

Primer momento de retroalimentación

Samantha Daniela Guanipa Ugas A01703936

2023-08-23

Planteamiento del problema

Una empresa automovilística china aspira a entrar en el mercado estadounidense. Desea establecer allí una unidad de fabricación y producir automóviles localmente para competir con sus contrapartes estadounidenses y europeas. Contrataron una empresa de consultoría de automóviles para identificar los principales factores de los que depende el precio de los automóviles, específicamente, en el mercado estadounidense, ya que pueden ser muy diferentes del mercado chino. Esencialmente, la empresa quiere saber:

1. Qué variables son significativas para predecir el precio de un automóvil
2. Qué tan bien describen esas variables el precio de un automóvil

```
M=read.csv("precios_autos.csv") #leer la base de datos
```

Teniendo la base de datos proporcionada, se subirá y será leída

Análisis de la base de datos

- Exploración de la base de datos

1. Cálculo de las medidas estadísticas de las variables cuantitativas y cualitativas

```
# Se crea un data frame con las variables numericas
```

```
numerical_df <- M %>% select(wheelbase, carlength, carwidth, carheight, curbweight, enginesize, stroke, compressionratio, horsepower, peakrpm, citympg, highwaympg, price)
head(numerical_df)
```

a) Cálculo de las medidas estadísticas de las variables cuantitativas

wheelbase	carlength	carwidth	carheight	curbweight	enginesize	stroke	compressionratio	horsepower	peakrpm	citympg	highwaympg	price
88.6	168.8	64.1	48.8	2548	130	2.68	9.0	111	5000	21	27	13495
88.6	168.8	64.1	48.8	2548	130	2.68	9.0	111	5000	21	27	16500
94.5	171.2	65.5	52.4	2823	152	3.47	9.0	154	5000	19	26	16500
99.8	176.6	66.2	54.3	2337	109	3.40	10.0	102	5500	24	30	13950
99.4	176.6	66.4	54.3	2824	136	3.40	8.0	115	5500	18	22	17450
99.8	177.3	66.3	53.1	2507	136	3.40	8.5	110	5500	19	25	15250

```
#Se calcula el resumen de la variable, su desviacion estandar, su varianza, su kurtosis y su sesgo
```

```
m0 <- round(c(as.numeric(summary(numerical_df$wheelbase)), sd(numerical_df$wheelbase), var(numerical_df$wheelbase)), 2)
```

```

m1 <- round(c(as.numeric(summary(numerical_df$carlength)), sd(numerical_df$carlength), var(numerical_df$carlength)))
m2 <- round(c(as.numeric(summary(numerical_df$carwidth)), sd(numerical_df$carwidth), var(numerical_df$carwidth)))
m3 <- round(c(as.numeric(summary(numerical_df$carheight)), sd(numerical_df$carheight), var(numerical_df$carheight)))
m4 <- round(c(as.numeric(summary(numerical_df$curbweight)), sd(numerical_df$curbweight), var(numerical_df$curbweight)))
m5 <- round(c(as.numeric(summary(numerical_df$enginesize)), sd(numerical_df$enginesize), var(numerical_df$enginesize)))
m6 <- round(c(as.numeric(summary(numerical_df$stroke)), sd(numerical_df$stroke), var(numerical_df$stroke)))
m7 <- round(c(as.numeric(summary(numerical_df$compressionratio)), sd(numerical_df$compressionratio), var(numerical_df$compressionratio)))
m8 <- round(c(as.numeric(summary(numerical_df$horsepower)), sd(numerical_df$horsepower), var(numerical_df$horsepower)))
m9 <- round(c(as.numeric(summary(numerical_df$peakrpm)), sd(numerical_df$peakrpm), var(numerical_df$peakrpm)))
m10 <- round(c(as.numeric(summary(numerical_df$citympg)), sd(numerical_df$citympg), var(numerical_df$citympg)))
m11 <- round(c(as.numeric(summary(numerical_df$highwaympg)), sd(numerical_df$highwaympg), var(numerical_df$highwaympg)))
m12 <- round(c(as.numeric(summary(numerical_df$price)), sd(numerical_df$price), var(numerical_df$price)))

m<-as.data.frame(rbind(m0,m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12))
row.names(m)=c("Wheelbase","Car length","Car width", "Car height", "Curb weight", "Engine size", "Stroke")
names(m)=c("Minimo", "Q1", "Mediana", "Media", "Q3", "Máximo", "Desviación Estándar", "Varianza", "Curtosis", "Sesgo")
m

```

	Minimo	Q1	Mediana	Media	Q3	Máximo	Desviación Estándar	Varianza	Curtosis	Sesgo
Wheelbase	86.60	94.50	97.00	98.757	102.40	120.90	6.022	36.262	0.925	1.035
Car length	141.10	166.30	173.20	174.049	183.10	208.10	12.337	152.209	-0.138	0.154
Car width	60.30	64.10	65.50	65.908	66.90	72.30	2.145	4.602	0.621	0.891
Car height	47.80	52.00	54.10	53.725	55.50	59.80	2.444	5.971	-0.487	0.062
Curb weight	1488.00	2145.00	2414.00	2555.566	2935.00	4066.00	520.680	271107.874	-0.099	0.671
Engine size	61.00	97.00	120.00	126.907	141.00	326.00	41.643	1734.114	5.069	1.919
Stroke	2.07	3.11	3.29	3.255	3.41	4.17	0.314	0.098	2.043	-0.680
Compression ratio	7.00	8.60	9.00	10.143	9.40	23.00	3.972	15.777	4.999	2.573
Horse power	48.00	70.00	95.00	104.117	116.00	288.00	39.544	1563.741	2.535	1.385
Peak rpm	4150.00	4800.00	5200.00	5125.122	5500.00	6600.00	476.986	227515.304	0.026	0.074
City mpg	13.00	19.00	24.00	25.220	30.00	49.00	6.542	42.800	0.501	0.654
Highway mpg	16.00	25.00	30.00	30.751	34.00	54.00	6.886	47.423	0.367	0.532
Price	5118.00	7788.00	10295.00	13276.711	16503.00	45400.00	7988.852	63821761.572	2.891	1.752

Se puede observar el resumen de las medidas estadísticas de cada variable cuantitativa. Como ejemplo, explicaré el resumen de la variable “Wheelbase”, que tiene como valor mínimo 86.60, como valor máximo 120.90, y el promedio de medidas es de 98.757 entre todos los elementos de esta clase. Sin embargo, el 25% de

los valores son iguales o menores a 94.50, y el 75% de los valores son iguales o menores a 102.40. Además, se tiene una desviación estándar de 6.22 que indica que los valores están agrupados cerca de la media. Se puede observar que la curtosis es mayor que cero, por lo que es positiva y sugiere una distribución más puntiaguda. Por último, se tiene un sesgo positivo, lo que indica que es una distribución asimétrica a la derecha. Así, cada variable tiene su significado en el presente resumen para su análisis a lo largo de este entregable.

```
library(dplyr)

# Se crea un data frame con las variables categoricas
categorical_df <- M %>% select(symboling, CarName, fueltype, carbody, drivewheel, enginelocation, enginetype, cylindernumber)
head(categorical_df)
```

b) Cálculo de las medidas estadísticas de las variables cualitativas

symboling	CarName	fueltype	carbody	drivewheel	enginelocation	enginetype	cylindernumber
3	alfa-romero giulia	gas	convertible	rwd	front	dohc	four
3	alfa-romero stelvio	gas	convertible	rwd	front	dohc	four
1	alfa-romero Quadrifoglio	gas	hatchback	rwd	front	ohcv	six
2	audi 100 ls	gas	sedan	fwd	front	ohc	four
2	audi 100ls	gas	sedan	4wd	front	ohc	five
2	audi fox	gas	sedan	fwd	front	ohc	five

```
# Se crean los inputs
input1 <- categorical_df$symboling
input2 <- categorical_df$carbody

# Se crea un dataframe con los inputs
df_related <- data.frame(Input1 = input1, Input2 = input2)

# Se cuenta la frecuencia de los inputs
tabla_frecuencias <- df_related %>%
  group_by(Input1, Input2) %>%
  summarise(Frequency = n()) %>%
  ungroup()
```

1) Se generarán las tablas de frecuencias con dos inputs

`summarise()` has grouped output by 'Input1'. You can override using the
`.groups` argument.

```
# Se nombran las columnas
names <- c("Symboling", "Car body", "Frequency")
colnames(tabla_frecuencias) <- names

# Tabla de frecuencias
tabla_frecuencias
```

Symboling	Car body	Frequency
-2	sedan	3
-1	hatchback	2
-1	sedan	13

Symboling	Car body	Frequency
-1	wagon	7
0	hardtop	1
0	hatchback	8
0	sedan	43
0	wagon	15
1	hardtop	1
1	hatchback	27
1	sedan	23
1	wagon	3
2	convertible	1
2	hardtop	4
2	hatchback	13
2	sedan	14
3	convertible	5
3	hardtop	2
3	hatchback	20

Como se puede observar, hay valores que no tienen un patrón como la categoría de riesgo 2 y el tipo de coche convertible que solo se repite 1 vez. Pero, sí se puede encontrar un patrón en la categoría de riesgo “0”, y el tipo de coche “sedán” ya que se repite 43 veces, o por ejemplo la categoría de riesgo “1” y el tipo de coche “hatchback” que se repite 27 veces.

Esta tabla puede ayudar a tener un conocimiento de qué categoría de riesgo puede tener cada tipo de coche mayormente, ya que se sigue un patrón con los datos que tienen una frecuencia alta.

```
# Se crean los inputs
input3 <- categorical_df$symboling
input4 <- categorical_df$fueltype

# Se crea un dataframe con los inputs
df_related1 <- data.frame(Input1 = input3, Input2 = input4)

# Se cuenta la frecuencia de los inputs
tabla_frecuencias1 <- df_related1 %>%
  group_by(Input1, Input2) %>%
  summarise(Frequency = n()) %>%
  ungroup()

## `summarise()` has grouped output by 'Input1'. You can override using the
## `.groups` argument.

# Se nombran las columnas
names1 <- c("Symboling", "Fuel type", "Frequency")
colnames(tabla_frecuencias1) <- names1

# Tabla de frecuencias
print(tabla_frecuencias1)

## # A tibble: 10 x 3
##   Symboling `Fuel type` Frequency
##   <int> <chr>         <int>
## 1     -2 gas             3
## 2     -1 diesel          5
## 3     -1 gas            17
```

```
## 4      0 diesel      11
## 5      0 gas        56
## 6      1 diesel      1
## 7      1 gas        53
## 8      2 diesel      3
## 9      2 gas        29
## 10     3 gas        27
```

Se puede encontrar un patrón en la siguiente tabla de frecuencias ya que se obtiene que los vehículos con categoría de riesgo “0” suelen utilizar gas con una frecuencia de 56. Por otra parte, los vehículos de una categoría de riesgo “1” también suelen usar gas con una frecuencia de 53. Por lo que se puede asumir que los carros que por su categoría se asumen como seguros suelen utilizar gas. Pero esto no es del todo claro ya que la categoría 3 se asume como riesgosa y hay 27 autos de esa categoría que utilizan gas.

Si se ve de otro punto de vista, hay más frecuencia entre los autos que usan gas de los que usan diésel sin importar su categoría de riesgo.

```
# Se crean los inputs
input5 <- categorical_df$drivewheel
input6 <- categorical_df$cylindernumber

# Se crea un dataframe con los inputs
df_related2 <- data.frame(Input1 = input5, Input2 = input6)

# Se cuenta la frecuencia de los inputs
tabla_frecuencias2 <- df_related2 %>%
  group_by(Input1, Input2) %>%
  summarise(Frequency = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Input1'. You can override using the
## `.groups` argument.
```

```
# Se nombran las columnas
names2 <- c("Drive wheel", "Cylinder number", "Frequency")
colnames(tabla_frecuencias2) <- names2

# Tabla de frecuencias
print(tabla_frecuencias2)
```

```
## # A tibble: 12 x 3
##   `Drive wheel` `Cylinder number` Frequency
##   <chr>         <chr>             <int>
## 1 4wd          five              2
## 2 4wd          four              7
## 3 fwd         five              5
## 4 fwd         four             111
## 5 fwd         six              3
## 6 fwd         three             1
## 7 rwd         eight             5
## 8 rwd         five              4
## 9 rwd         four             41
## 10 rwd        six             21
## 11 rwd        twelve             1
## 12 rwd        two              4
```

Se puede observar que hay una frecuencia alta de 111 y se puede asumir que los autos con una rueda motriz

fwd suele utilizar cuatro cilindros.

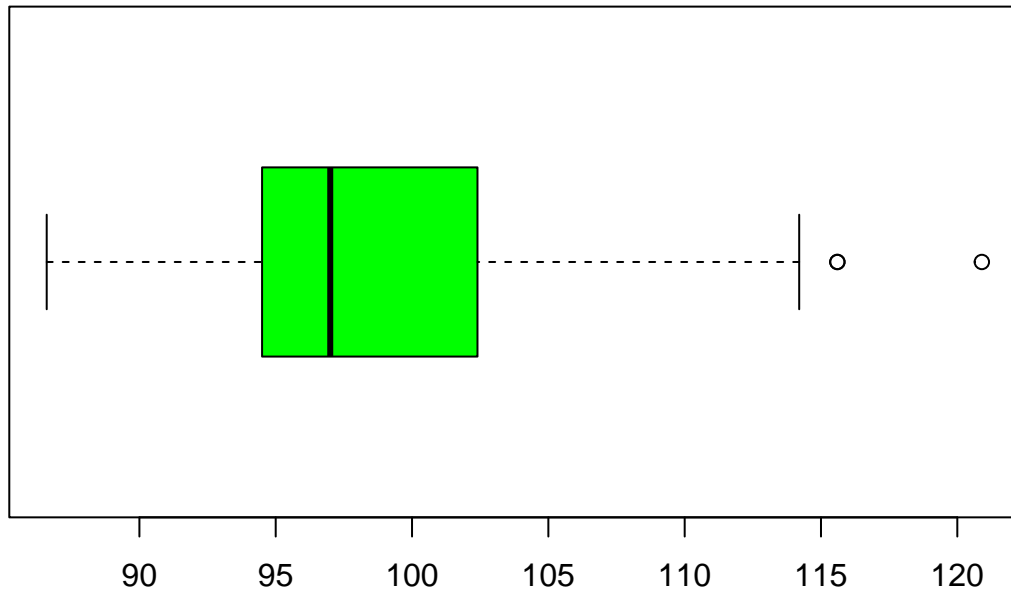
2. Exploración de los datos usando herramientas de visualización

a) Variables cuantitativas

```
#Boxplot para wheelbase
```

```
boxplot(numerical_df$wheelbase, horizontal = TRUE, col = "green", main = "Distribución de la distancia entre ejes de los automóviles")
```

