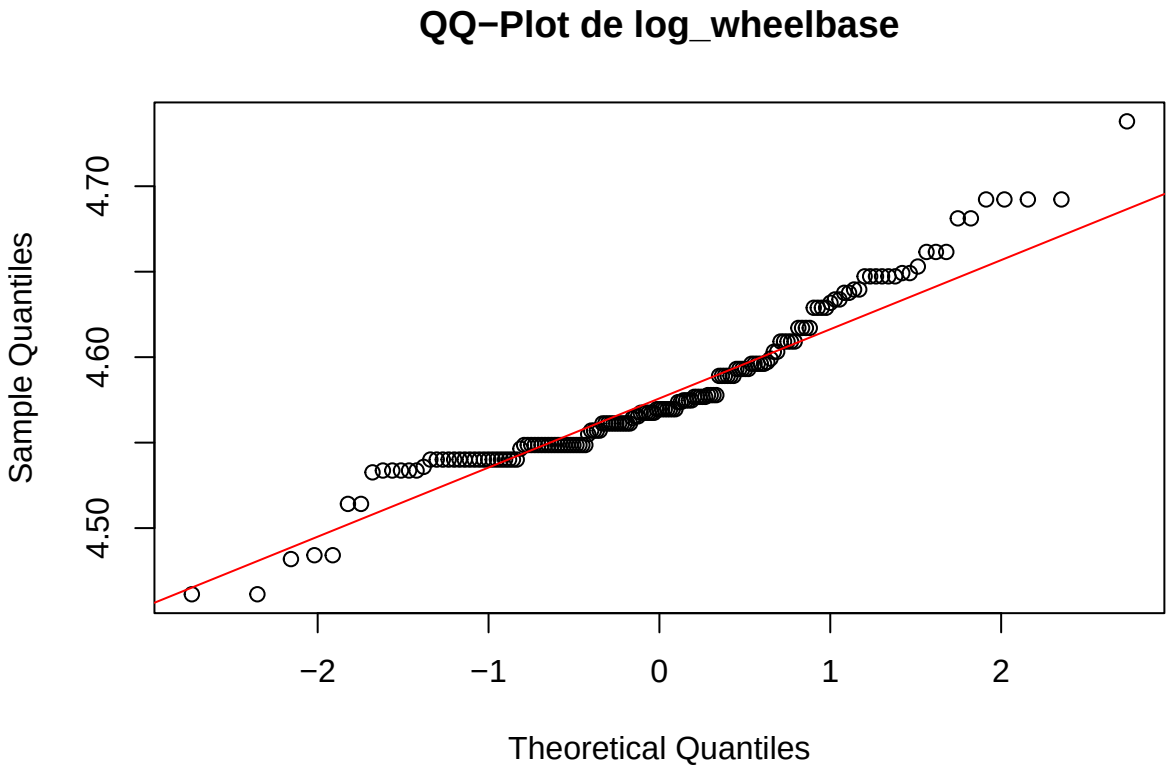
Normalización

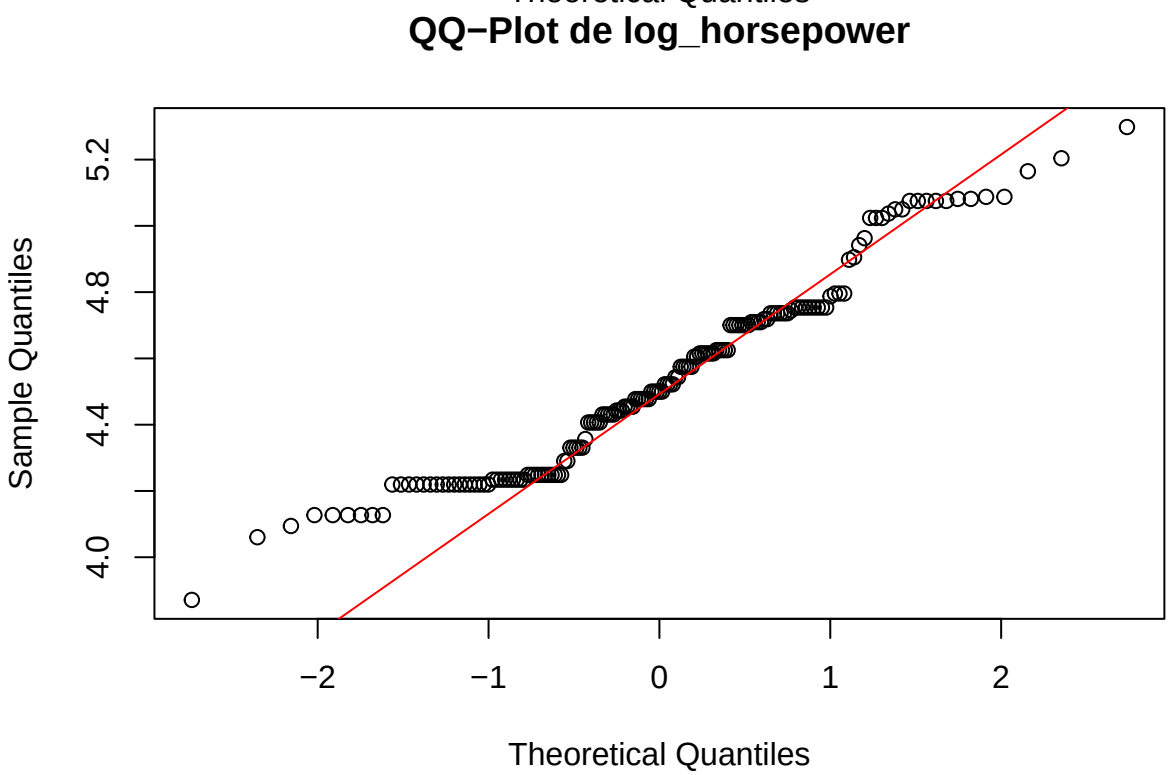
Se intentará normalizar las variables mediante transformaciones. El primer intento de normalización será aplicar una transformación logaritmica.

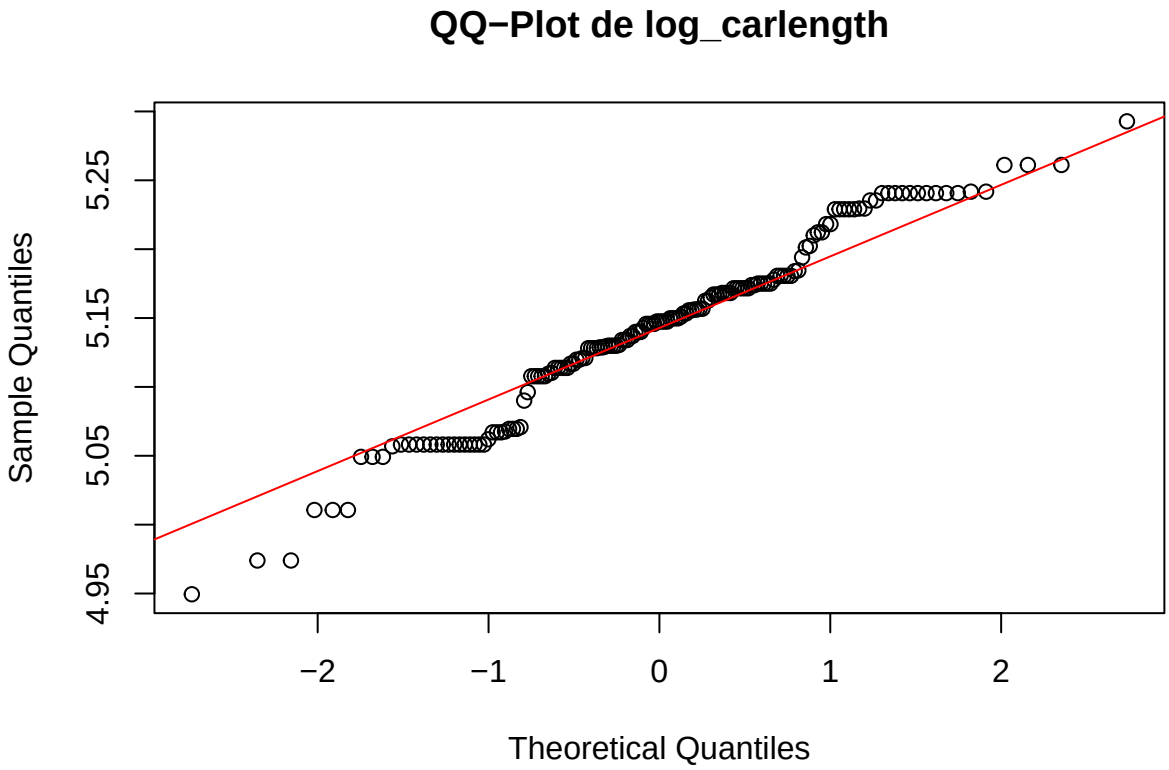
```r
# Aplicar la transformación logarítmica a las variables
df_seleccionado$log_price <- log(df_seleccionado$price)
df_seleccionado$log_wheelbase <- log(df_seleccionado$wheelbase)
df_seleccionado$log_horsepower <- log(df_seleccionado$horsepower)
```

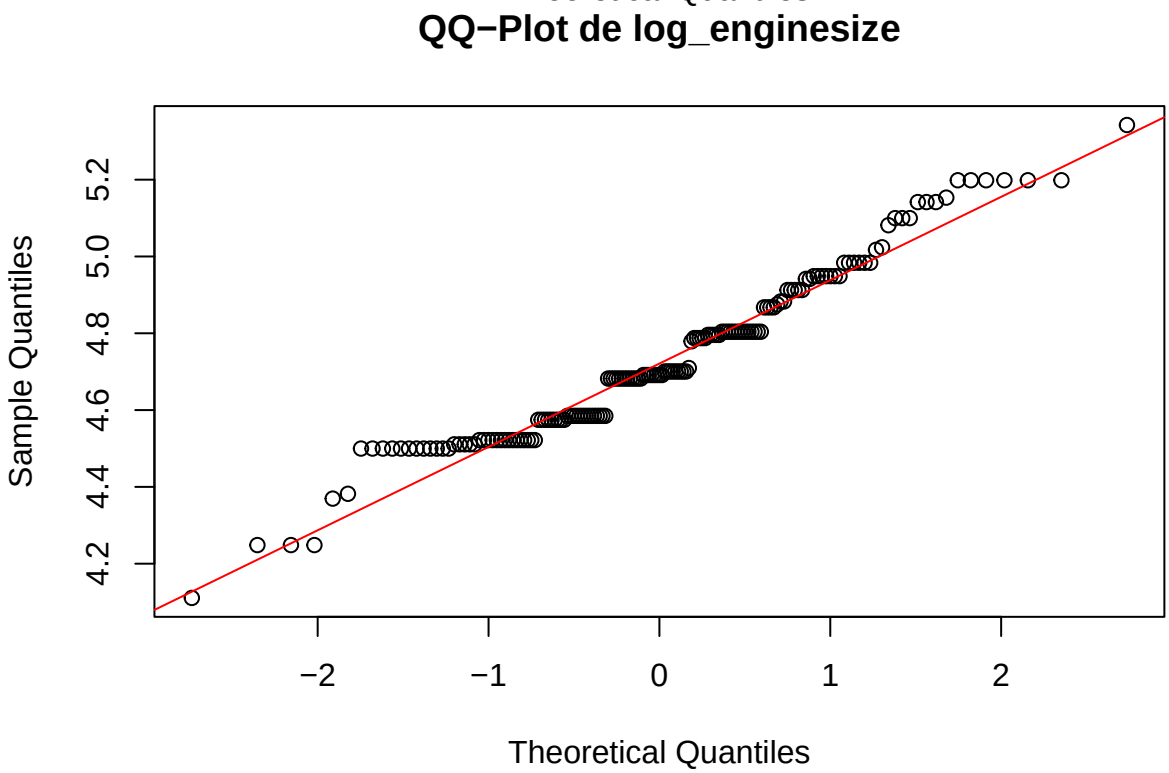
```r
df_seleccionado$log_carlength <- log(df_seleccionado$carlength)
df_seleccionado$log_enginesize <- log(df_seleccionado$enginesize)

# Verificar la normalidad de las variables transformadas con QQ-plots
library(ggplot2)

plot_qqplot <- function(data, column_name) {
  column_data <- data[[column_name]]
  qqnorm(column_data, main = paste("QQ-Plot de", column_name))
  qqline(column_data, col = "red")
}

for (i in c('log_price', 'log_wheelbase', 'log_horsepower', 'log_carlength', 'log_enginesize')){
  plot_qqplot(df_seleccionado, i)
}
```




QQ−Plot de log_wheelbase
Sample Quantiles
4.50
4.60
4.70
−2
−1
0
1
2
Theoretical Quantiles


QQ−Plot de log_horsepower
Sample Quantiles
4.0
4.4
4.8
5.2
−2
−1
0
1
2
Theoretical Quantiles





Anderson Darling test para variables transformadas

```
ad_test_price <- ad.test(df_seleccionado$log_price)
ad_test_wheelbase <- ad.test(df_seleccionado$log_wheelbase)
ad_test_horsepower <- ad.test(df_seleccionado$log_horsepower)
ad_test_carlength <- ad.test(df_seleccionado$log_carlength)
```

```r
ad_test_enginesize <- ad.test(df_seleccionado$log_enginesize)

# Mostrar los resultados de la prueba de Anderson-Darling
print("Anderson-Darling test for 'log_price':")
```

```
## [1] "Anderson-Darling test for 'log_price':"
```

```r
print(ad_test_price)
```

```
##
##  Anderson-Darling normality test
##
## data:  df_seleccionado$log_price
## A = 2.3788, p-value = 4.81e-06
```

```r
print("Anderson-Darling test for 'log_wheelbase':")
```

```
## [1] "Anderson-Darling test for 'log_wheelbase':"
```

```r
print(ad_test_wheelbase)
```

```
##
##  Anderson-Darling normality test
##
## data:  df_seleccionado$log_wheelbase
## A = 3.6575, p-value = 3.626e-09
```

```r
print("Anderson-Darling test for 'log_horsepower':")
```

```
## [1] "Anderson-Darling test for 'log_horsepower':"
```

```r
print(ad_test_horsepower)
```

```
##
##  Anderson-Darling normality test
##
## data:  df_seleccionado$log_horsepower
## A = 2.446, p-value = 3.291e-06
```

```r
print("Anderson-Darling test for 'log_carlength':")
```

```
## [1] "Anderson-Darling test for 'log_carlength':"
```

```r
print(ad_test_carlength)
```

```
##
##  Anderson-Darling normality test
##
## data:  df_seleccionado$log_carlength
## A = 1.377, p-value = 0.001403
```

```r
print("Anderson-Darling test for 'log_enginesize':")
```

```
## [1] "Anderson-Darling test for 'log_enginesize':"
```

```r
print(ad_test_enginesize)
```

```
##
##  Anderson-Darling normality test
##
```

```
## data:  df_seleccionado$log_enginesize
## A = 2.3015, p-value = 7.443e-06
```

-Los valores p son muy pequeños en todas las variables, lo que indica que los datos transformados ('log_price', 'log_wheelbase', 'log_horsepower', 'log_carlength', 'log_enginesize') no siguen una distribución normal.

- Dado que los valores p son significativamente menores que un nivel de significancia típico de 0.05, podemos rechazar la hipótesis nula de normalidad en todos los casos.
- Esto significa que las variables transformadas no son normalmente distribuidas. Es importante tener en cuenta que la normalidad no es un requisito estricto para todas las técnicas de análisis estadístico. Dependerá de tus objetivos analíticos específicos si esta falta de normalidad es un problema o no. Puedes considerar métodos estadísticos que no requieran normalidad o explorar otras transformaciones de datos si es necesario.

Intentando normalizar con yeo-johnson

```
# Eliminar las variables transformadas logarítmicamente del DataFrame
df_seleccionado <- df_seleccionado[, !names(df_seleccionado) %in% c("log_price", "log_wheelbase", "log_
```