Distribución de la distancia entre ejes de los automóviles



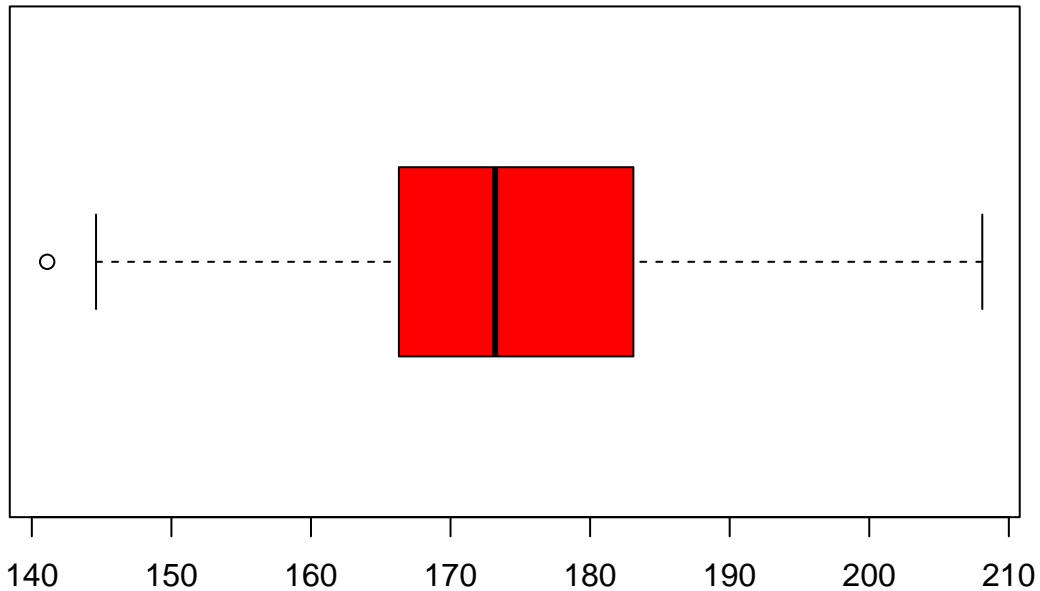
Boxplots

Se observa una distribución sesgada a la derecha con dos datos atípicos.

```
#Boxplot para car length
```

```
boxplot(numerical_df$carlength, horizontal = TRUE, col = "red", main = "Distribución de las longitudes de los automóviles")
```

Distribución de las longitudes de los automóviles

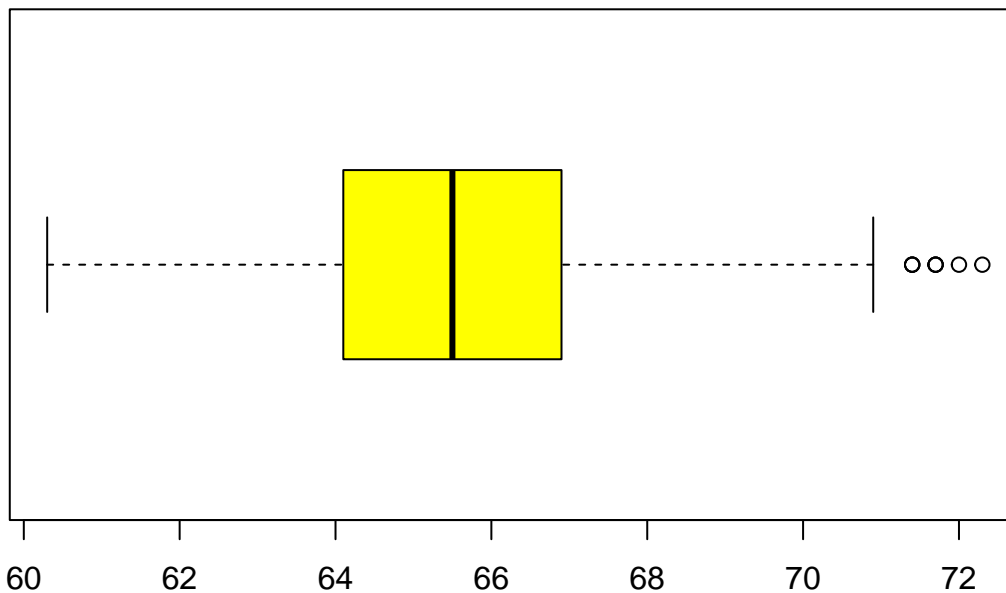


Se observa un solo dato atípico antes del mínimo. Además, una distribución sesgada a la derecha.

#Boxplot para car width

```
boxplot(numerical_df$carwidth, horizontal = TRUE, col = "yellow", main = "Distribución del ancho de los  
automóviles")
```

Distribución del ancho de los automóviles



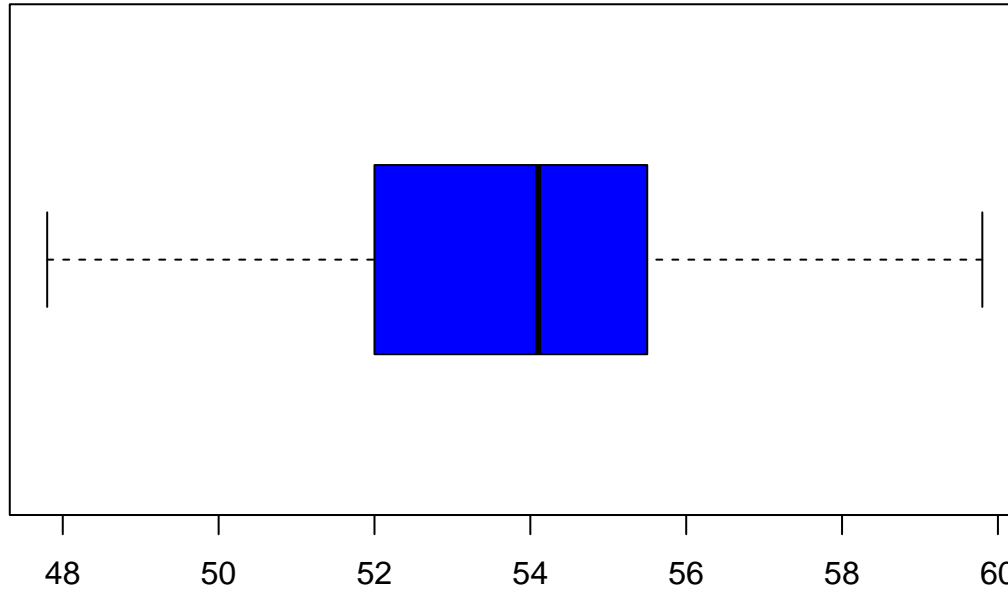
Se observa una distribución normalizada, pero teniendo en cuenta que hay datos atípicos entre 71 y 73.

#Boxplot para car height

```
boxplot(numerical_df$carheight, horizontal = TRUE, col = "blue", main = "Distribución de las alturas de los  
automóviles")
```

```
automóviles")
```

Distribución de las alturas de los automóviles

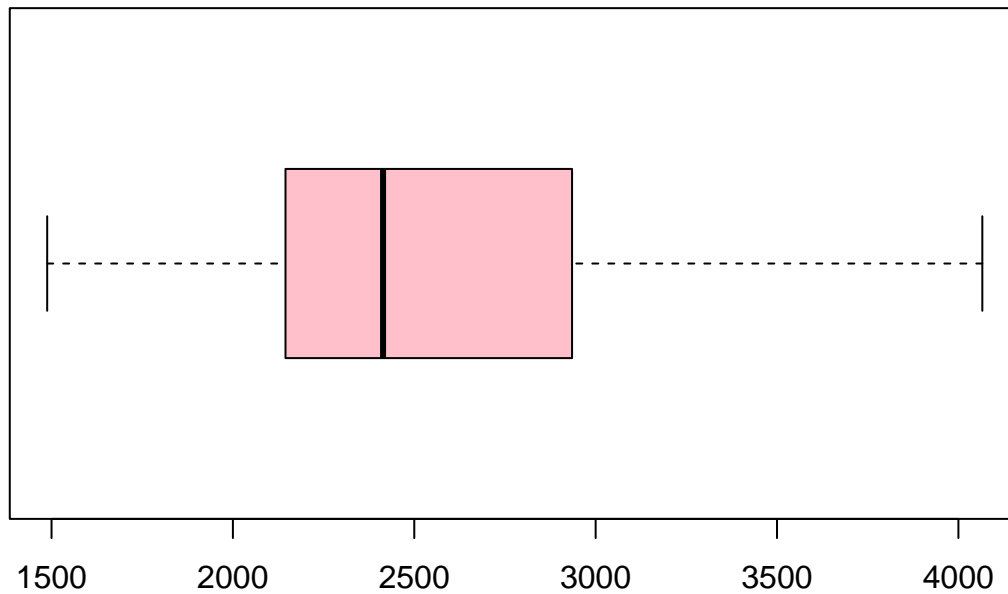


Se observa una distribución sesgada a la izquierda pero sin la presencia de datos atípicos.

```
#Boxplot para curb weight
```

```
boxplot(numerical_df$curbweight, horizontal = TRUE, col = "pink", main = "Distribución de los pesos en vac o de los  
autom viles")
```

Distribuci n de los pesos en vac o de los autom viles

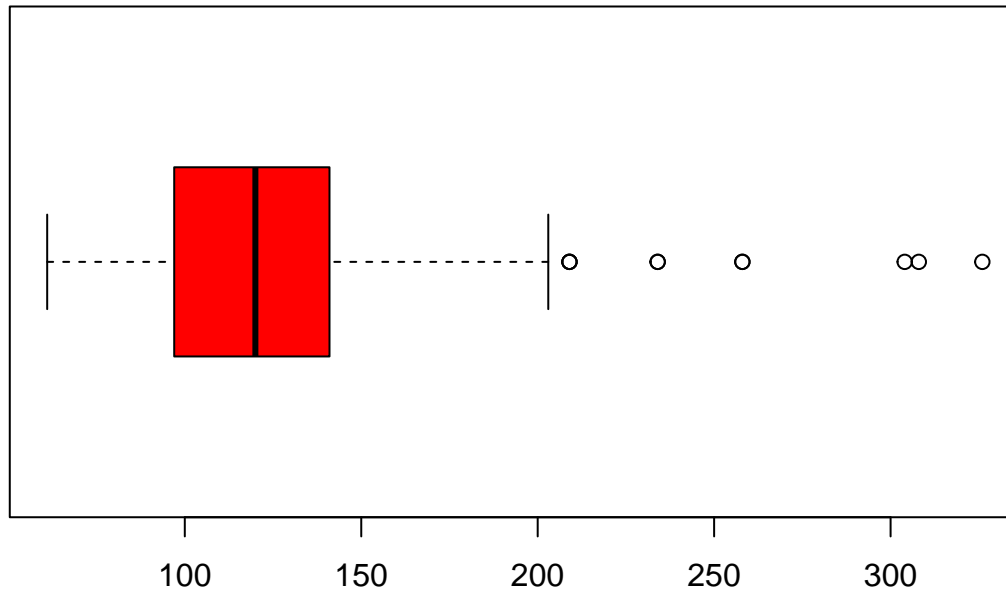


Se observa una distribuci n sesgada a la derecha pero sin la presencia de datos at picos.


```
#Boxplot para engine size
```

```
boxplot(numerical_df$enginesize, horizontal = TRUE, col = "red", main = "Distribución de los tamaños de  
automóviles")
```

Distribución de los tamaños de los motores para los automóviles

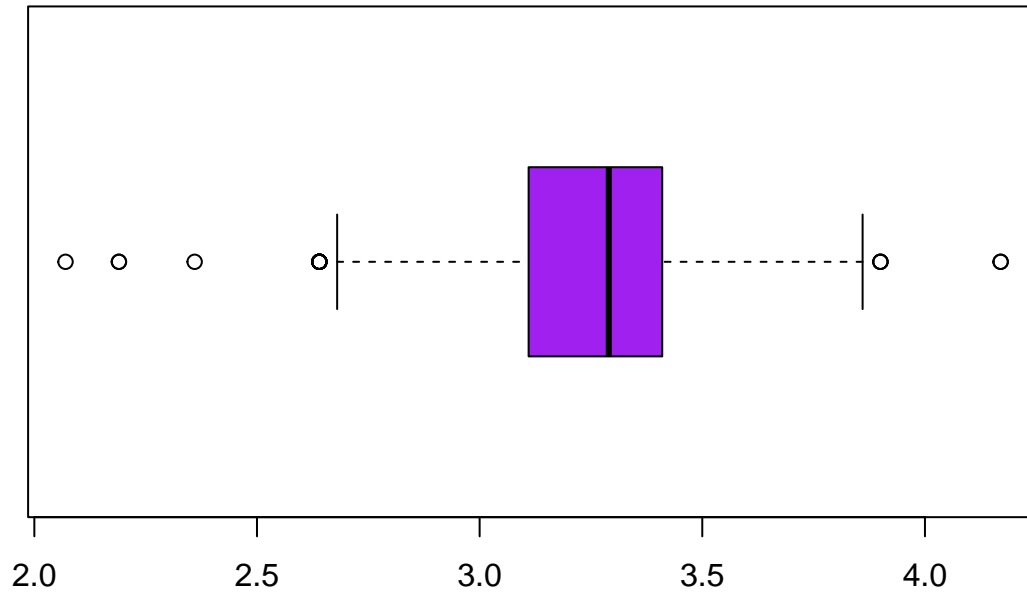


Se observa una gran cantidad de datos atípicos desde 225 aproximadamente, hasta más de 300. Sin embargo se ve una distribución normal.

```
#Boxplot para stroke
```

```
boxplot(numerical_df$stroke, horizontal = TRUE, col = "purple", main = "Distribución de tiempos de los  
automóviles")
```

Distribución de tiempos de los motores de los automóviles

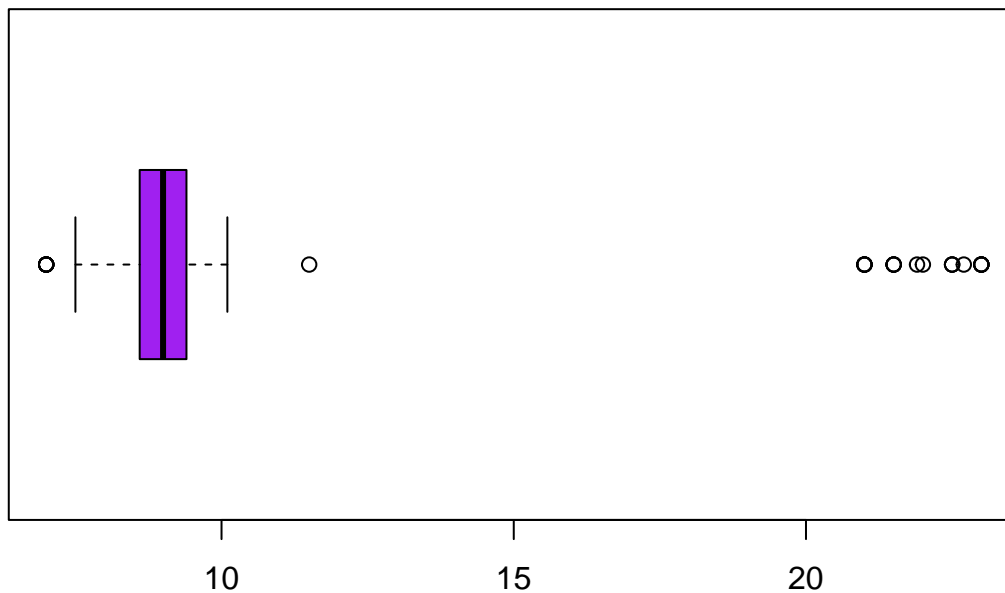


Se observa que hay valores atípicos entre 2 y 2.5, además de más de 4. Se observa una distribución asimétrica a la izquierda.