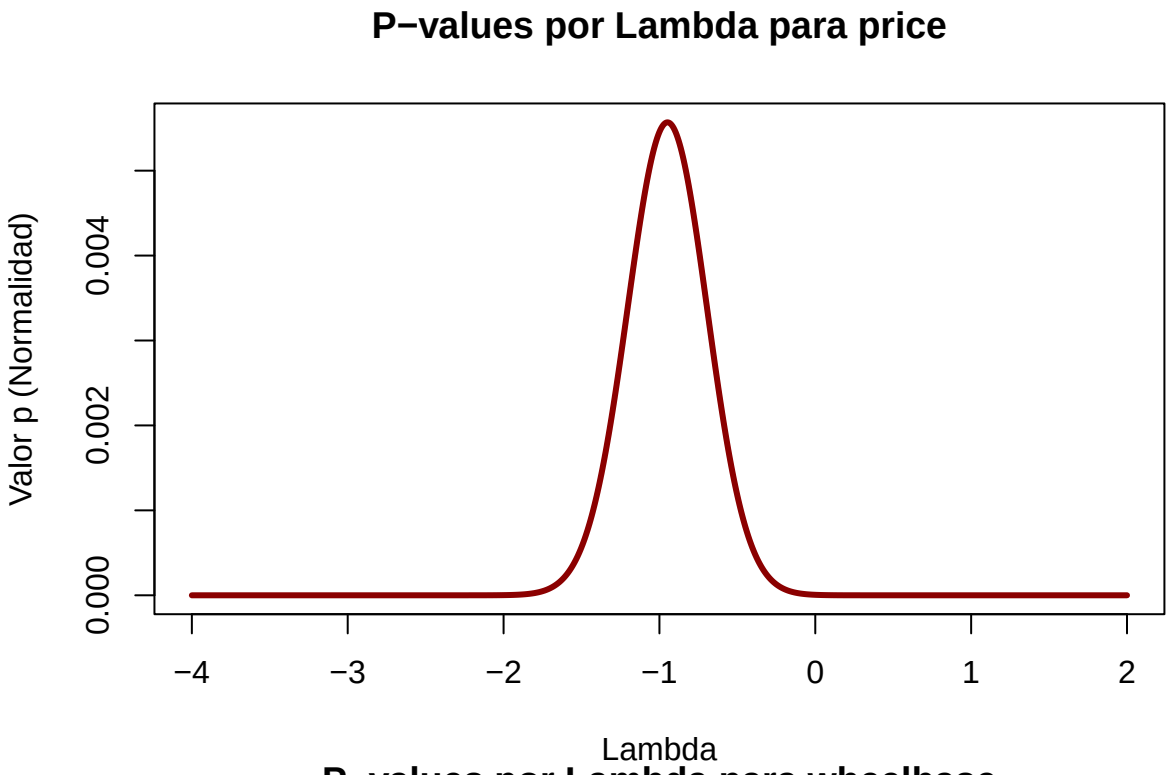
```
library(nortest)
library(VGAM)
```

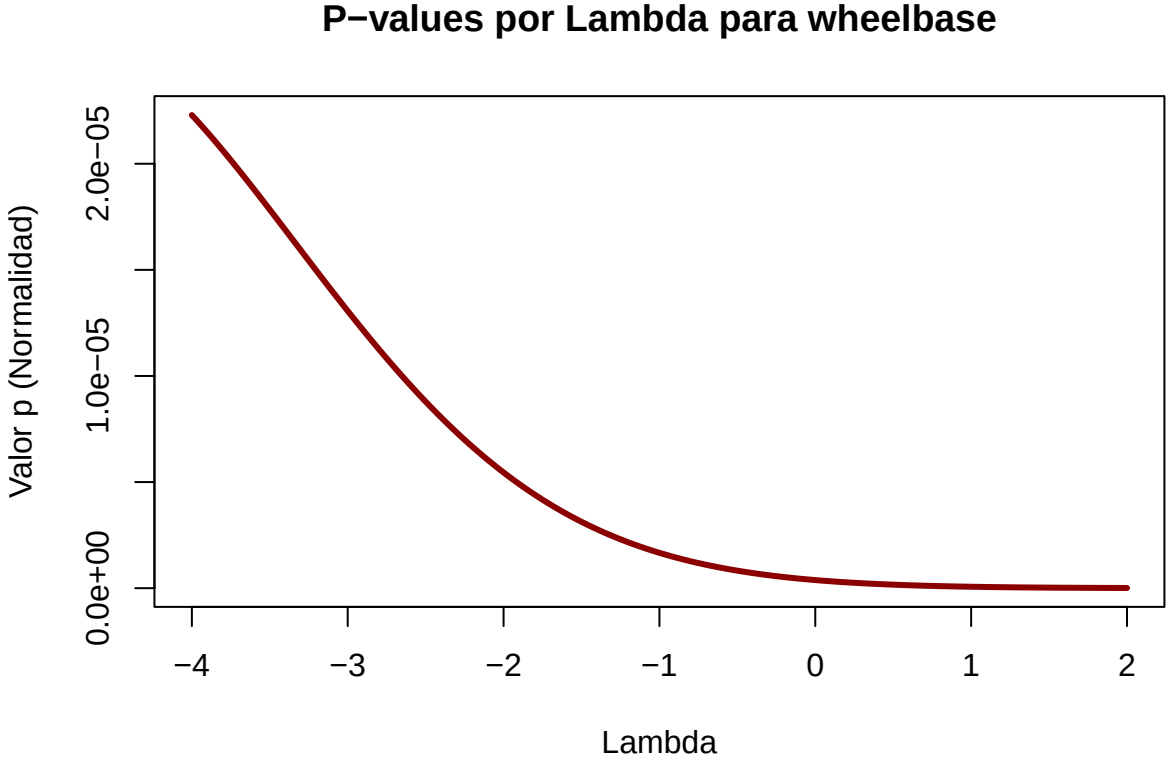
```
## Loading required package: stats4
```

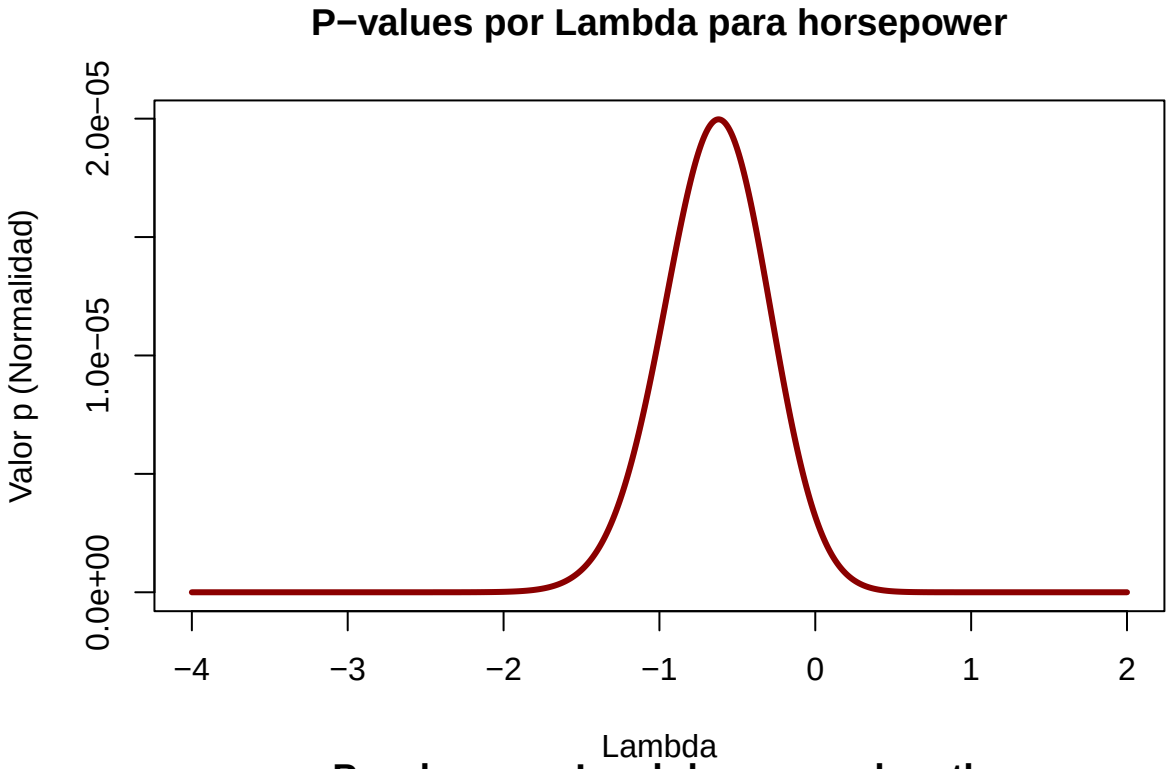
```
## Loading required package: splines
```

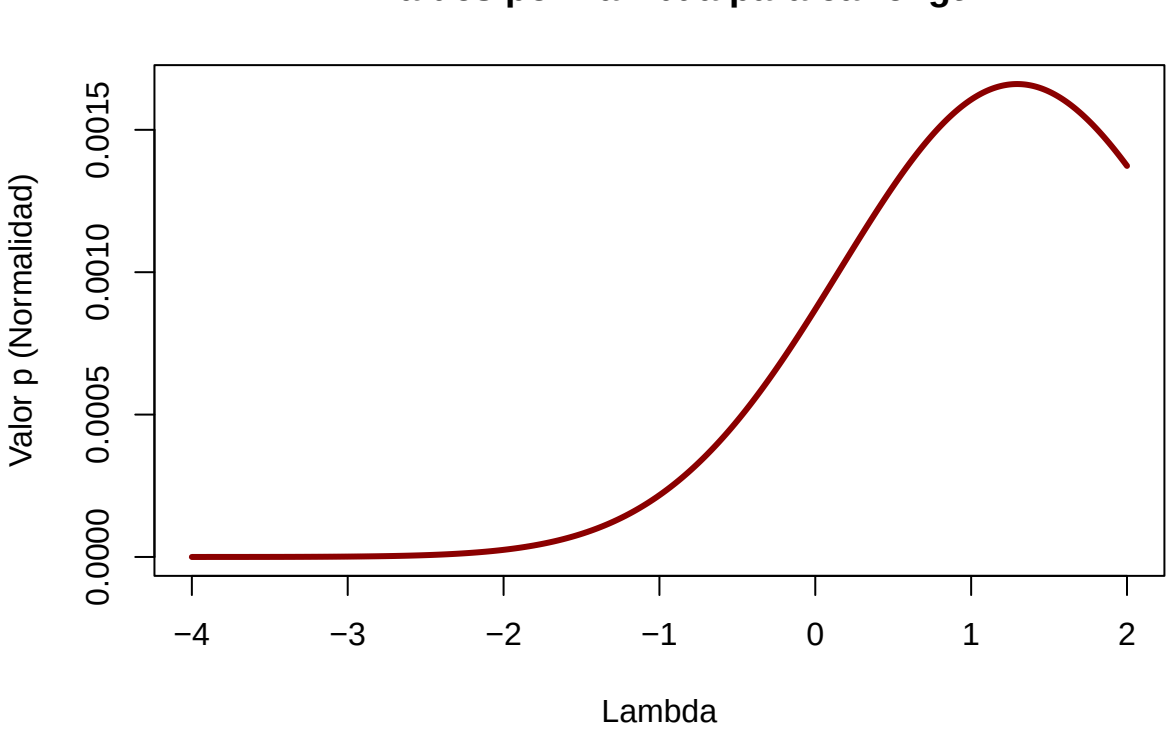
```
plot_p_values_by_lambda <- function(data, variable_name) {
  val_without_0 <- data[[variable_name]][data[[variable_name]] != 0]
  q1c <- quantile(val_without_0, probs = 0.25)
  q3c <- quantile(val_without_0, probs = 0.75)
  ric <- IQR(val_without_0)

  val <- val_without_0[val_without_0 < q3c + 1.5 * ric]

  lp <- seq(-4, 2, 0.001)
  nlp <- length(lp)
  n <- length(val)

  D <- matrix(as.numeric(NA), ncol = 2, nrow = nlp)

  for (i in 1:nlp) {
    d <- yeo.johnson(val, lambda = lp[i])
    p <- ad.test(d)
    D[i, ] <- c(lp[i], p$p.value)
  }

  N <- as.data.frame(D)

  plot(N$V1, N$V2,
       type = "l",
       col = "darkred",
       lwd = 3,
       xlab = "Lambda",
       ylab = "Valor p (Normalidad)",
       main = paste("P-values por Lambda para", variable_name))
}
```