#Boxplot para compression ratio

```
boxplot(numerical_df$compressionratio, horizontal = TRUE, col = "purple", main = "Distribución de la relación de compresión de los automóviles")
```

Distribución de la relación de compresión de los automóviles

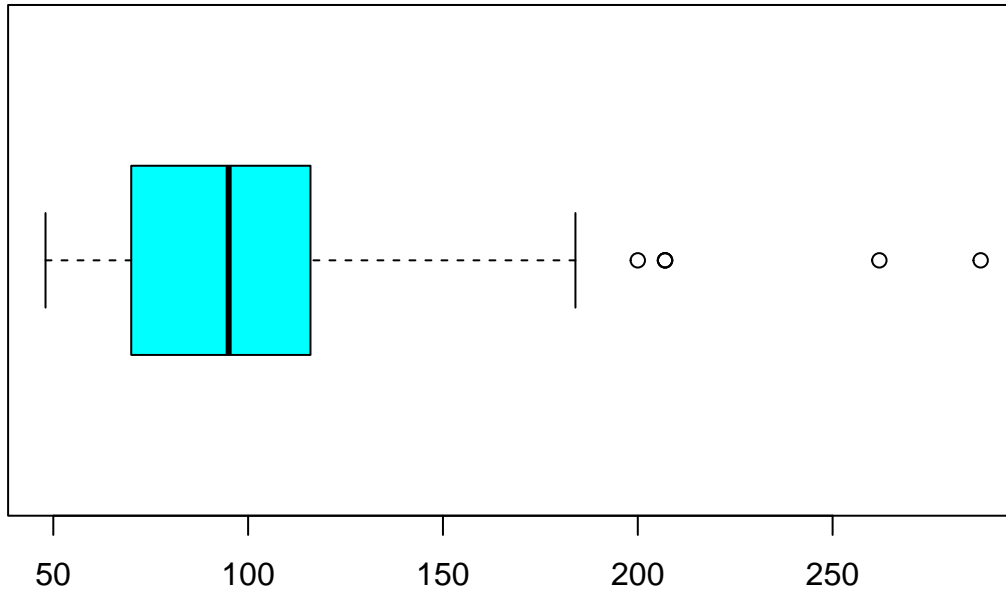


En este boxplot se observan valores atípicos mayores a 20. Por otra parte, hay una distribución normal ya que la media está centrada.

```
#Boxplot para horse power
```

```
boxplot(numerical_df$horsepower, horizontal = TRUE, col = "cyan", main = "Distribución de los caballos de potencia de los automóviles")
```

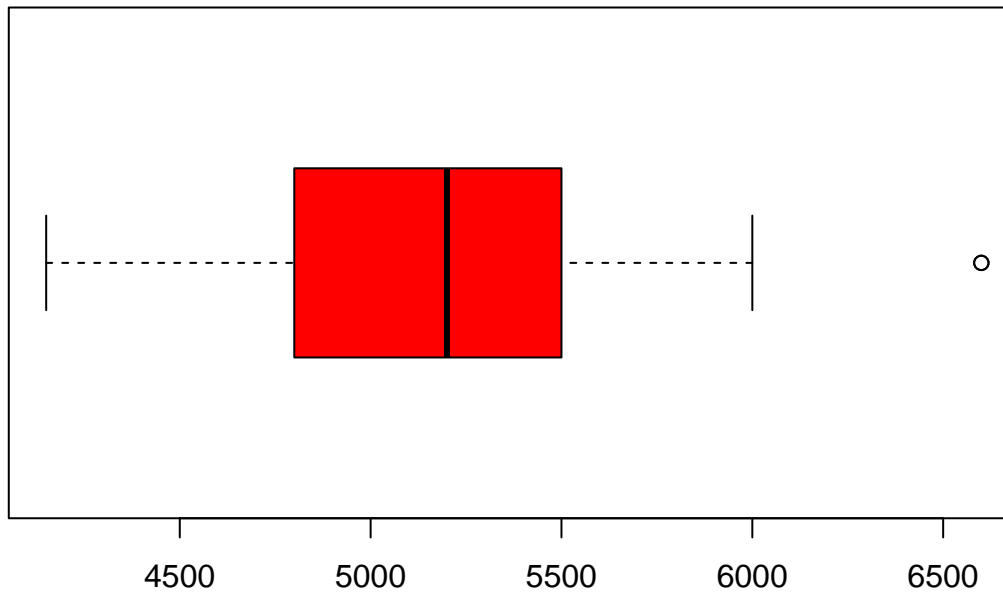
Distribución de los caballos de potencia de los automóviles



```
#Boxplot para peak rpm
```

```
boxplot(numerical_df$peakrpm, horizontal = TRUE, col = "red", main = "Distribución de RPM de los automóviles")
```

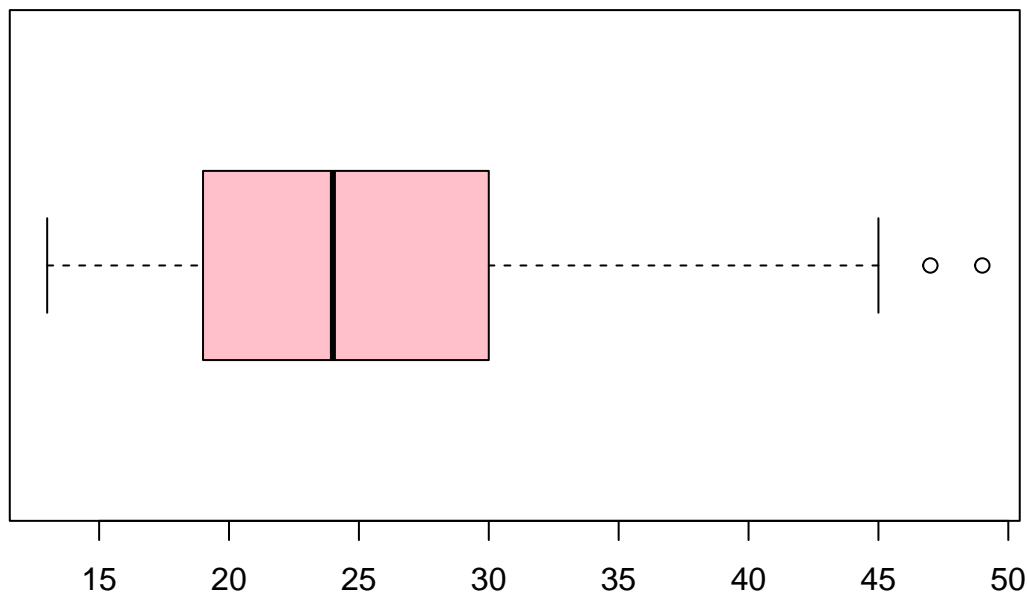
Distribución de RPM de los automóviles



```
#Boxplot para city mpg
```

```
boxplot(numerical_df$citympg, horizontal = TRUE, col = "pink", main = "Distribución de kilometraje en ciudad de los automóviles")
```

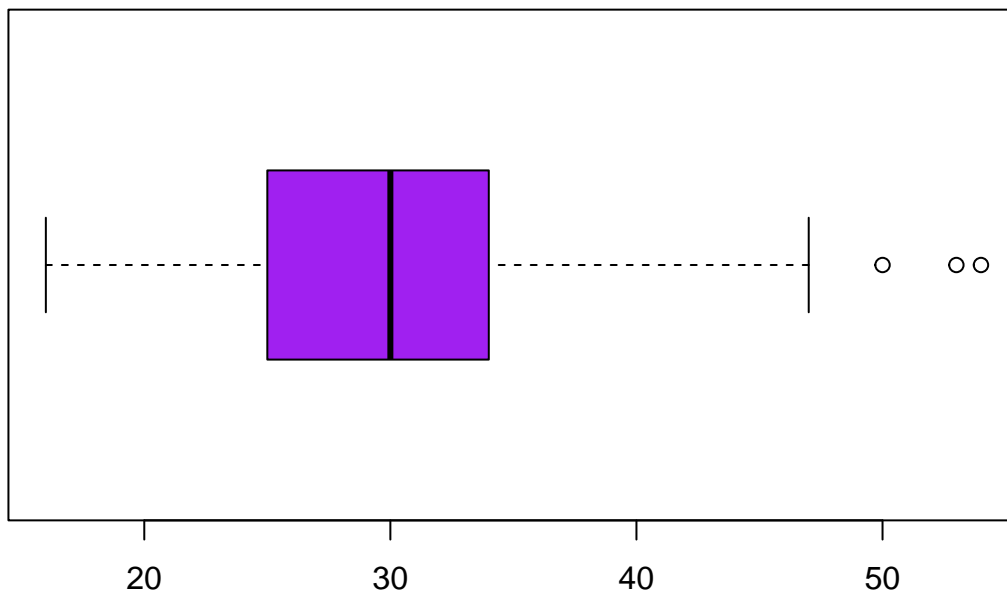
Distribución de kilometraje en ciudad de los automóviles



```
#Boxplot para highway mpg
```

```
boxplot(numerical_df$highwaympg, horizontal = TRUE, col = "purple", main = "Distribución de kilometraje
```

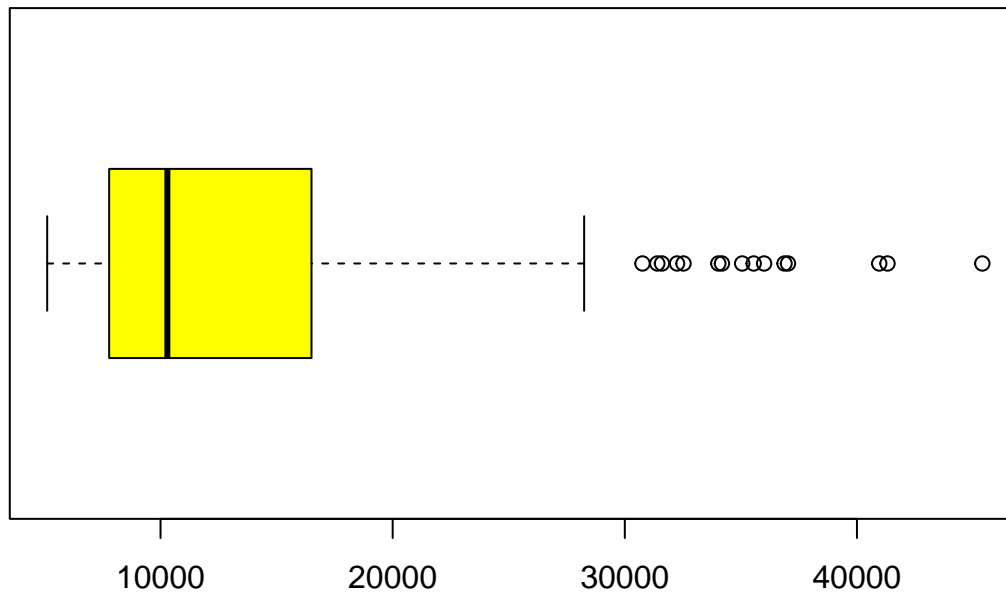
Distribución de kilometraje en autopista de los automóviles



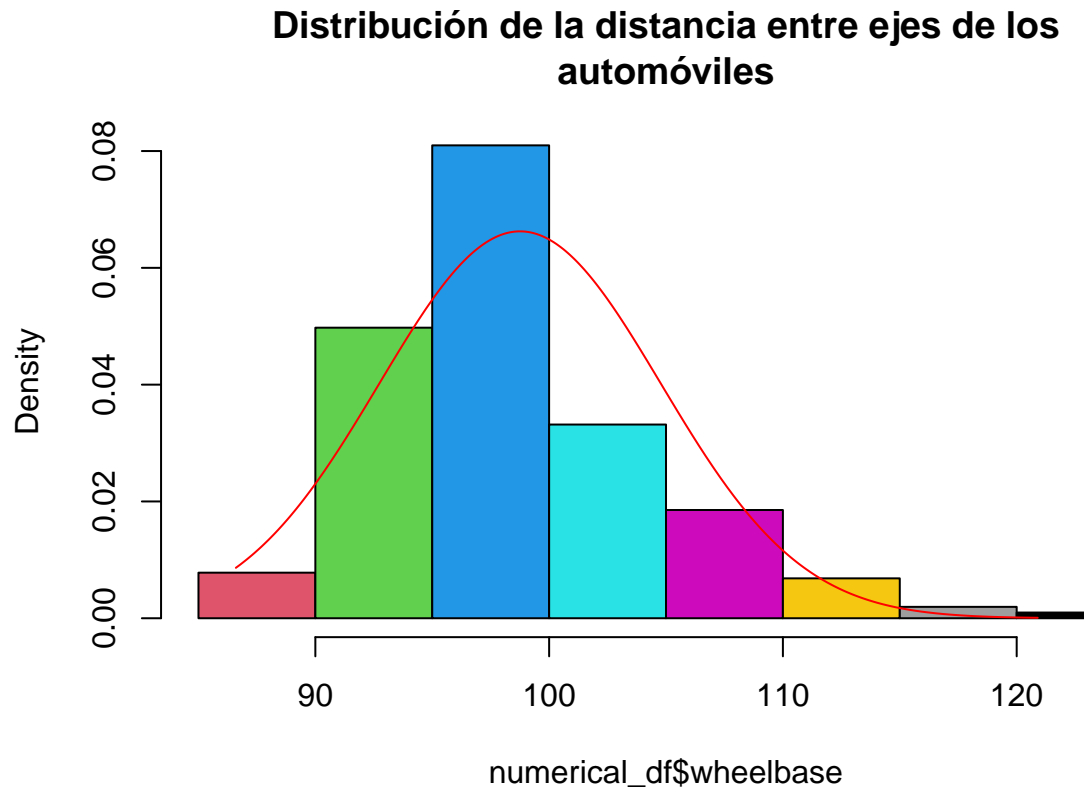
```
#Boxplot para price
```

```
boxplot(numerical_df$price, horizontal = TRUE, col = "yellow", main = "Distribución de precios de los a
```

Distribución de precios de los automóviles



```
#Histograma para wheelbase  
hist(numerical_df$wheelbase, prob = TRUE, col = 2:10, main = "Distribución de la distancia entre ejes d  
automóviles")  
x=seq(min(numerical_df$wheelbase),max(numerical_df$wheelbase),0.1)  
y=dnorm(x,mean(numerical_df$wheelbase),sd(numerical_df$wheelbase))  
lines(x,y,col="red")
```