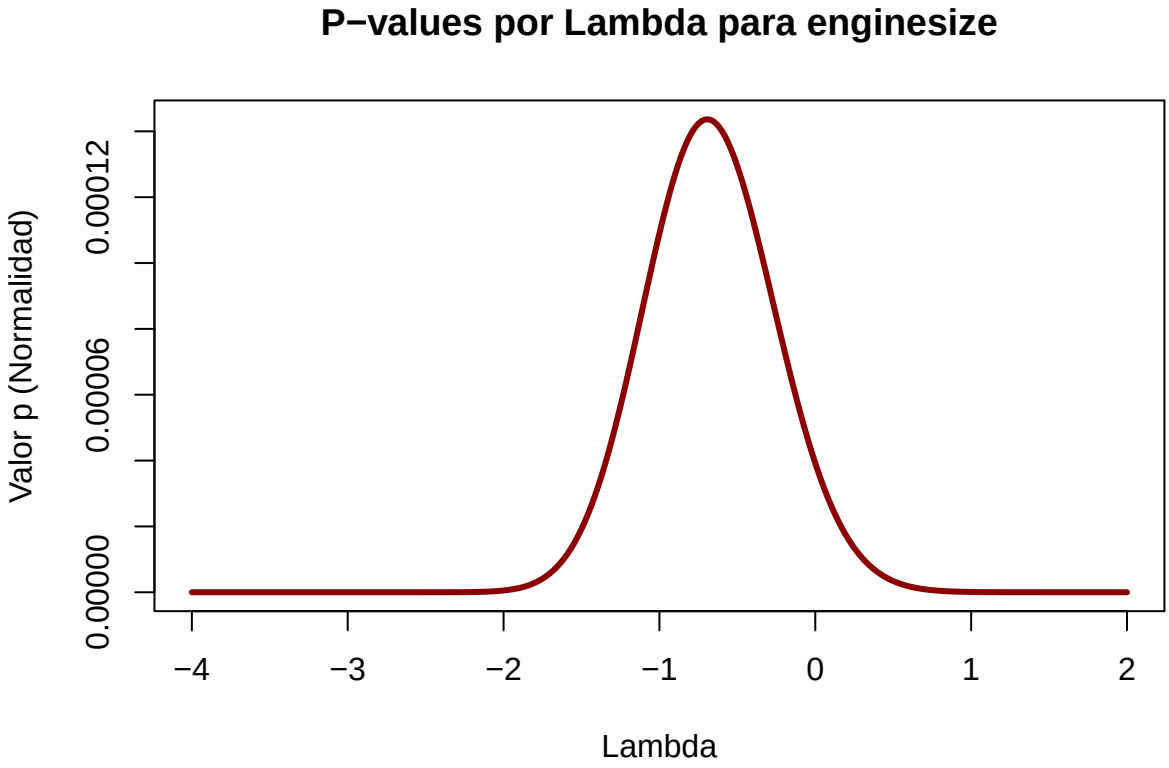
```r
for (i in c('price', 'wheelbase', 'horsepower', 'carlength', 'enginesize')){
  plot_p_values_by_lambda(df_seleccionado, i)
}
```






P−values por Lambda para horsepower
Valor p (Normalidad)
0.0e+00
1.0e−05
2.0e−05
−4
−3
−2
−1
0
1
2
Lambda


P−values por Lambda para carlength
Valor p (Normalidad)
0.0000
0.0005
0.0010
0.0015
−4
−3
−2
−1
0
1
2
Lambda