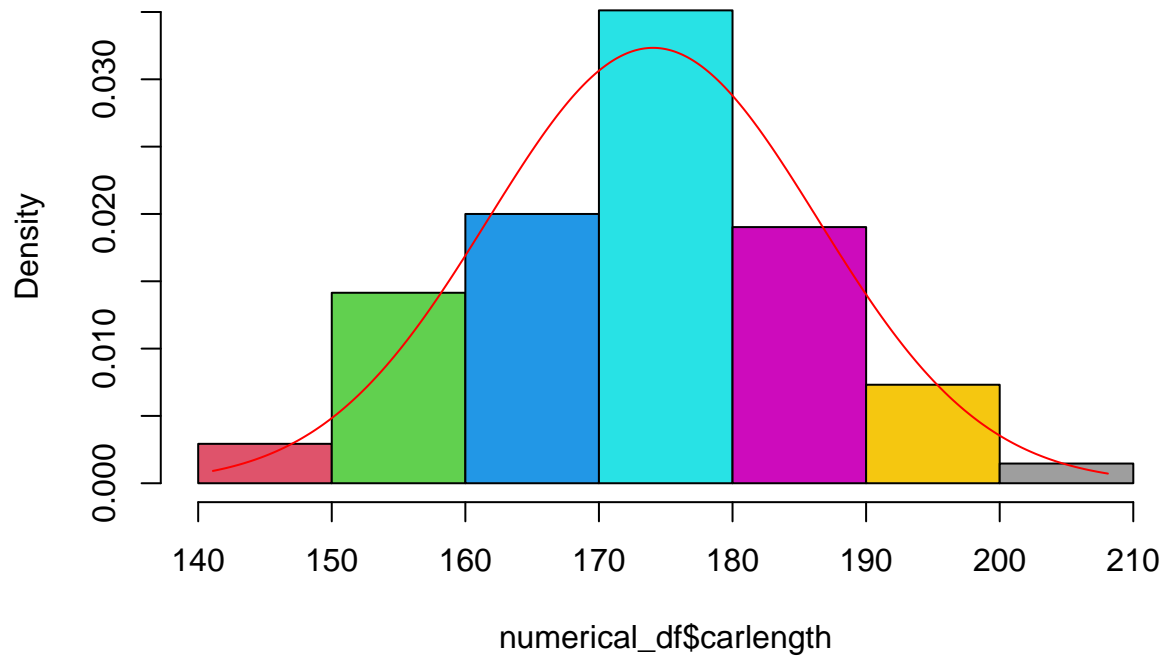


Histogramas y densidades

Se observa que hay una distribución relativamente normalizada ya que no hay una inclinación notoria en la curva.

```
#Histograma para car length
hist(numerical_df$carlength, prob = TRUE, col = 2:10, main = "Distribución de las longitudes de los
automóviles")
x1=seq(min(numerical_df$carlength),max(numerical_df$carlength),0.1)
y1=dnorm(x1,mean(numerical_df$carlength),sd(numerical_df$carlength))
lines(x1,y1,col="red")
```

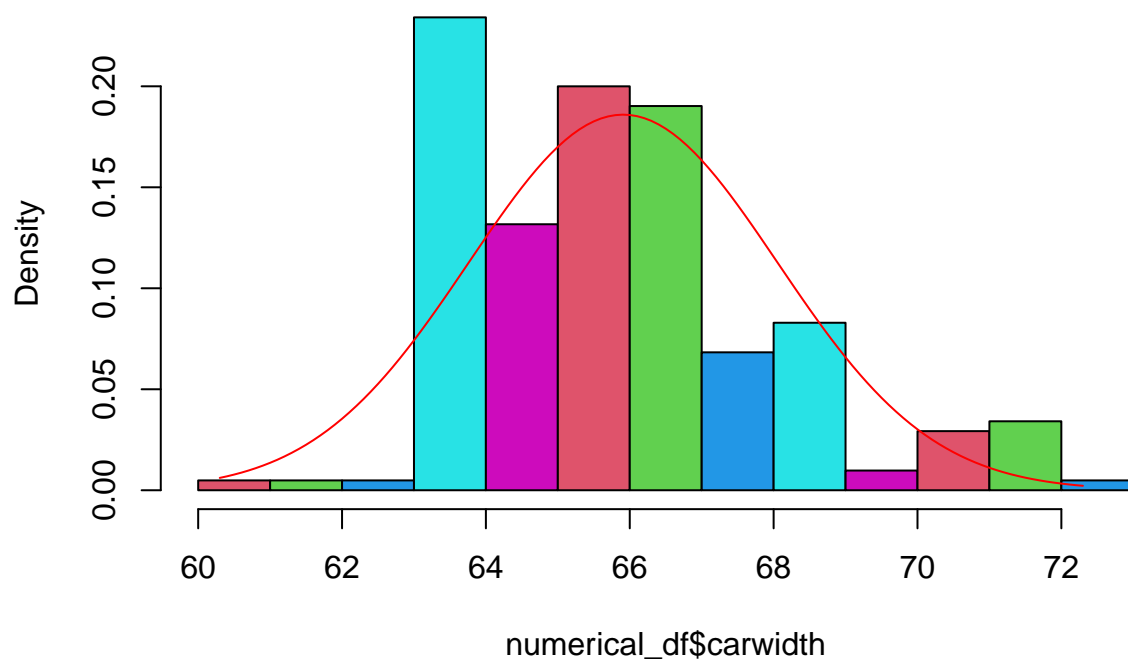
Distribución de las longitudes de los automóviles



Se observa que hay una distribución relativamente normalizada ya que no hay una inclinación notoria en la curva.

```
#Histograma para car width  
hist(numerical_df$carwidth, prob = TRUE, col = 2:6, main = "Distribución del ancho de los  
automóviles")  
x2=seq(min(numerical_df$carwidth),max(numerical_df$carwidth),0.1)  
y2=dnorm(x2,mean(numerical_df$carwidth),sd(numerical_df$carwidth))  
lines(x2,y2,col="red")
```

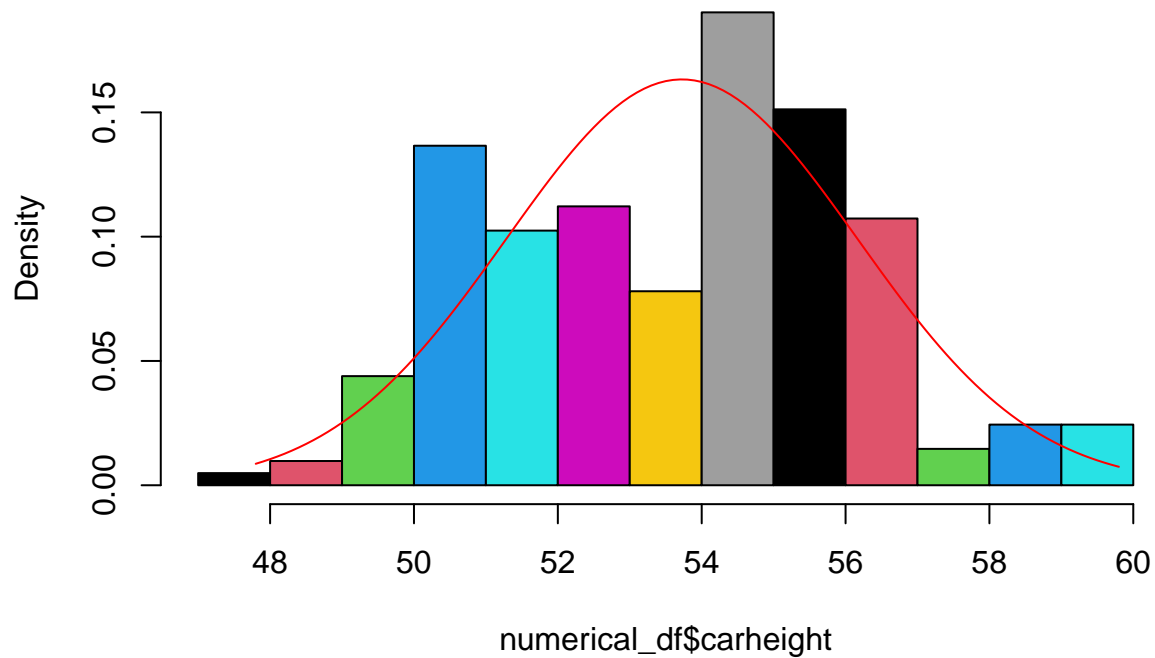
Distribución del ancho de los automóviles



Se observa que hay una distribución relativamente normalizada ya que no hay una inclinación notoria en la curva.

```
#Histograma para car height
hist(numerical_df$carheight, prob = TRUE, col = 1:8, main = "Distribución de las alturas de los
automóviles")
x3=seq(min(numerical_df$carheight),max(numerical_df$carheight),0.1)
y3=dnorm(x3,mean(numerical_df$carheight),sd(numerical_df$carheight))
lines(x3,y3,col="red")
```


Distribución de las alturas de los automóviles



Se observa que hay una distribución relativamente normalizada ya que no hay una inclinación notoria en la curva.

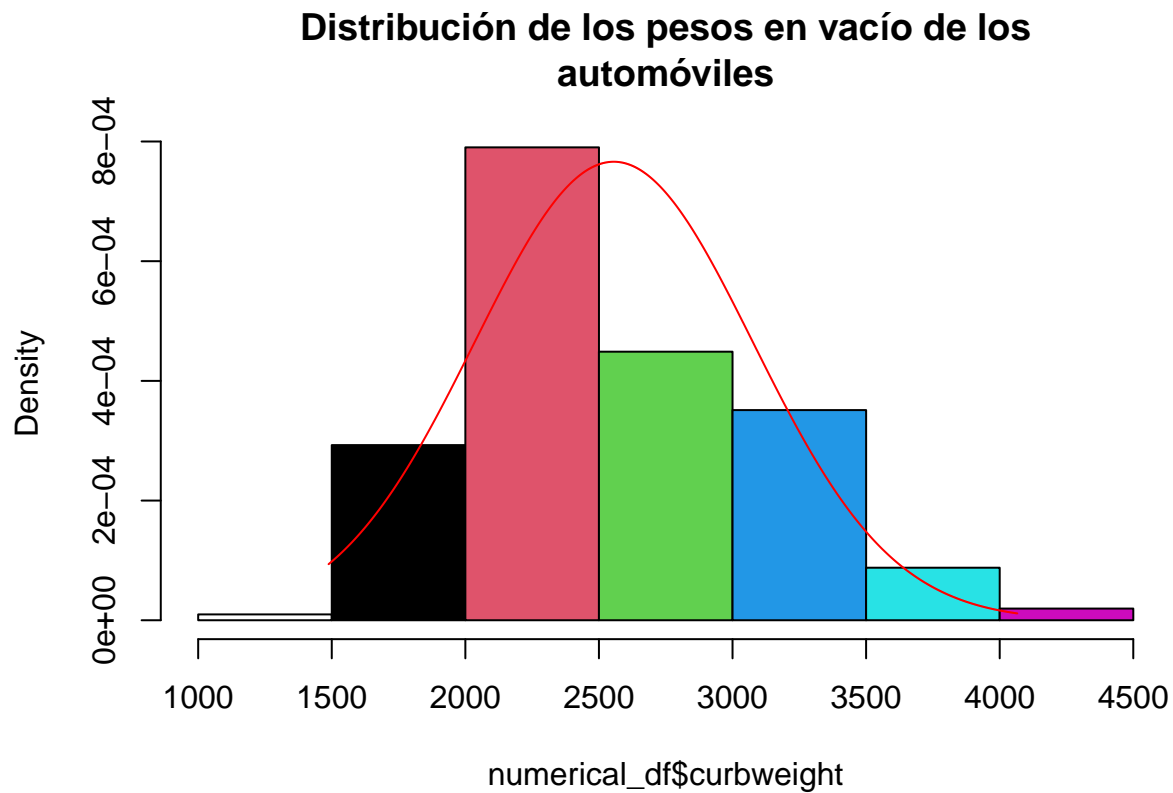
#Histograma para curb weight

```
hist(numerical_df$curbweight, prob = TRUE, col = 0:8, main = "Distribución de los pesos en vacío de los  
automóviles")
```

```
x4=seq(min(numerical_df$curbweight),max(numerical_df$curbweight),0.1)
```

```
y4=dnorm(x4,mean(numerical_df$curbweight),sd(numerical_df$curbweight))
```

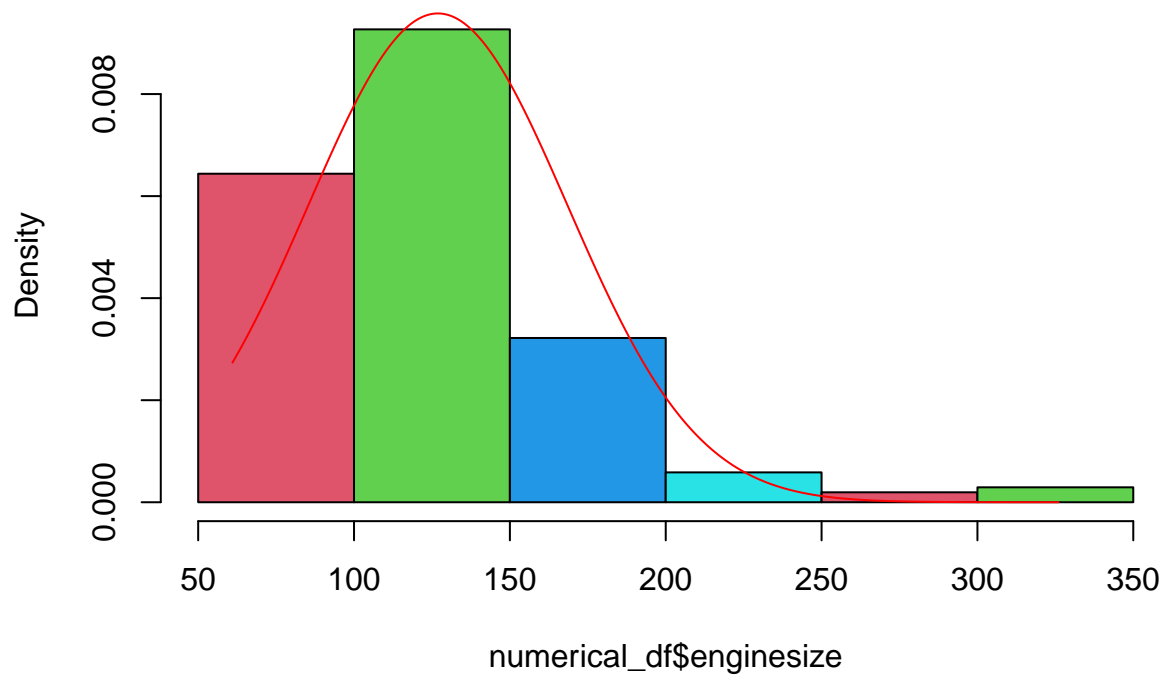
```
lines(x4,y4,col="red")
```



Se observa que hay una distribución relativamente normalizada ya que no hay una inclinación notoria en la curva.

```
#Histograma para engine size
hist(numerical_df$enginesize, prob = TRUE, col = 2:5, main = "Distribución de los tamaños de los motores de los automóviles")
x5=seq(min(numerical_df$enginesize),max(numerical_df$enginesize),0.1)
y5=dnorm(x5,mean(numerical_df$enginesize),sd(numerical_df$enginesize))
lines(x5,y5,col="red")
```

Distribución de los tamaños de los motores para los automóviles



Aquí hay una distribución sesgada a la derecha ya que el mayor volumen se concentra a la izquierda de la gráfica.