En las gráficas de p value vs lambda para la pruebad de yeo-johnson demuestra que no es posible alcanzar el valo p mayor a 0.05 para comprobar normalidad en las variables. Sin embargo, como ya se mencionó es importante tener en cuenta que la normalidad no es un requisito estricto para todas las técnicas de análisis estadístico.

```
df_seleccionado
```

```
##       price cylindernumber wheelbase horsepower carlength enginesize
## 1   13495.0               4      88.6        111     168.8        130
## 2   16500.0               4      88.6        111     168.8        130
## 3   16500.0               6      94.5        154     171.2        152
## 4   13950.0               4      99.8        102     176.6        109
## 5   17450.0               5      99.4        115     176.6        136
## 6   15250.0               5      99.8        110     177.3        136
## 7   17710.0               5     105.8        110     192.7        136
## 8   18920.0               5     105.8        110     192.7        136
## 9   23875.0               5     105.8        140     192.7        131
## 11  16430.0               4     101.2        101     176.8        108
## 12  16925.0               4     101.2        101     176.8        108
## 13  20970.0               6     101.2        121     176.8        164
## 14  21105.0               6     101.2        121     176.8        164
## 15  24565.0               6     103.5        121     189.0        164
## 16  30760.0               6     103.5        182     189.0        209
## 19   5151.0               3      88.4         48     141.1         61
## 20   6295.0               4      94.5         70     155.9         90
## 21   6575.0               4      94.5         70     158.8         90
## 22   5572.0               4      93.7         68     157.3         90
## 23   6377.0               4      93.7         68     157.3         90
## 24   7957.0               4      93.7        102     157.3         98
## 25   6229.0               4      93.7         68     157.3         90
```

```
## 26   6692.0             4      93.7         68     157.3         90
## 27   7609.0             4      93.7         68     157.3         90
## 28   8558.0             4      93.7        102     157.3         98
## 29   8921.0             4     103.3         88     174.6        122
## 31   6479.0             4      86.6         58     144.6         92
## 32   6855.0             4      86.6         76     144.6         92
## 33   5399.0             4      93.7         60     150.0         79
## 34   6529.0             4      93.7         76     150.0         92
## 35   7129.0             4      93.7         76     150.0         92
## 36   7295.0             4      96.5         76     163.4         92
## 37   7295.0             4      96.5         76     157.1         92
## 38   7895.0             4      96.5         86     167.5        110
## 39   9095.0             4      96.5         86     167.5        110
## 40   8845.0             4      96.5         86     175.4        110
## 41  10295.0             4      96.5         86     175.4        110
## 42  12945.0             4      96.5        101     175.4        110
## 43  10345.0             4      96.5        100     169.1        110
## 44   6785.0             4      94.3         78     170.7        111
## 45   8916.5             4      94.5         70     155.9         90
## 46   8916.5             4      94.5         70     155.9         90
## 47  11048.0             4      96.0         90     172.6        119
## 51   5195.0             4      93.1         68     159.1         91
## 52   6095.0             4      93.1         68     159.1         91
## 53   6795.0             4      93.1         68     159.1         91
## 54   6695.0             4      93.1         68     166.8         91
## 55   7395.0             4      93.1         68     166.8         91
## 56  10945.0             2      95.3        101     169.0         70
## 57  11845.0             2      95.3        101     169.0         70
## 58  13645.0             2      95.3        101     169.0         70
## 59  15645.0             2      95.3        135     169.0         80
## 60   8845.0             4      98.8         84     177.8        122
## 61   8495.0             4      98.8         84     177.8        122
## 62  10595.0             4      98.8         84     177.8        122
## 63  10245.0             4      98.8         84     177.8        122
## 65  11245.0             4      98.8         84     177.8        122
## 66  18280.0             4     104.9        120     175.0        140
## 76  16503.0             4     102.7        175     178.4        140
## 77   5389.0             4      93.7         68     157.3         92
## 78   6189.0             4      93.7         68     157.3         92
## 79   6669.0             4      93.7         68     157.3         92
## 80   7689.0             4      93.0        102     157.3         98
## 81   9959.0             4      96.3        116     173.0        110
## 82   8499.0             4      96.3         88     173.0        122
## 86   6989.0             4      96.3         88     172.4        122
## 87   8189.0             4      96.3         88     172.4        122
## 88   9279.0             4      96.3        116     172.4        110
## 89   9279.0             4      96.3        116     172.4        110
## 90   5499.0             4      94.5         69     165.3         97
## 92   6649.0             4      94.5         69     165.3         97
## 93   6849.0             4      94.5         69     165.3         97
## 94   7349.0             4      94.5         69     170.2         97
## 95   7299.0             4      94.5         69     165.3         97
## 96   7799.0             4      94.5         69     165.6         97
## 97   7499.0             4      94.5         69     165.3         97
```

```
## 98   7999.0             4      94.5         69     170.2         97
## 99   8249.0             4      95.1         69     162.4         97
## 100  8949.0             4      97.2         97     173.4        120
## 101  9549.0             4      97.2         97     173.4        120
## 102 13499.0             6     100.4        152     181.7        181
## 103 14399.0             6     100.4        152     184.6        181
## 104 13499.0             6     100.4        152     184.6        181
## 105 17199.0             6      91.3        160     170.7        181
## 106 19699.0             6      91.3        200     170.7        181
## 107 18399.0             6      99.2        160     178.5        181
## 108 11900.0             4     107.9         97     186.7        120
## 110 12440.0             4     114.2         97     198.9        120
## 116 16630.0             4     107.9         97     186.7        120
## 119  5572.0             4      93.7         68     157.3         90
## 120  7957.0             4      93.7        102     157.3         98
## 121  6229.0             4      93.7         68     157.3         90
## 122  6692.0             4      93.7         68     167.3         90
## 123  7609.0             4      93.7         68     167.3         98
## 124  8921.0             4     103.3         88     174.6        122
## 126 22018.0             4      94.5        143     168.9        151
## 131  9295.0             4      96.1         90     181.5        132
## 132  9895.0             4      96.1         90     176.8        132
## 133 11850.0             4      99.1        110     186.6        121
## 134 12170.0             4      99.1        110     186.6        121
## 136 15510.0             4      99.1        110     186.6        121
## 137 18150.0             4      99.1        160     186.6        121
## 138 18620.0             4      99.1        160     186.6        121
## 140  7053.0             4      93.7         73     157.9        108
## 141  7603.0             4      93.3         73     157.3        108
## 142  7126.0             4      97.2         82     172.0        108
## 143  7775.0             4      97.2         82     172.0        108
## 144  9960.0             4      97.2         94     172.0        108
## 145  9233.0             4      97.0         82     172.0        108
## 146 11259.0             4      97.0        111     172.0        108
## 147  7463.0             4      97.0         82     173.5        108
## 148 10198.0             4      97.0         94     173.5        108
## 149  8013.0             4      96.9         82     173.6        108
## 150 11694.0             4      96.9        111     173.6        108
## 151  5348.0             4      95.7         62     158.7         92
## 152  6338.0             4      95.7         62     158.7         92
## 153  6488.0             4      95.7         62     158.7         92
## 154  6918.0             4      95.7         62     169.7         92
## 155  7898.0             4      95.7         62     169.7         92
## 156  8778.0             4      95.7         62     169.7         92
## 157  6938.0             4      95.7         70     166.3         98
## 158  7198.0             4      95.7         70     166.3         98
## 161  7738.0             4      95.7         70     166.3         98
## 162  8358.0             4      95.7         70     166.3         98
## 163  9258.0             4      95.7         70     166.3         98
## 164  8058.0             4      94.5         70     168.7         98
## 165  8238.0             4      94.5         70     168.7         98
## 166  9298.0             4      94.5        112     168.7         98
## 167  9538.0             4      94.5        112     168.7         98
## 168  8449.0             4      98.4        116     176.2        146
```

```
## 169  9639.0             4      98.4        116     176.2        146
## 170  9989.0             4      98.4        116     176.2        146
## 171 11199.0             4      98.4        116     176.2        146
## 172 11549.0             4      98.4        116     176.2        146
## 173 17669.0             4      98.4        116     176.2        146
## 174  8948.0             4     102.4         92     175.6        122
## 176  9988.0             4     102.4         92     175.6        122
## 177 10898.0             4     102.4         92     175.6        122
## 178 11248.0             4     102.4         92     175.6        122
## 179 16558.0             6     102.9        161     183.5        171
## 180 15998.0             6     102.9        161     183.5        171
## 181 15690.0             6     104.5        156     187.8        171
## 182 15750.0             6     104.5        156     187.8        161
## 184  7975.0             4      97.3         85     171.7        109
## 186  8195.0             4      97.3         85     171.7        109
## 187  8495.0             4      97.3         85     171.7        109
## 189  9995.0             4      97.3        100     171.7        109
## 190 11595.0             4      94.5         90     159.3        109
## 191  9980.0             4      94.5         90     165.7        109
## 192 13295.0             5     100.4        110     180.2        136
## 194 12290.0             4     100.4         88     183.1        109
## 195 12940.0             4     104.3        114     188.8        141
## 196 13415.0             4     104.3        114     188.8        141
## 197 15985.0             4     104.3        114     188.8        141
## 198 16515.0             4     104.3        114     188.8        141
## 199 18420.0             4     104.3        162     188.8        130
## 200 18950.0             4     104.3        162     188.8        130
## 201 16845.0             4     109.1        114     188.8        141
## 202 19045.0             4     109.1        160     188.8        141
## 203 21485.0             6     109.1        134     188.8        173
## 205 22625.0             4     109.1        114     188.8        141
##     convertible hatchback sedan wagon hardtop
## 1             1         0     0     0       0
## 2             1         0     0     0       0
## 3             0         1     0     0       0
## 4             0         0     1     0       0
## 5             0         0     1     0       0
## 6             0         0     1     0       0
## 7             0         0     1     0       0
## 8             0         0     0     1       0
## 9             0         0     1     0       0
## 11            0         0     1     0       0
## 12            0         0     1     0       0
## 13            0         0     1     0       0
## 14            0         0     1     0       0
## 15            0         0     1     0       0
## 16            0         0     1     0       0
## 19            0         1     0     0       0
## 20            0         1     0     0       0
## 21            0         0     1     0       0
## 22            0         1     0     0       0
## 23            0         1     0     0       0
## 24            0         1     0     0       0
## 25            0         1     0     0       0
```

```
## 26          0         0     1     0         0
## 27          0         0     1     0         0
## 28          0         0     1     0         0
## 29          0         0     0     1         0
## 31          0         1     0     0         0
## 32          0         1     0     0         0
## 33          0         1     0     0         0
## 34          0         1     0     0         0
## 35          0         1     0     0         0
## 36          0         0     1     0         0
## 37          0         0     0     1         0
## 38          0         1     0     0         0
## 39          0         1     0     0         0
## 40          0         0     1     0         0
## 41          0         0     1     0         0
## 42          0         0     1     0         0
## 43          0         0     1     0         0
## 44          0         0     1     0         0
## 45          0         0     1     0         0
## 46          0         0     1     0         0
## 47          0         1     0     0         0
## 51          0         1     0     0         0
## 52          0         1     0     0         0
## 53          0         1     0     0         0
## 54          0         0     1     0         0
## 55          0         0     1     0         0
## 56          0         1     0     0         0
## 57          0         1     0     0         0
## 58          0         1     0     0         0
## 59          0         1     0     0         0
## 60          0         1     0     0         0
## 61          0         0     1     0         0
## 62          0         1     0     0         0
## 63          0         0     1     0         0
## 65          0         1     0     0         0
## 66          0         0     1     0         0
## 76          0         1     0     0         0
## 77          0         1     0     0         0
## 78          0         1     0     0         0
## 79          0         1     0     0         0
## 80          0         1     0     0         0
## 81          0         1     0     0         0
## 82          0         1     0     0         0
## 86          0         0     1     0         0
## 87          0         0     1     0         0
## 88          0         0     1     0         0
## 89          0         0     1     0         0
## 90          0         0     1     0         0
## 92          0         0     1     0         0
## 93          0         0     1     0         0
## 94          0         0     0     1         0
## 95          0         0     1     0         0
## 96          0         1     0     0         0
## 97          0         0     1     0         0
```

```
## 98          0         0     0     1        0
## 99          0         0     0     0        1
## 100         0         1     0     0        0
## 101         0         0     1     0        0
## 102         0         0     1     0        0
## 103         0         0     0     1        0
## 104         0         0     1     0        0
## 105         0         1     0     0        0
## 106         0         1     0     0        0
## 107         0         1     0     0        0
## 108         0         0     1     0        0
## 110         0         0     0     1        0
## 116         0         0     1     0        0
## 119         0         1     0     0        0
## 120         0         1     0     0        0
## 121         0         1     0     0        0
## 122         0         0     1     0        0
## 123         0         0     1     0        0
## 124         0         0     0     1        0
## 126         0         1     0     0        0
## 131         0         0     0     1        0
## 132         0         1     0     0        0
## 133         0         1     0     0        0
## 134         0         0     1     0        0
## 136         0         0     1     0        0
## 137         0         1     0     0        0
## 138         0         0     1     0        0
## 140         0         1     0     0        0
## 141         0         1     0     0        0
## 142         0         0     1     0        0
## 143         0         0     1     0        0
## 144         0         0     1     0        0
## 145         0         0     1     0        0
## 146         0         0     1     0        0
## 147         0         0     0     1        0
## 148         0         0     0     1        0
## 149         0         0     0     1        0
## 150         0         0     0     1        0
## 151         0         1     0     0        0
## 152         0         1     0     0        0
## 153         0         1     0     0        0
## 154         0         0     0     1        0
## 155         0         0     0     1        0
## 156         0         0     0     1        0
## 157         0         0     1     0        0
## 158         0         1     0     0        0
## 161         0         0     1     0        0
## 162         0         1     0     0        0
## 163         0         0     1     0        0
## 164         0         0     1     0        0
## 165         0         1     0     0        0
## 166         0         0     1     0        0
## 167         0         1     0     0        0
## 168         0         0     0     0        1
```

```
## 169          0          0     0     0     1
## 170          0          1     0     0     0
## 171          0          0     0     0     1
## 172          0          1     0     0     0
## 173          1          0     0     0     0
## 174          0          0     1     0     0
## 176          0          1     0     0     0
## 177          0          0     1     0     0
## 178          0          1     0     0     0
## 179          0          1     0     0     0
## 180          0          1     0     0     0
## 181          0          0     1     0     0
## 182          0          0     0     1     0
## 184          0          0     1     0     0
## 186          0          0     1     0     0
## 187          0          0     1     0     0
## 189          0          0     1     0     0
## 190          1          0     0     0     0
## 191          0          1     0     0     0
## 192          0          0     1     0     0
## 194          0          0     0     1     0
## 195          0          0     1     0     0
## 196          0          0     0     1     0
## 197          0          0     1     0     0
## 198          0          0     0     1     0
## 199          0          0     1     0     0
## 200          0          0     0     1     0
## 201          0          0     1     0     0
## 202          0          0     1     0     0
## 203          0          0     1     0     0
## 205          0          0     1     0     0
```

ANALIZA LOS DATOS Y PREGUNTA BASE

Dado el contexto del proyecto y los datos proporcionados, elegiré la herramienta estadística de **regresión lineal múltiple** como una de las técnicas para analizar y validar el modelo. Además, utilizaré **pruebas de hipótesis de medias** para evaluar la significancia de los coeficientes de regresión y la influencia de las variables predictoras en la variable de respuesta.