#Histograma para stroke

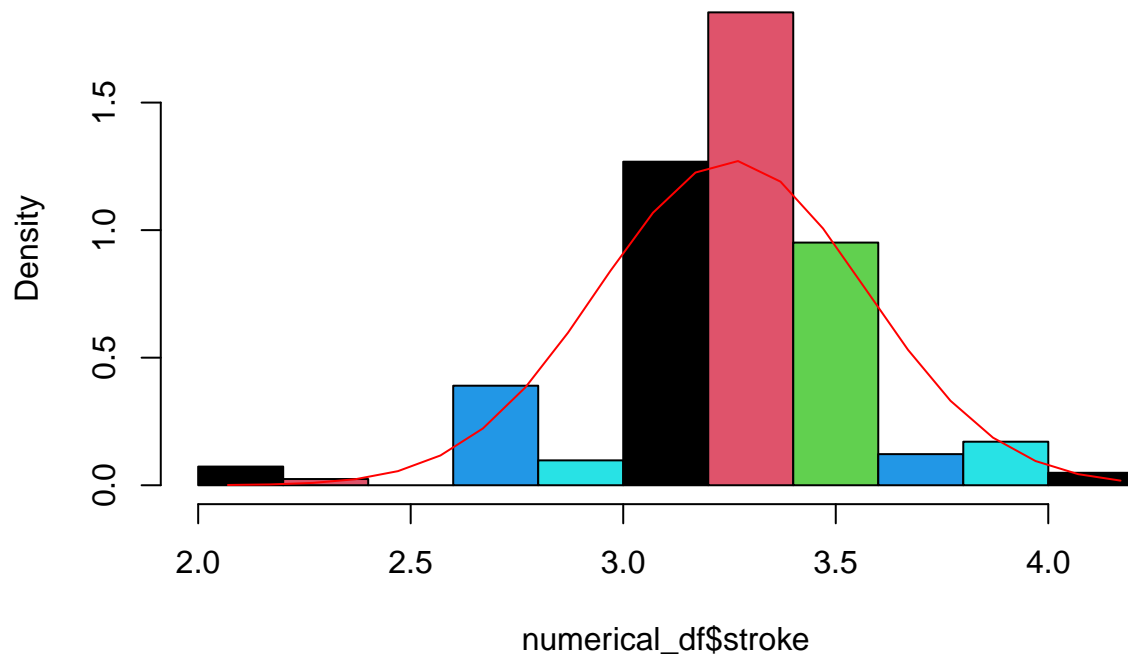
```
hist(numerical_df$stroke, prob = TRUE, col = 1:5, main = "Distribución de tiempos de los motores de los  
automóviles")
```

```
x6=seq(min(numerical_df$stroke),max(numerical_df$stroke),0.1)
```

```
y6=dnorm(x6,mean(numerical_df$stroke),sd(numerical_df$stroke))
```

```
lines(x6,y6,col="red")
```

Distribución de tiempos de los motores de los automóviles

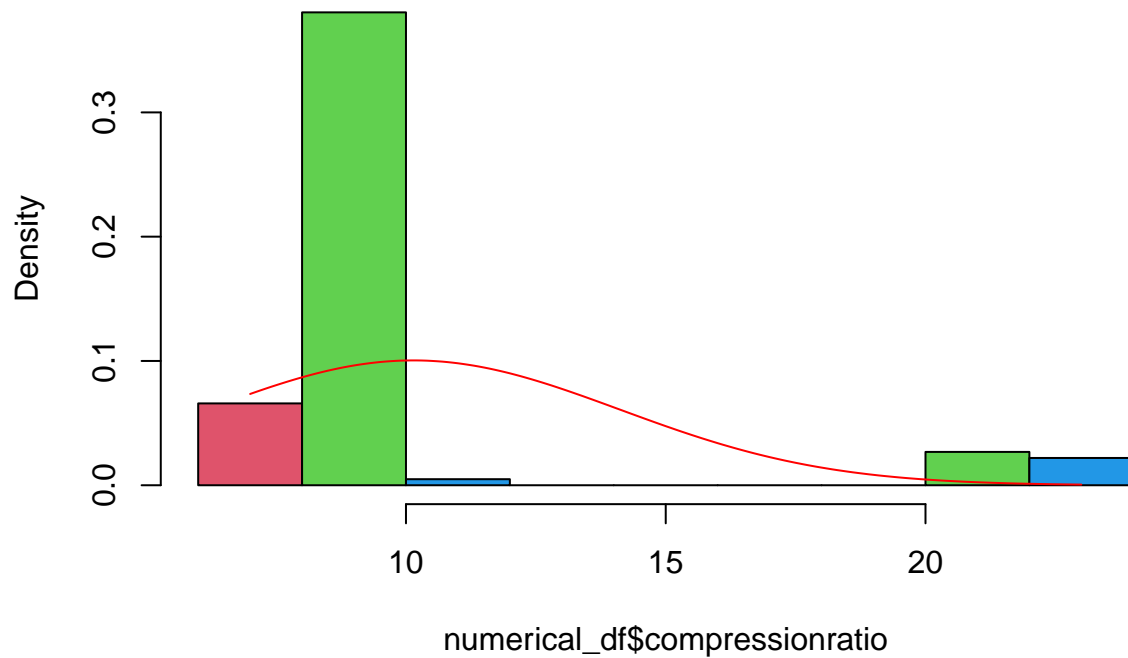


Por el contrario, en esta se observa una distribución sesgada a la izquierda ya que el volumen se concentra a la derecha de la gráfica.

```
#Histograma para compression ratio
hist(numerical_df$compressionratio, prob = TRUE, col = 10:15, main = "Distribución de la relación de compresión de los
automóviles")

x7=seq(min(numerical_df$compressionratio),max(numerical_df$compressionratio),0.1)
y7=dnorm(x7,mean(numerical_df$compressionratio),sd(numerical_df$compressionratio))
lines(x7,y7,col="red")
```

Distribución de la relación de compresión de los automóviles



En esta gráfica de relación de compresión de los automóviles está sesgada a la derecha pero se observa una gran cantidad de datos atípicos que pueden afectar el análisis.

```
#Histograma para horse power
```

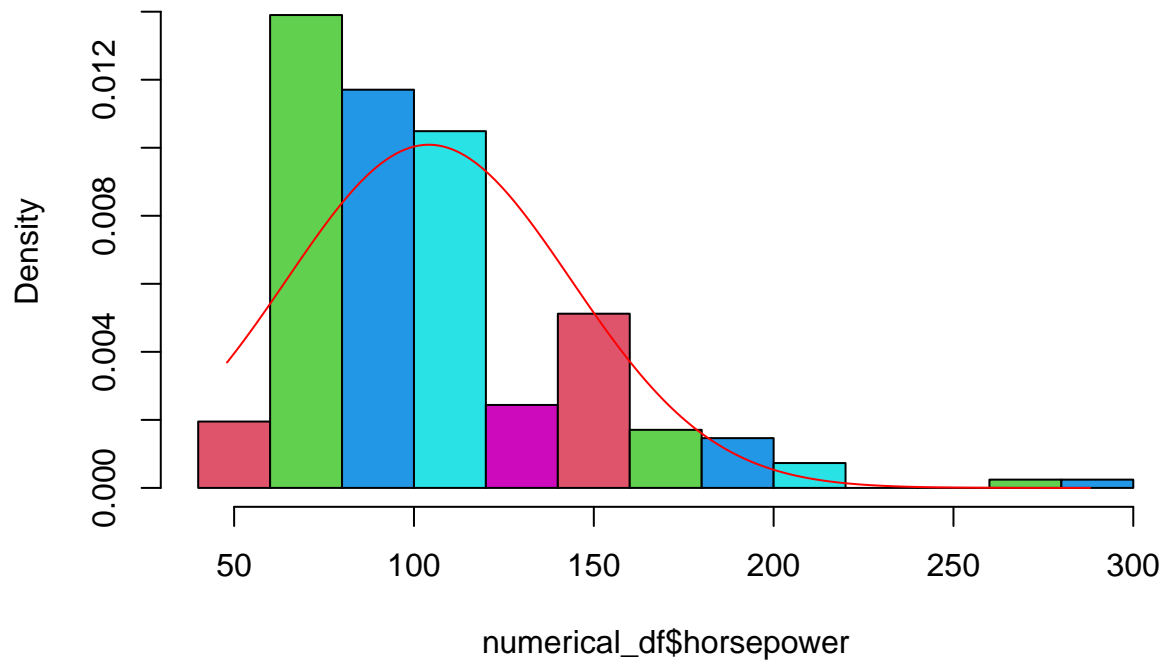
```
hist(numerical_df$horsepower, prob = TRUE, col = 10:14, main = "Distribución de los caballos de potencia")
```

```
x8=seq(min(numerical_df$horsepower),max(numerical_df$horsepower),0.1)
```

```
y8=dnorm(x8,mean(numerical_df$horsepower),sd(numerical_df$horsepower))
```

```
lines(x8,y8,col="red")
```

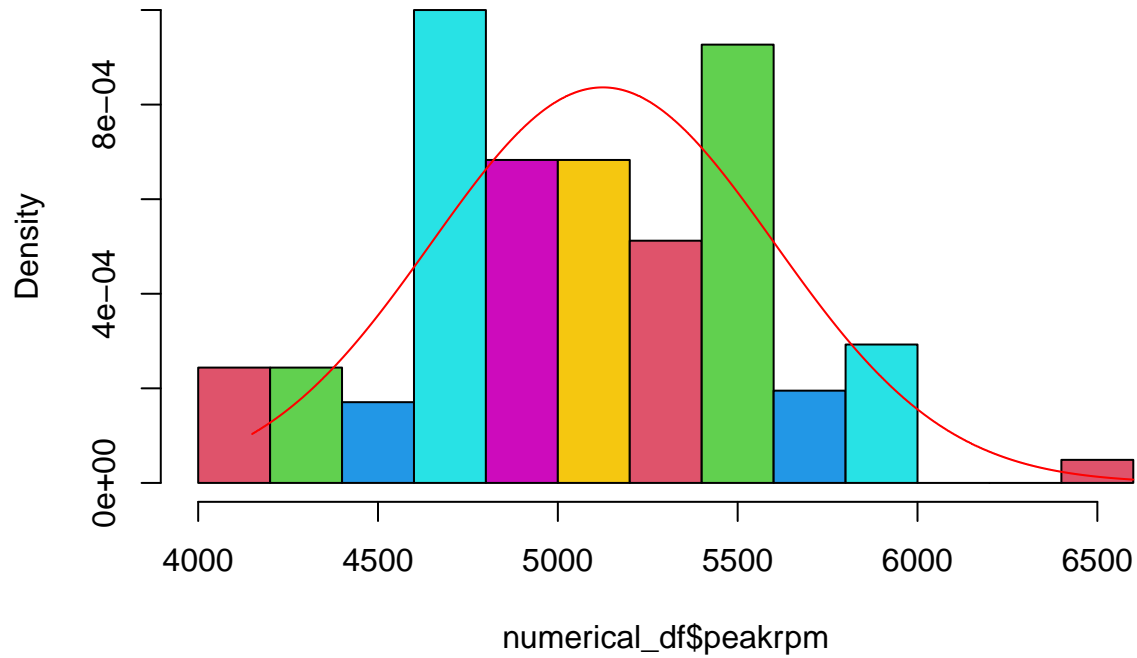
Distribución de los caballos de potencia de los automóviles



En esta gráfica de los caballos de potencia se observan datos atípicos entre 250 y 300 que generan ruido al análisis, pero se sabe que está sesgado a la derecha.

```
#Histograma para peak rpm
hist(numerical_df$peakrpm, prob = TRUE, col = 10:15, main = "Distribución de RPM de los automóviles")
x9=seq(min(numerical_df$peakrpm),max(numerical_df$peakrpm),0.1)
y9=dnorm(x9,mean(numerical_df$peakrpm),sd(numerical_df$peakrpm))
lines(x9,y9,col="red")
```

Distribución de RPM de los automóviles

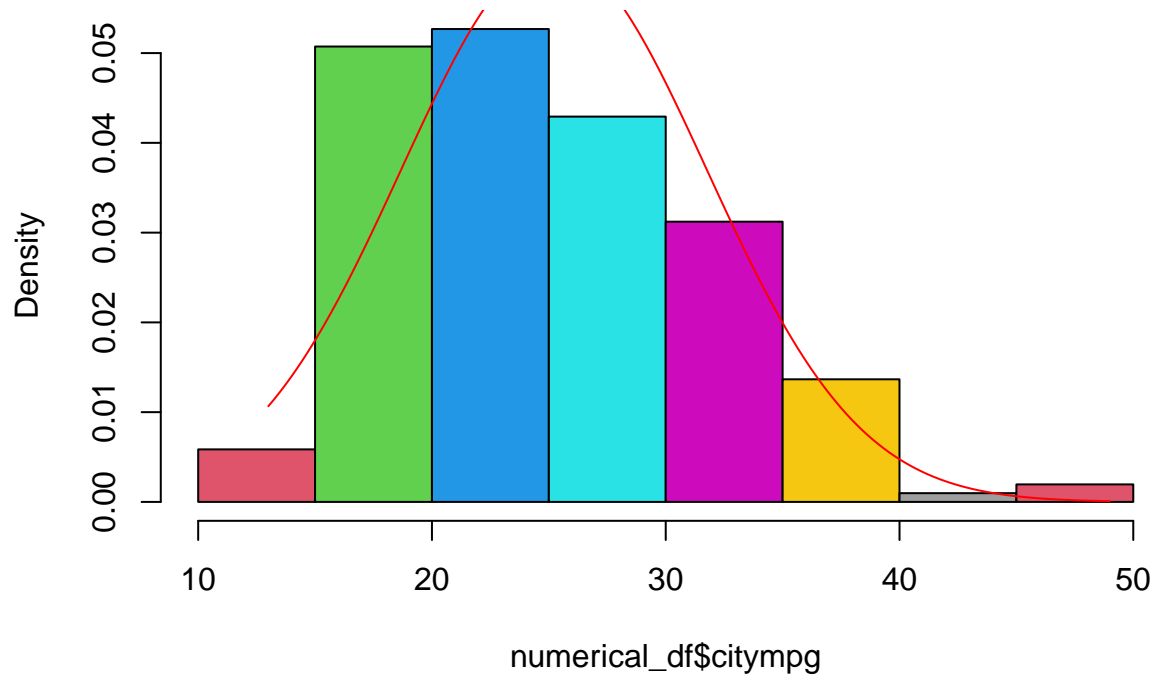


Hay pocos datos atípicos, pero se observa que la gráfica para los RPM está normalizada.

#Histograma para city mpg

```
hist(numerical_df$citympg, prob = TRUE, col = 2:8, main = "Distribución de kilometraje en ciudad de los  
x10=seq(min(numerical_df$citympg),max(numerical_df$citympg),0.1)  
y10=dnorm(x10,mean(numerical_df$citympg),sd(numerical_df$citympg))  
lines(x10,y10,col="red")
```

Distribución de kilometraje en ciudad de los automóviles

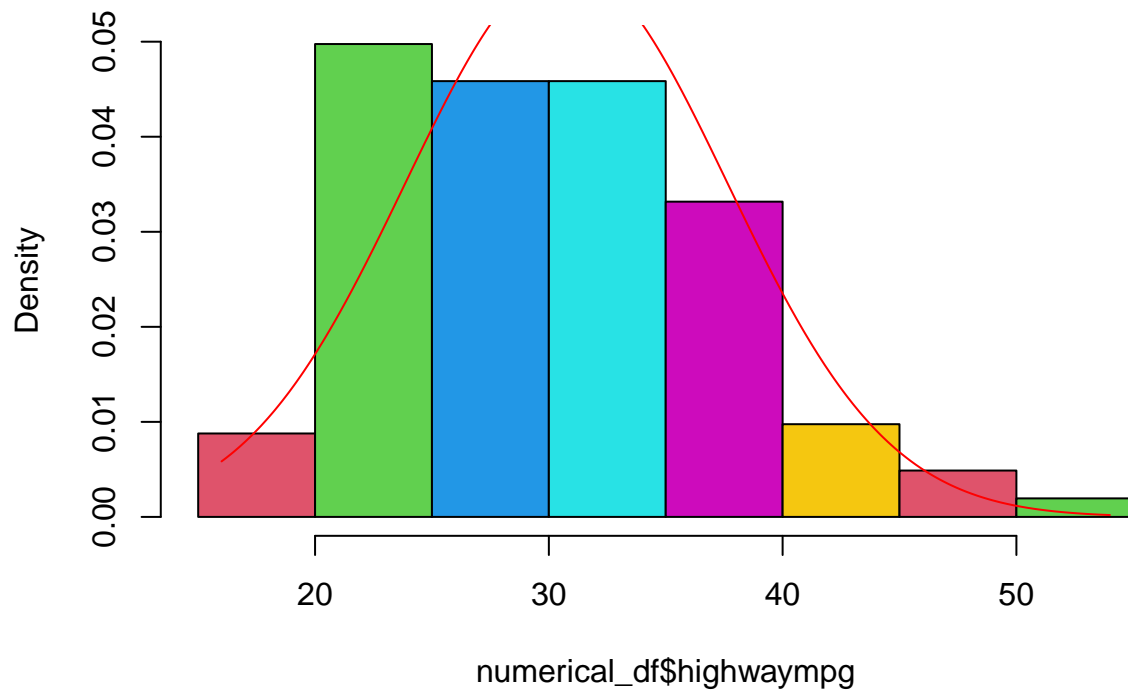


Se observa que está normalizada pero un poco influenciada por los datos atípicos.

#Histograma para highway mpg

```
hist(numerical_df$highwaympg, prob = TRUE, col = 10:15, main = "Distribución de kilometraje en autopista",  
x11=seq(min(numerical_df$highwaympg),max(numerical_df$highwaympg),0.1)  
y11=dnorm(x11,mean(numerical_df$highwaympg),sd(numerical_df$highwaympg))  
lines(x11,y11,col="red")
```

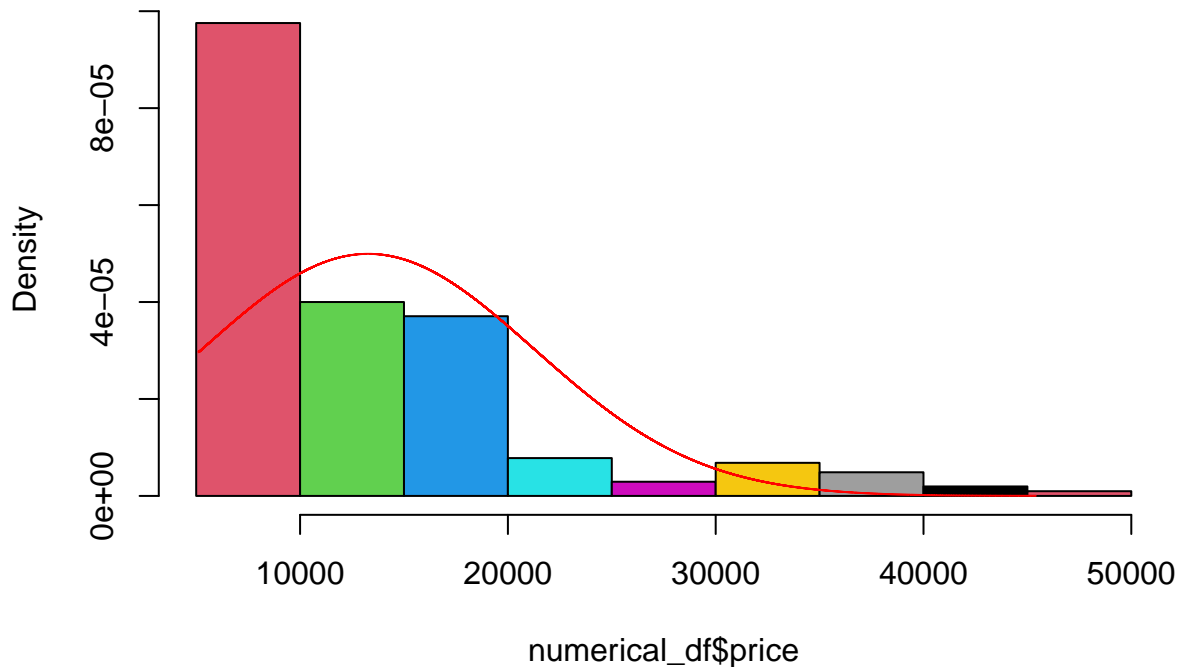

Distribución de kilometraje en autopista de los automóviles



Para los MPG del highway se observa normalizada.

```
#Histograma para price  
hist(numerical_df$price, prob = TRUE, col = 2:15, main = "Distribución de precios de los automóviles")  
x12=seq(min(numerical_df$price),max(numerical_df$price),0.1)  
y12=dnorm(x12,mean(numerical_df$price),sd(numerical_df$price))  
lines(x12,y12,col="red")
```

Distribución de precios de los automóviles



El precio se observa que su gráfica está sesgada a la derecha con varios valores atípicos.

Coefficiente de correlación El coeficiente de correlación puede ayudar a seleccionar las variables importantes para el análisis de las características de los automóviles que determinan su precio. Por lo tanto se evaluará la correlación de cada variable con respecto al precio en este inciso.

```
price <- numerical_df$price
wheelbase <- numerical_df$wheelbase
r= cor(price,wheelbase)
cat("El coeficiente de correlación entre el precio de los automóviles y la distancia entre ejes de los a
    r =",r)
```

```
## El coeficiente de correlación entre el precio de los automóviles y la distancia entre ejes de los au
##      r = 0.5778156
```

```
price <- numerical_df$price
carlength <- numerical_df$carlength
r1= cor(price,carlength)
cat("El coeficiente de correlación entre el precio de los automóviles y la longitud de los automóviles e
    r =",r1)
```

```
## El coeficiente de correlación entre el precio de los automóviles y la longitud de los automóviles es
##      r = 0.68292
```

```
price <- numerical_df$price
carwidth <- numerical_df$carwidth
r2= cor(price,carwidth)
cat("El coeficiente de correlación entre el precio de los automóviles y el ancho de los automóviles está
    r =",r2)
```

```
## El coeficiente de correlación entre el precio de los automóviles y el ancho de los automóviles está
##      r = 0.7593253
```

```

price <- numerical_df$price
carheight <- numerical_df$carheight
r3= cor(price,carheight)
cat("El coeficiente de correlación entre el precio de los automóviles y la altura de los automóviles es",
    r =",r3)

## El coeficiente de correlación entre el precio de los automóviles y la altura de los automóviles está
##     r = 0.1193362

price <- numerical_df$price
curbweight <- numerical_df$curbweight
r4= cor(price,curbweight)
cat("El coeficiente de correlación entre el precio de los automóviles y los pesos en vacío de los autom
    r =",r4)

## El coeficiente de correlación entre el precio de los automóviles y los pesos en vacío de los autom
##     r = 0.8353049

price <- numerical_df$price
enginesize <- numerical_df$enginesize
r5= cor(price,enginesize)
cat("El coeficiente de correlación entre el precio de los automóviles y el tamaño de los motores de los
    r =",r5)

## El coeficiente de correlación entre el precio de los automóviles y el tamaño de los motores de los a
##     r = 0.8741448

price <- numerical_df$price
stroke <- numerical_df$stroke
r6= cor(price,stroke)
cat("El coeficiente de correlación entre el precio de los automóviles y tiempos de los motores de los a
    r =",r6)

## El coeficiente de correlación entre el precio de los automóviles y tiempos de los motores de los aut
##     r = 0.07944308

price <- numerical_df$price
compressionratio <- numerical_df$compressionratio
r7= cor(price,compressionratio)
cat("El coeficiente de correlación entre el precio de los automóviles y la relación de compresión de los
    r =",r7)

## El coeficiente de correlación entre el precio de los automóviles y la relación de compresión de los a
##     r = 0.06798351

price <- numerical_df$price
horsepower <- numerical_df$horsepower
r8= cor(price,horsepower)
cat("El coeficiente de correlación entre el precio de los automóviles y los caballos de potencia de los
    r =",r8)

## El coeficiente de correlación entre el precio de los automóviles y los caballos de potencia de los a
##     r = 0.8081388

price <- numerical_df$price
peakrpm <- numerical_df$peakrpm
r9= cor(price,peakrpm)
cat("El coeficiente de correlación entre el precio de los automóviles y las RPM de los automóviles está

```

```

    r =",r9)

## El coeficiente de correlación entre el precio de los automóviles y las RPM de los automóviles está d
##      r = -0.08526715

price <- numerical_df$price
citympg <- numerical_df$citympg
r10= cor(price,citympg)
cat("El coeficiente de correlación entre el precio de los automóviles y los kilometrajes en ciudad de l
    r =",r10)

## El coeficiente de correlación entre el precio de los automóviles y los kilometrajes en ciudad de los
##      r = -0.6857513

price <- numerical_df$price
highwaympg <- numerical_df$highwaympg
r11= cor(price,highwaympg)
cat("El coeficiente de correlación entre el precio de los automóviles y los kilometrajes en carretera d
    r =",r11)

## El coeficiente de correlación entre el precio de los automóviles y los kilometrajes en carretera de
##      r = -0.6975991

#Coeficientes de correlación entre todas las combinaciones posibles de variables numericas

data <- numerical_df

# Calcular la matriz de correlación
cor_matrix <- cor(data[, c("wheelbase", "carlength", "carwidth", "carheight", "curbweight", "enginesize

print(cor_matrix)

##           wheelbase  carlength  carwidth  carheight  curbweight
## wheelbase      1.0000000  0.8745875  0.7951436  0.58943476  0.7763863
## carlength      0.8745875  1.0000000  0.8411183  0.49102946  0.8777285
## carwidth       0.7951436  0.8411183  1.0000000  0.27921032  0.8670325
## carheight      0.5894348  0.4910295  0.2792103  1.00000000  0.2955717
## curbweight     0.7763863  0.8777285  0.8670325  0.29557173  1.0000000
## enginesize      0.5693287  0.6833599  0.7354334  0.06714874  0.8505941
## stroke         0.1609590  0.1295326  0.1829417 -0.05530667  0.1687900
## compressionratio 0.2497858  0.1584137  0.1811286  0.26121423  0.1513617
## horsepower     0.3532945  0.5526230  0.6407321 -0.10880206  0.7507393
## peakrpm        -0.3604687 -0.2872422 -0.2200123 -0.32041072 -0.2662432
## citympg         -0.4704136 -0.6709087 -0.6427043 -0.04863963 -0.7574138
## highwaympg      -0.5440819 -0.7046616 -0.6772179 -0.10735763 -0.7974648
## price          0.5778156  0.6829200  0.7593253  0.11933623  0.8353049
##           enginesize      stroke  compressionratio  horsepower
## wheelbase      0.56932868  0.16095905           0.24978585  0.35329448
## carlength      0.68335987  0.12953261           0.15841371  0.55262297
## carwidth       0.73543340  0.18294169           0.18112863  0.64073208
## carheight      0.06714874 -0.05530667           0.26121423 -0.10880206
## curbweight     0.85059407  0.16879004           0.15136174  0.75073925
## enginesize      1.00000000  0.20312859           0.02897136  0.80976865
## stroke         0.20312859  1.00000000           0.18611011  0.08093954
## compressionratio 0.02897136  0.18611011           1.00000000 -0.20432623

```

```
## horsepower      0.80976865  0.08093954      -0.20432623  1.00000000
## peakrpm         -0.24465983 -0.06796375      -0.43574051  0.13107251
## citympg         -0.65365792 -0.04214475       0.32470142 -0.80145618
## highwaympg      -0.67746991 -0.04393093       0.26520139 -0.77054389
## price           0.87414480  0.07944308       0.06798351  0.80813882
##                peakrpm    citympg    highwaympg    price
## wheelbase      -0.36046875 -0.47041361 -0.54408192  0.57781560
## carlength      -0.28724220 -0.67090866 -0.70466160  0.68292002
## carwidth       -0.22001230 -0.64270434 -0.67721792  0.75932530
## carheight      -0.32041072 -0.04863963 -0.10735763  0.11933623
## curbweight     -0.26624318 -0.75741378 -0.79746479  0.83530488
## enginesize     -0.24465983 -0.65365792 -0.67746991  0.87414480
## stroke         -0.06796375 -0.04214475 -0.04393093  0.07944308
## compressionratio -0.43574051  0.32470142  0.26520139  0.06798351
## horsepower     0.13107251 -0.80145618 -0.77054389  0.80813882
## peakrpm        1.00000000 -0.11354438 -0.05427481 -0.08526715
## citympg        -0.11354438  1.00000000  0.97133704 -0.68575134
## highwaympg     -0.05427481  0.97133704  1.00000000 -0.69759909
## price         -0.08526715 -0.68575134 -0.69759909  1.00000000
```