Regresión Lineal Múltiple: - **Justificación**: La regresión lineal múltiple es una herramienta adecuada para analizar la relación entre múltiples variables predictoras (características de los automóviles) y una variable de respuesta continua (el precio de los automóviles). Dado que el objetivo principal parece ser predecir el precio de los automóviles en función de diversas características, la regresión lineal múltiple permite modelar esta relación y evaluar la contribución de cada variable predictora.

Pruebas de Hipótesis de Medias: - **Justificación**: Las pruebas de hipótesis de medias son útiles para determinar si existe una diferencia significativa en la variable de respuesta (precio) entre diferentes grupos o categorías de variables predictoras categóricas (como el tipo de carrocería o el tipo de motor). Esto puede proporcionar información valiosa sobre cómo las características categóricas influyen en el precio de los automóviles.

Para validar el modelo de regresión lineal múltiple y evaluar los supuestos requeridos por el modelo, realizaré las siguientes acciones:

1. **Linealidad**: Verificaré la linealidad mediante gráficos de dispersión de las variables predictoras frente a la variable de respuesta y asegurándome de que no haya patrones no lineales evidentes en los residuos.

2. **Independencia de Errores**: Examinaré la independencia de errores mediante gráficos de autocorrelación de los residuos y asegurándome de que no haya patrones discernibles.
3. **Homocedasticidad**: Evaluaré la homocedasticidad mediante gráficos de residuos frente a valores ajustados y pruebas estadísticas como el test de Breusch-Pagan o White para verificar la constancia de la varianza de los residuos.
4. **Normalidad de Errores**: Realizaré pruebas de normalidad de los residuos, como el test de Shapiro-Wilk o gráficos QQ-plot, para verificar si los residuos siguen una distribución normal.
5. **No Multicolinealidad**: Comprobaré la multicolinealidad calculando la matriz de correlación entre las variables predictoras y evaluando si existe una alta correlación entre ellas.

Una vez realizadas estas verificaciones y, si es necesario, aplicadas las transformaciones adecuadas en caso de violación de supuestos, procederé a construir el modelo de regresión lineal múltiple. Luego, utilizaré pruebas de hipótesis para evaluar la significancia de los coeficientes de regresión y determinar qué variables predictoras tienen un impacto significativo en el precio de los automóviles. Esto ayudará a proporcionar una comprensión más sólida de cómo se relacionan las características de los automóviles con sus precios.

Modelo de regresion lineal multiple

```
modelo_regresion <- lm(price ~ ., data = df_seleccionado)

summary(modelo_regresion)
```

```
##
## Call:
## lm(formula = price ~ ., data = df_seleccionado)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5060.4 -1390.3  -204.6   931.8  8333.9
##
## Coefficients: (1 not defined because of singularities)
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -34508.765   5821.400  -5.928 2.01e-08 ***
## cylindernumber    335.406    516.027   0.650 0.516694
## wheelbase         252.540     79.016   3.196 0.001697 **
## horsepower         90.527     11.279   8.026 2.61e-13 ***
## carlength          44.249     42.624   1.038 0.300866
## enginesize          6.191     21.268   0.291 0.771366
## convertible      6686.099   1698.310   3.937 0.000126 ***
## hatchback        2195.988   1279.660   1.716 0.088200 .
## sedan            2651.576   1302.308   2.036 0.043492 *
## wagon            1565.714   1397.376   1.120 0.264294
## hardtop                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2364 on 151 degrees of freedom
## Multiple R-squared:  0.7706, Adjusted R-squared:  0.7569
## F-statistic: 56.36 on 9 and 151 DF,  p-value: < 2.2e-16
```

Los resultados de la regresión lineal múltiple indican lo siguiente:

1. **Residuals**: Esta sección muestra estadísticas descriptivas de los residuos del modelo. Los residuos son las diferencias entre los valores observados y los valores predichos por el modelo. En este caso, los

residuos varían desde -5060.4 hasta 8333.9.

2. **Coefficients**: Esta tabla muestra los coeficientes de regresión estimados para cada una de las variables independientes en el modelo:
    - **Intercept**: El valor estimado del intercepto es -34508.765. Representa el valor estimado de la variable dependiente (price) cuando todas las variables independientes son iguales a cero.
    - **cylindernumber**: El coeficiente estimado para cylindernumber es 335.406, pero no es significativamente diferente de cero, ya que el valor p es 0.516694. Esto sugiere que no hay evidencia suficiente para afirmar que cylindernumber tiene un efecto significativo en el precio.
    - **wheelbase**: El coeficiente estimado para wheelbase es 252.540, y es estadísticamente significativo (valor p = 0.001697). Esto sugiere que existe una relación significativa entre la longitud de la distancia entre ejes (wheelbase) y el precio.
    - **horsepower**: El coeficiente estimado para horsepower es 90.527, y es altamente significativo (valor p = 2.61e-13). Esto indica que la potencia del motor (horsepower) tiene un efecto significativo en el precio.
    - **carlength**: El coeficiente estimado para carlength es 44.249, pero no es significativamente diferente de cero (valor p = 0.300866). No hay evidencia suficiente para afirmar que la longitud del automóvil (carlength) tiene un efecto significativo en el precio.
    - **enginesize**: El coeficiente estimado para enginesize es 6.191, pero no es significativamente diferente de cero (valor p = 0.771366). No hay evidencia suficiente para afirmar que el tamaño del motor (enginesize) tiene un efecto significativo en el precio.
    - **convertible**: El coeficiente estimado para convertible es 6686.099 y es significativo (valor p = 0.000126). Esto sugiere que el tipo de automóvil convertible tiene un efecto significativo en el precio.
    - **hatchback**: El coeficiente estimado para hatchback es 2195.988, pero no es significativamente diferente de cero (valor p = 0.088200). No hay evidencia suficiente para afirmar que el tipo de automóvil hatchback tiene un efecto significativo en el precio.
    - **sedan**: El coeficiente estimado para sedan es 2651.576 y es significativo (valor p = 0.043492). Esto sugiere que el tipo de automóvil sedán tiene un efecto significativo en el precio.
    - **wagon**: El coeficiente estimado para wagon es 1565.714, pero no es significativamente diferente de cero (valor p = 0.264294). No hay evidencia suficiente para afirmar que el tipo de automóvil wagon tiene un efecto significativo en el precio.
    - **hardtop**: La variable hardtop tiene un valor "NA" en todos los coeficientes, lo que sugiere que puede haber problemas de multicolinealidad o falta de variabilidad en esta variable.
3. **Residual standard error**: Esta es una medida de la variabilidad no explicada por el modelo. En este caso, es de aproximadamente 2364.
4. **Multiple R-squared**: Representa la proporción de la variabilidad en la variable dependiente (price) que es explicada por el modelo. En este caso, el modelo explica aproximadamente el 77.06% de la variabilidad en el precio.
5. **Adjusted R-squared**: Similar al R-cuadrado múltiple, pero ajustado por el número de variables independientes en el modelo. En este caso, es de aproximadamente 75.69%.
6. **F-statistic**: Esta estadística se utiliza para evaluar si al menos una de las variables independientes tiene un efecto significativo en la variable dependiente. Un valor pequeño del p-valor (p-value) indica que al menos una variable es significativa. En este caso, el p-valor es extremadamente pequeño (p-value: < 2.2e-16), lo que sugiere que al menos una de las variables independientes es significativa en la predicción del precio.

En resumen, el modelo de regresión lineal múltiple sugiere que las variables horsepower, wheelbase, convertible y sedan son significativas para predecir el precio de los automóviles, mientras que otras variables no lo son.

A partir de aquí comenzaremos a retirar variables una por una dependiendo de su valor p y observaremos los resultados de los modelos para mantener el mejor. De primera quitaremos **hardtop** por su error y **enginezise** por tener el mayor valor P.

```
df_seleccionado_sin_vars <- df_seleccionado[, !(names(df_seleccionado) %in% c("hardtop", "enginesize"))]

modelo <- lm(price ~ ., data = df_seleccionado_sin_vars)

summary(modelo)
```

```
##
## Call:
## lm(formula = price ~ ., data = df_seleccionado_sin_vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5015.2 -1397.5  -212.0   896.3  8478.1
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -35306.931   5119.812  -6.896 1.34e-10 ***
## cylindernumber    454.576    313.230   1.451 0.148771
## wheelbase         253.975     78.625   3.230 0.001516 **
## horsepower         92.140      9.794   9.408  < 2e-16 ***
## carlength          49.119     39.085   1.257 0.210782
## convertible      6679.749   1693.049   3.945 0.000121 ***
## hatchback        2093.443   1226.513   1.707 0.089897 .
## sedan            2521.847   1220.000   2.067 0.040422 *
## wagon            1424.698   1306.778   1.090 0.277334
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2357 on 152 degrees of freedom
## Multiple R-squared:  0.7705, Adjusted R-squared:  0.7584
## F-statistic: 63.78 on 8 and 152 DF,  p-value: < 2.2e-16
```

Sigue retirar la variable **wagon**

```
df_seleccionado_sin_vars <- df_seleccionado[, !(names(df_seleccionado) %in% c("hardtop", "enginesize",

modelo <- lm(price ~ ., data = df_seleccionado_sin_vars)

summary(modelo)
```

```
##
## Call:
## lm(formula = price ~ ., data = df_seleccionado_sin_vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5021.4 -1399.8  -209.4   917.5  8499.3
##
## Coefficients:
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -34928.356   5111.170  -6.834 1.84e-10 ***
## cylindernumber    485.196    312.161   1.554  0.12218
## wheelbase         255.527     78.660   3.248  0.00143 **
## horsepower         90.491      9.682   9.346  < 2e-16 ***
## carlength          53.043     38.943   1.362  0.17518
## convertible      5551.346   1340.668   4.141 5.70e-05 ***
## hatchback         949.369    635.320   1.494  0.13715
## sedan            1336.350    553.528   2.414  0.01695 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2358 on 153 degrees of freedom
## Multiple R-squared:  0.7687, Adjusted R-squared:  0.7581
## F-statistic: 72.64 on 7 and 153 DF,  p-value: < 2.2e-16
```

Ahora la variable **carlength**

```
df_seleccionado_sin_vars <- df_seleccionado[, !(names(df_seleccionado) %in% c("hardtop", "enginesize",

modelo <- lm(price ~ ., data = df_seleccionado_sin_vars)

summary(modelo)
```

```
##
## Call:
## lm(formula = price ~ ., data = df_seleccionado_sin_vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4929.5 -1491.2  -271.4  1133.5  8373.5
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -33945.746   5074.034  -6.690 3.89e-10 ***
## cylindernumber    453.488    312.155   1.453   0.1483
## wheelbase         334.714     53.133   6.300 2.98e-09 ***
## horsepower         97.346      8.294  11.738  < 2e-16 ***
## convertible      5561.100   1344.366   4.137 5.78e-05 ***
## hatchback         675.854    604.421   1.118   0.2652
## sedan            1251.441    551.532   2.269   0.0247 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2365 on 154 degrees of freedom
## Multiple R-squared:  0.7659, Adjusted R-squared:  0.7568
## F-statistic: 83.97 on 6 and 154 DF,  p-value: < 2.2e-16
```

```
df_seleccionado_sin_vars <- df_seleccionado[, !(names(df_seleccionado) %in% c("hardtop", "enginesize",

modelo <- lm(price ~ ., data = df_seleccionado_sin_vars)

summary(modelo)
```

```
##
## Call:
```

```
## lm(formula = price ~ ., data = df_seleccionado_sin_vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4944.1 -1461.3  -159.3  1204.6  8377.6
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -31475.867   4571.716  -6.885 1.35e-10 ***
## cylindernumber    432.349    311.833   1.386   0.1676
## wheelbase         313.425     49.645   6.313 2.74e-09 ***
## horsepower         98.925      8.179  12.095  < 2e-16 ***
## convertible      4976.614   1239.586   4.015 9.24e-05 ***
## sedan             820.394    394.762   2.078   0.0393 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2367 on 155 degrees of freedom
## Multiple R-squared:  0.764,  Adjusted R-squared:  0.7564
## F-statistic: 100.3 on 5 and 155 DF,  p-value: < 2.2e-16
```

```r
df_seleccionado_sin_vars <- df_seleccionado[, !(names(df_seleccionado) %in% c("hardtop", "enginesize",

modelo <- lm(price ~ ., data = df_seleccionado_sin_vars)

summary(modelo)
```

```
##
## Call:
## lm(formula = price ~ ., data = df_seleccionado_sin_vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4487.5 -1540.5  -180.8  1203.2  8836.1
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -30292.968   4504.659  -6.725 3.14e-10 ***
## wheelbase      314.239     49.788   6.311 2.73e-09 ***
## horsepower     104.249      7.243  14.393  < 2e-16 ***
## convertible   4878.128   1241.202   3.930 0.000127 ***
## sedan          884.032    393.242   2.248 0.025974 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2373 on 156 degrees of freedom
## Multiple R-squared:  0.7611, Adjusted R-squared:  0.7549
## F-statistic: 124.2 on 4 and 156 DF,  p-value: < 2.2e-16
```

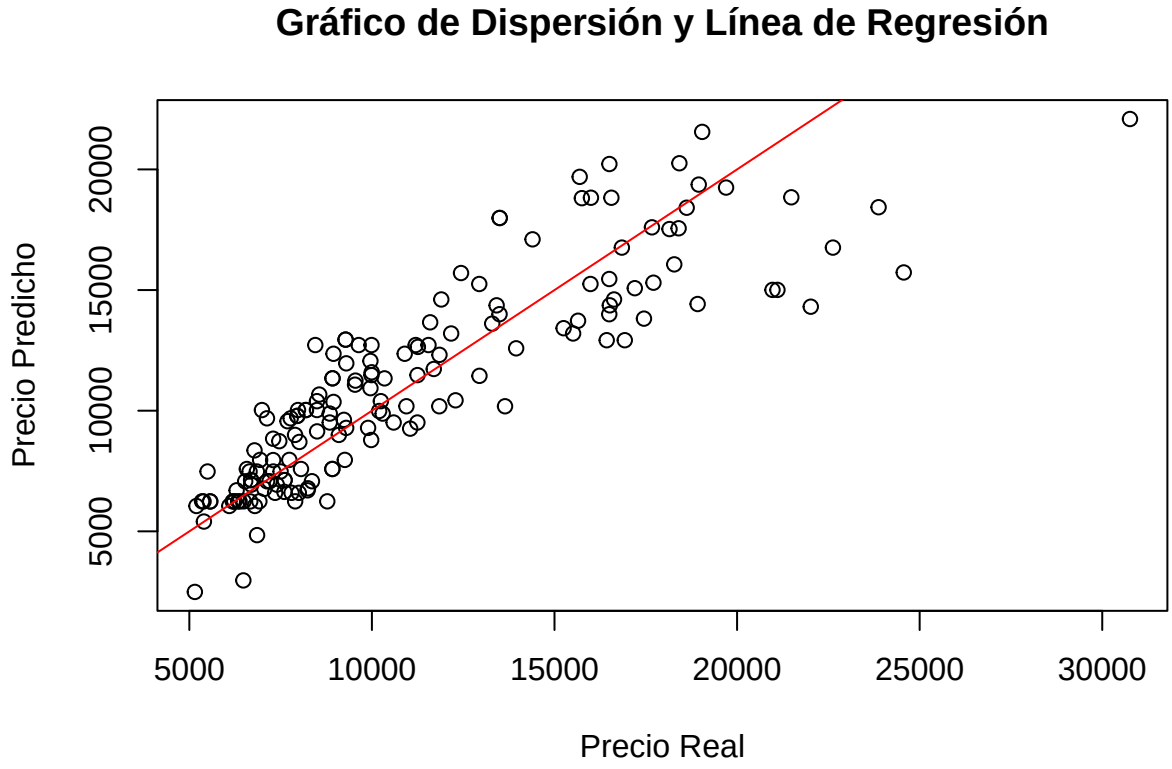
En general, el modelo tiene un coeficiente de determinación ajustado (Adjusted R-squared) de aproximadamente 0.7549, lo que significa que alrededor del 75.49% de la variabilidad en el precio se explica por las variables incluidas en el modelo. El F-statistic es significativo (p-valor < 0.001), lo que sugiere que al menos una de las variables independientes tiene un efecto significativo en el precio.

Este modelo, después de retirar las variables que no eran estadísticamente significativas, muestra una buena capacidad para explicar la variabilidad en el precio de los automóviles. Las variables "wheelbase",

“horsepower”, “convertible” y “sedan” son las que más contribuyen a la predicción del precio.

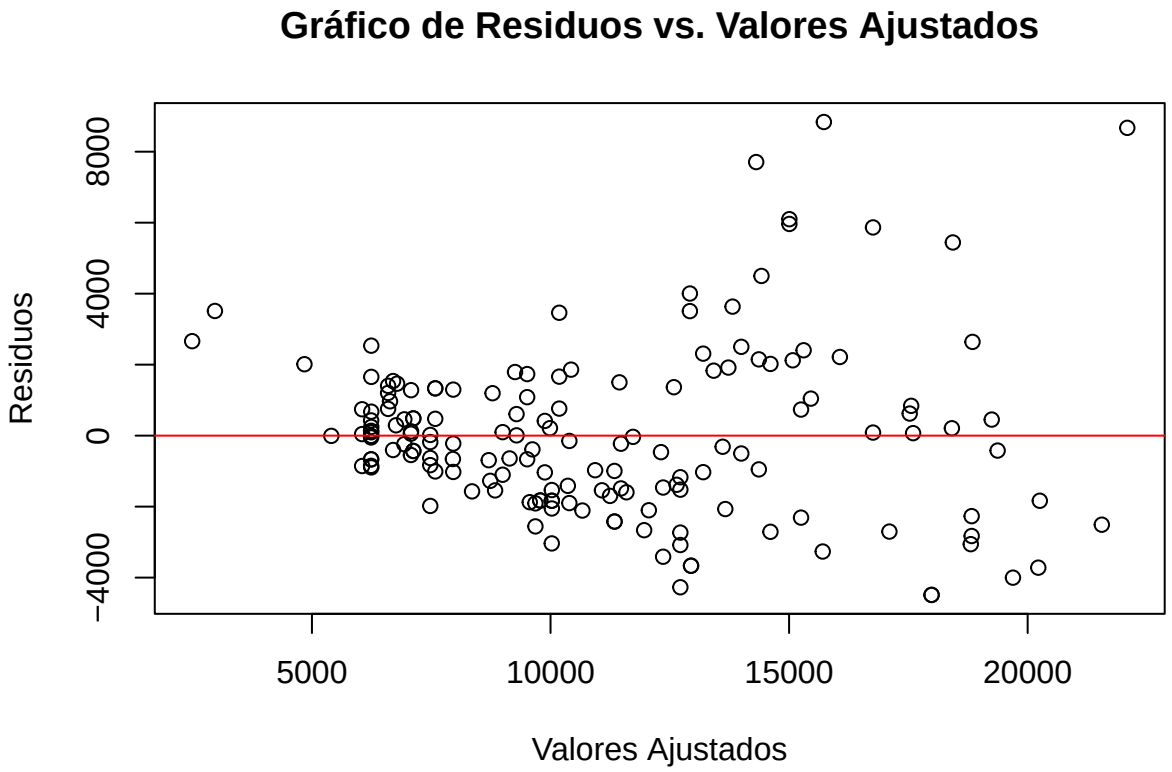
###Visualización del modelo

```
# Gráfico de dispersión y línea de regresión
plot(df_seleccionado_sin_vars$price, fitted(modelo),
     xlab = "Precio Real", ylab = "Precio Predicho",
     main = "Gráfico de Dispersión y Línea de Regresión")
abline(0, 1, col = "red")
```