Se puede observar que price tiene una correlación positiva y mayor que cero con: curbweight, horsepower, carwidth, enginesize, lo que quiere decir que cuando estas variables aumentan, entonces muy probablemente el precio aumentará. Sin embargo, tiene una correlación negativa con: city mpg, y highwaympg, lo que quiere decir que cuando estas aumenten (en kilometraje), el precio del vehículo tenderá a disminuir.

```
#nuevo data frame
data <- numerical_df
#Variables
y_variable <- "price"
x_variables <- colnames(data)[colnames(data) != y_variable]

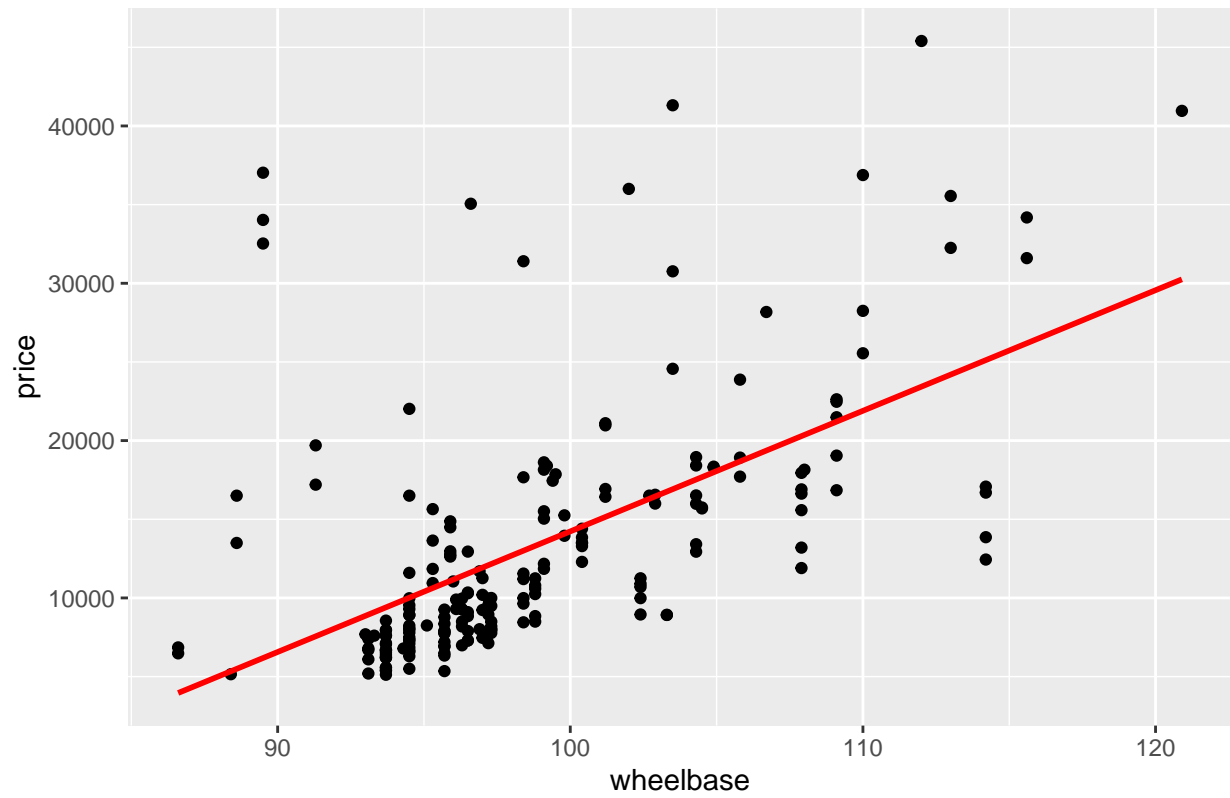
# Diagramas de dispersion
for (x_var in x_variables) {
  p <- ggplot(data, aes(x = !!sym(x_var), y = !!sym(y_variable))) +
    geom_point() +
    geom_smooth(method = "lm", col = "red", se = FALSE) + # Agregar línea de regresión
    labs(title = paste("Scatter plot of", x_var, "vs", y_variable),
         x = x_var,
         y = y_variable)

  print(p)
}
```

Diagrama de dispersión

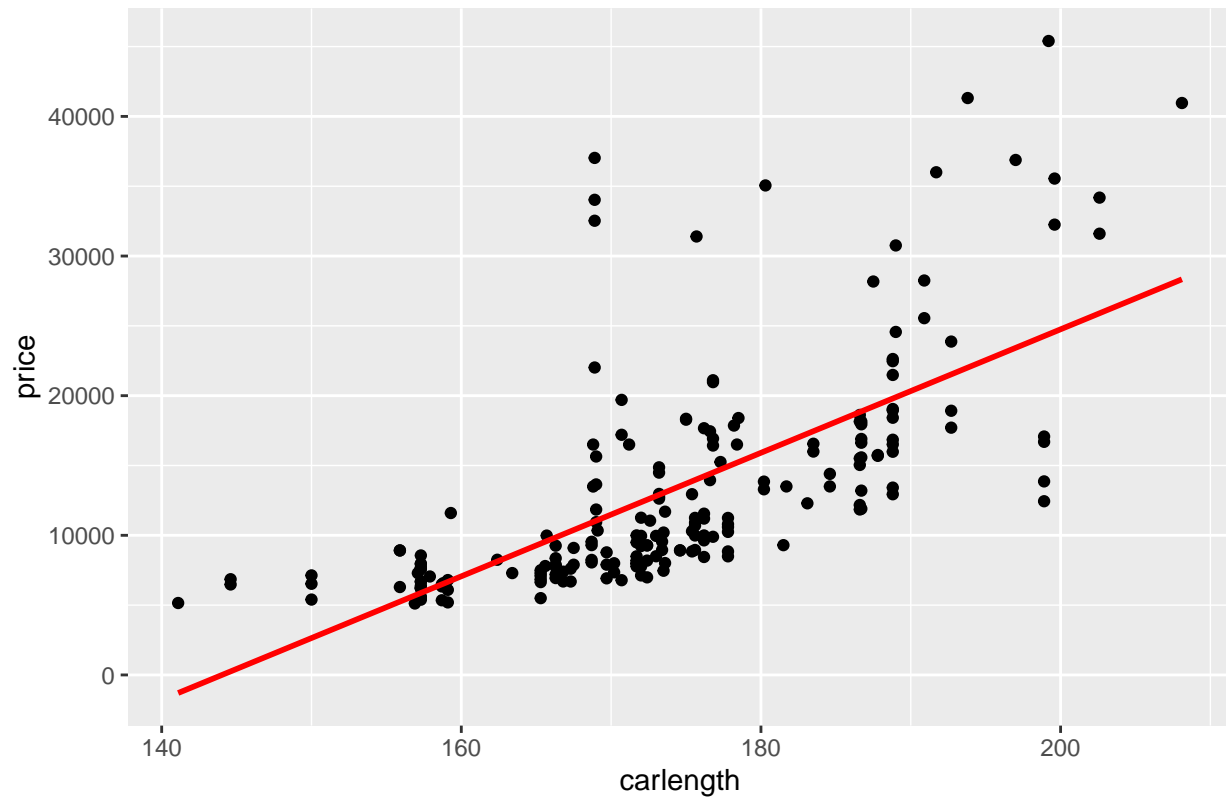
```
## `geom_smooth()` using formula = 'y ~ x'
```

Scatter plot of wheelbase vs price



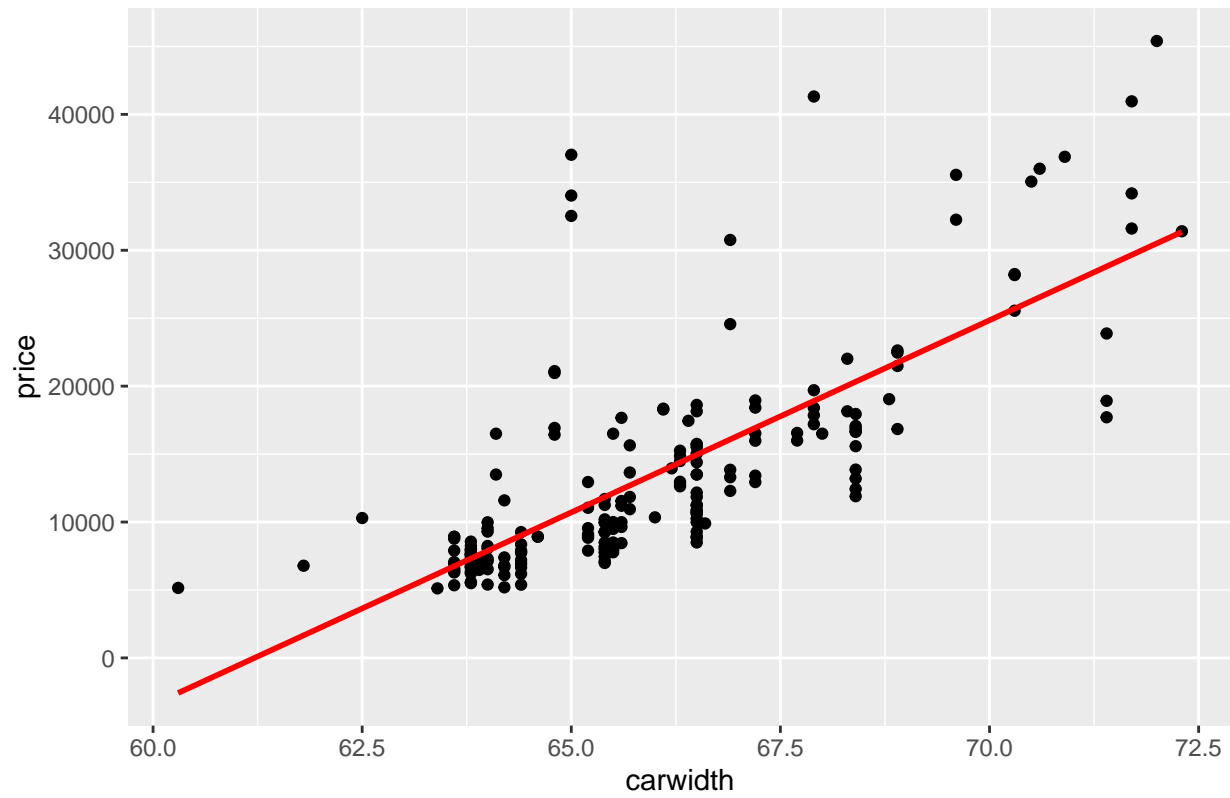
```
## `geom_smooth()` using formula = 'y ~ x'
```

Scatter plot of carlength vs price



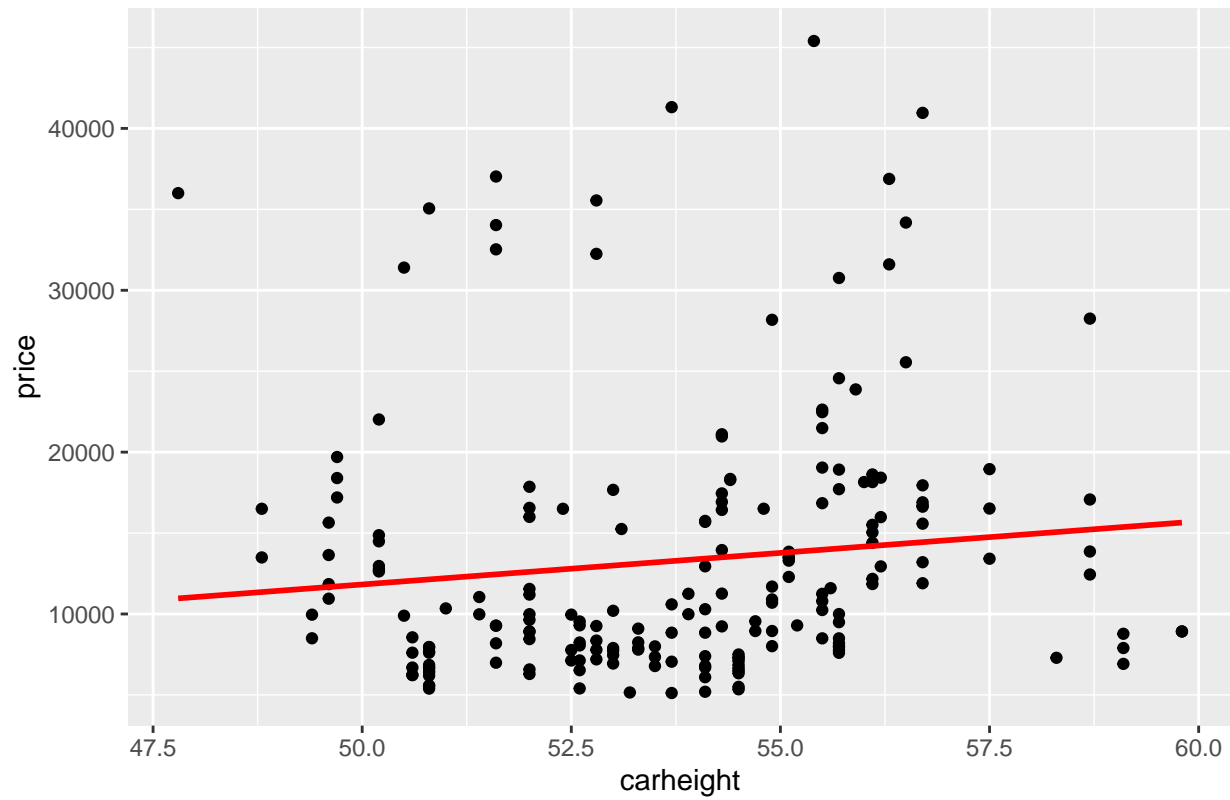
```
## `geom_smooth()` using formula = 'y ~ x'
```