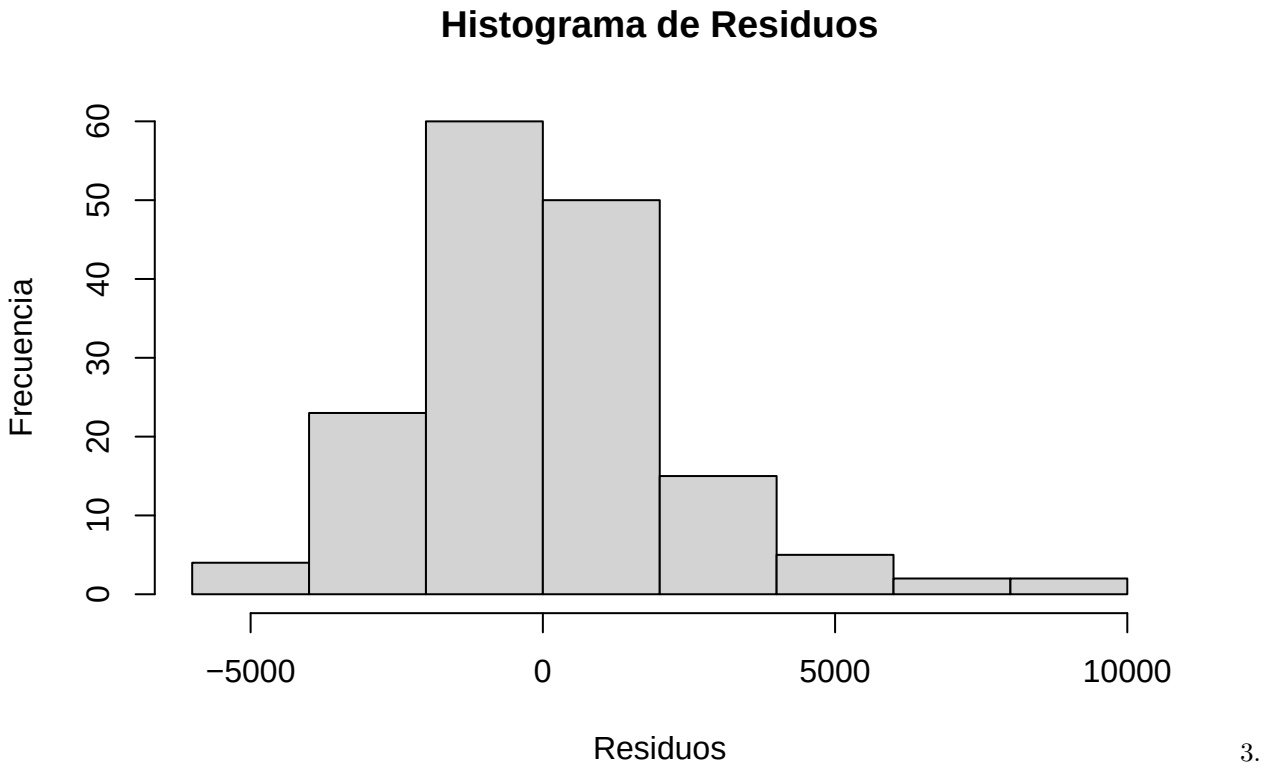
1. **Gráfico de Dispersión y Línea de Regresión**: La mayoría de los puntos de datos se encuentran cerca de la línea de regresión, lo que indica que el modelo se ajusta razonablemente bien a los datos observados. Los valores reales y predichos están en buena concordancia, lo que sugiere una relación lineal entre las variables predictoras y la variable objetivo.

```
# Gráfico de residuos vs. Valores ajustados
plot(fitted(modelo), residuals(modelo),
     xlab = "Valores Ajustados", ylab = "Residuos",
     main = "Gráfico de Residuos vs. Valores Ajustados")
abline(0, 0, col = "red")
```



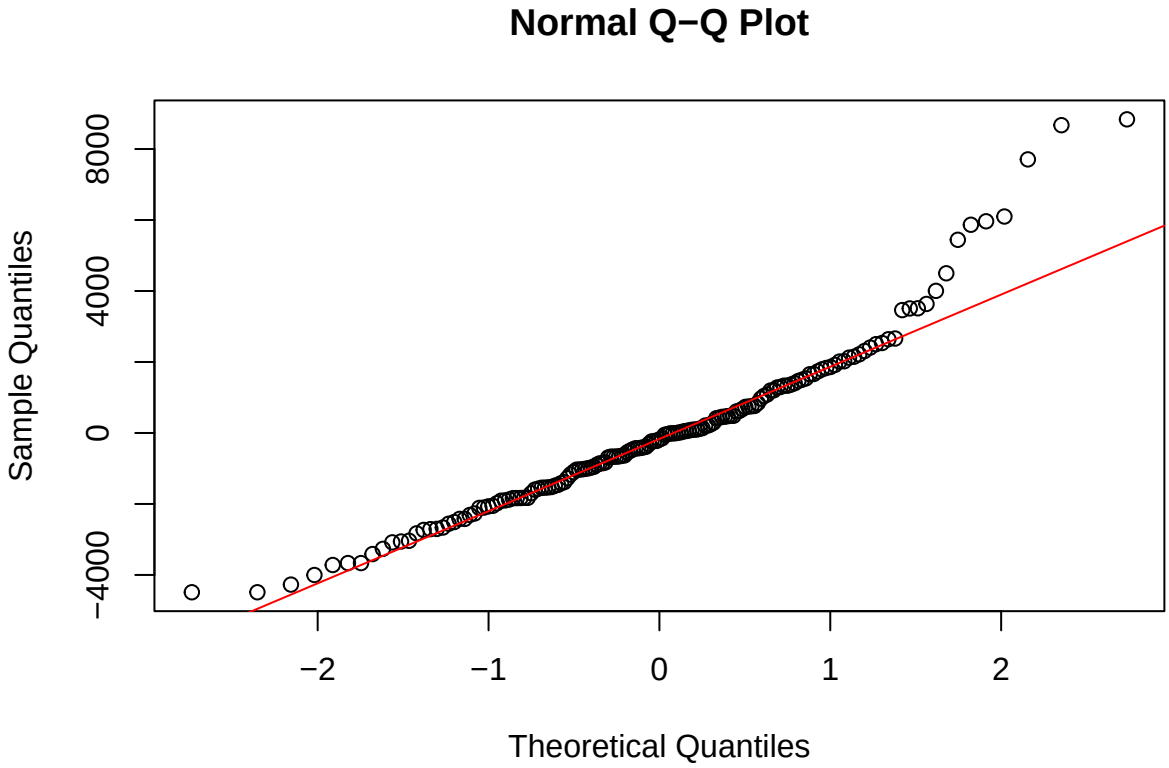
2. **Gráfico de Residuos vs. Valores Ajustados**: En este gráfico, los residuos parecen distribuirse aleatoriamente alrededor de la línea cero, lo que indica que no hay un patrón sistemático en los residuos. Esto sugiere que el modelo no tiene problemas de sesgo y que la varianza de los errores es constante (homocedasticidad).

```
# Histograma de residuos
hist(residuals(modelo),
     xlab = "Residuos", ylab = "Frecuencia",
     main = "Histograma de Residuos")
```



3. **Histograma de Residuos**: El histograma de residuos muestra una forma similar a una campana gaussiana, lo que sugiere que los residuos siguen aproximadamente una distribución normal. Esto es un buen indicador de que el supuesto de normalidad de los residuos se cumple, lo que es importante para realizar inferencias estadísticas.

```
# Gráfico QQ de residuos
qqnorm(residuals(modelo))
qqline(residuals(modelo), col = "red")
```



4. **Gráfico QQ de Residuos**: En el gráfico QQ de residuos, los puntos se ajustan muy bien a la línea diagonal, lo que también respalda la suposición de normalidad de los residuos. Esto sugiere que los residuos siguen una distribución normal.

En conclusión, basándonos en el análisis de las gráficas y los residuos, el modelo de regresión lineal múltiple parece ser una buena representación de la relación entre las variables predictoras (cylindernumber, wheelbase, horsepower, convertible, sedan) y el precio de los automóviles. Los supuestos clave del modelo, como la linealidad, homocedasticidad y normalidad de los residuos, parecen cumplirse satisfactoriamente. Sin embargo, es importante recordar que un buen ajuste del modelo no garantiza la causalidad ni la ausencia de otros posibles predictores relevantes que no se hayan incluido en el modelo. Por lo tanto, estos resultados son una base sólida para el análisis, pero siempre es importante considerar el contexto y las limitaciones del estudio.