Scatter plot of carwidth vs price



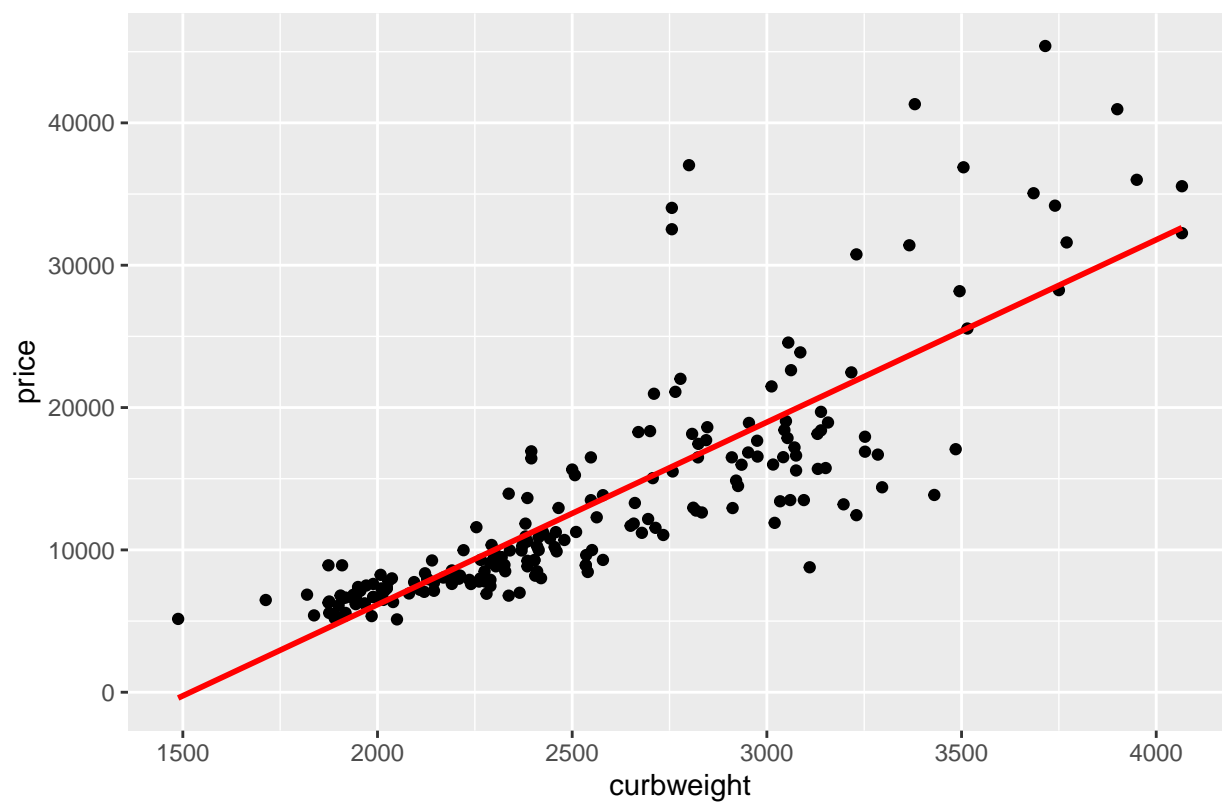
```
## `geom_smooth()` using formula = 'y ~ x'
```


Scatter plot of carheight vs price



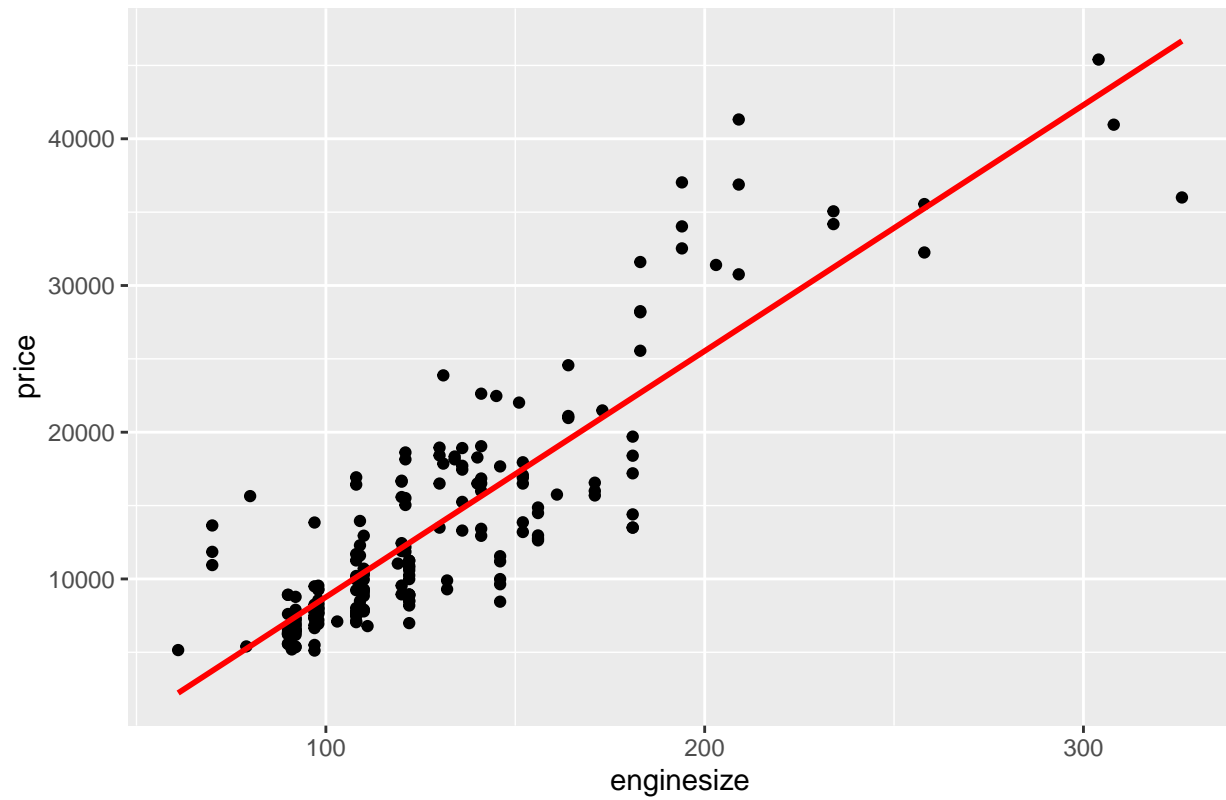
```
## `geom_smooth()` using formula = 'y ~ x'
```

Scatter plot of curbweight vs price



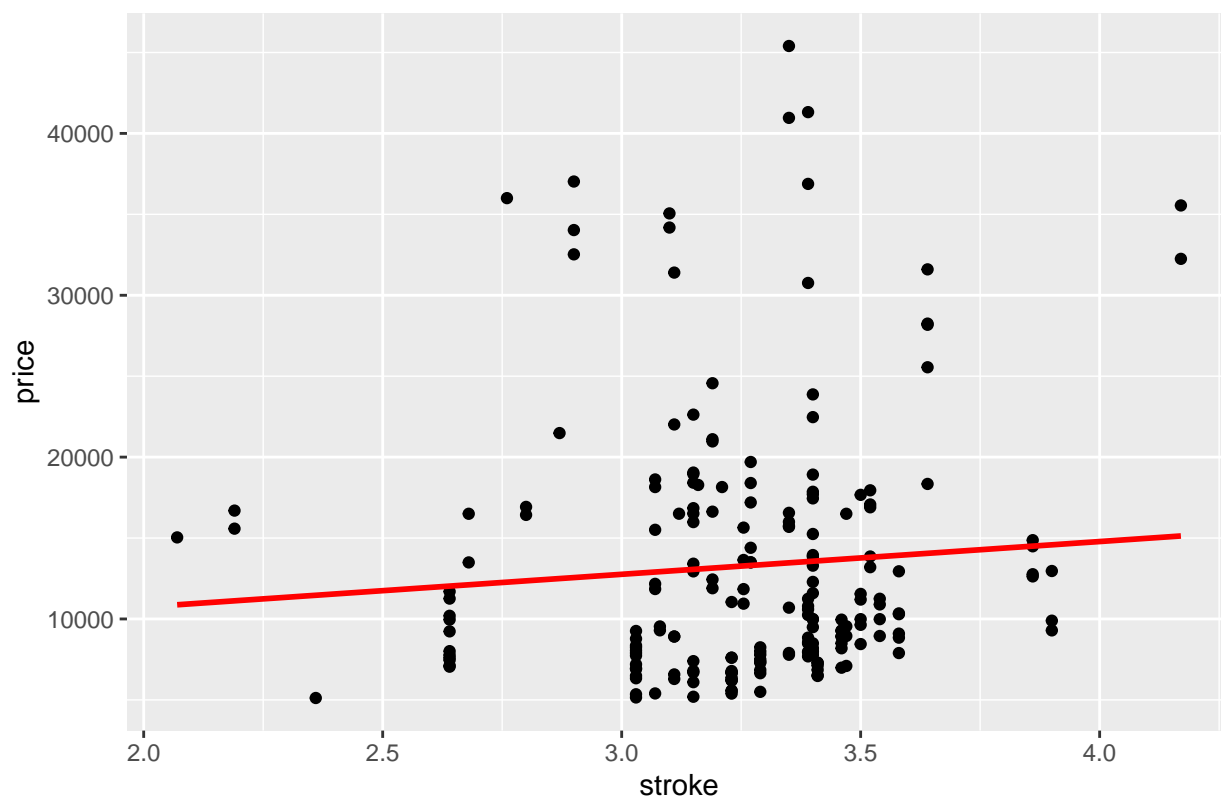
```
## `geom_smooth()` using formula = 'y ~ x'
```

Scatter plot of enginesize vs price



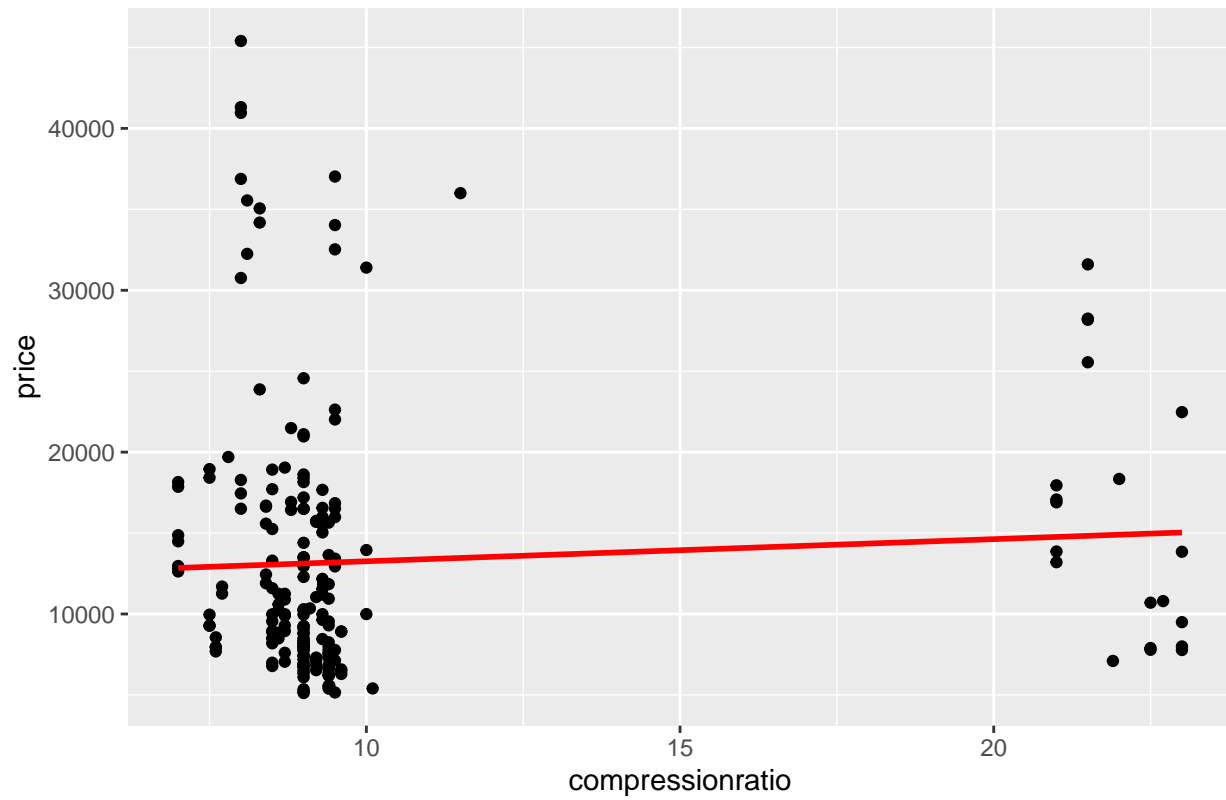
```
## `geom_smooth()` using formula = 'y ~ x'
```

Scatter plot of stroke vs price



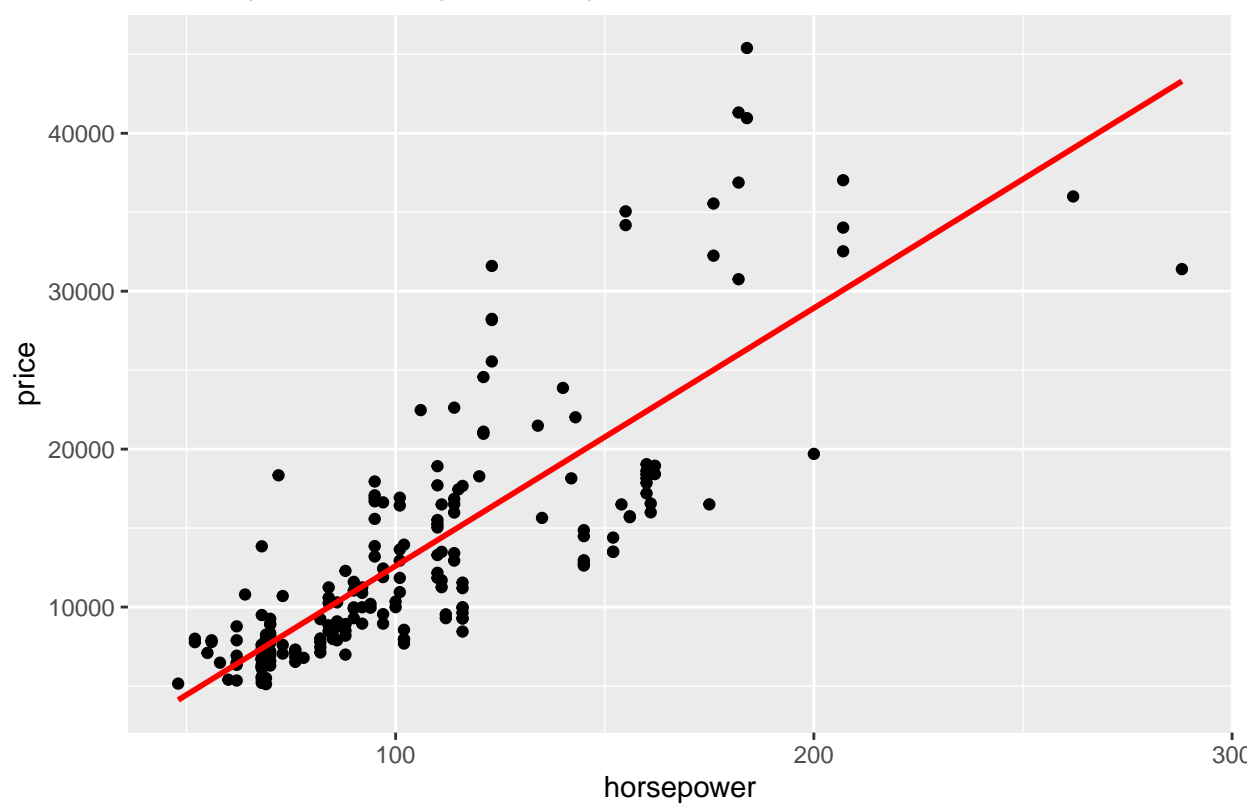
```
## `geom_smooth()` using formula = 'y ~ x'
```

Scatter plot of compressionratio vs price



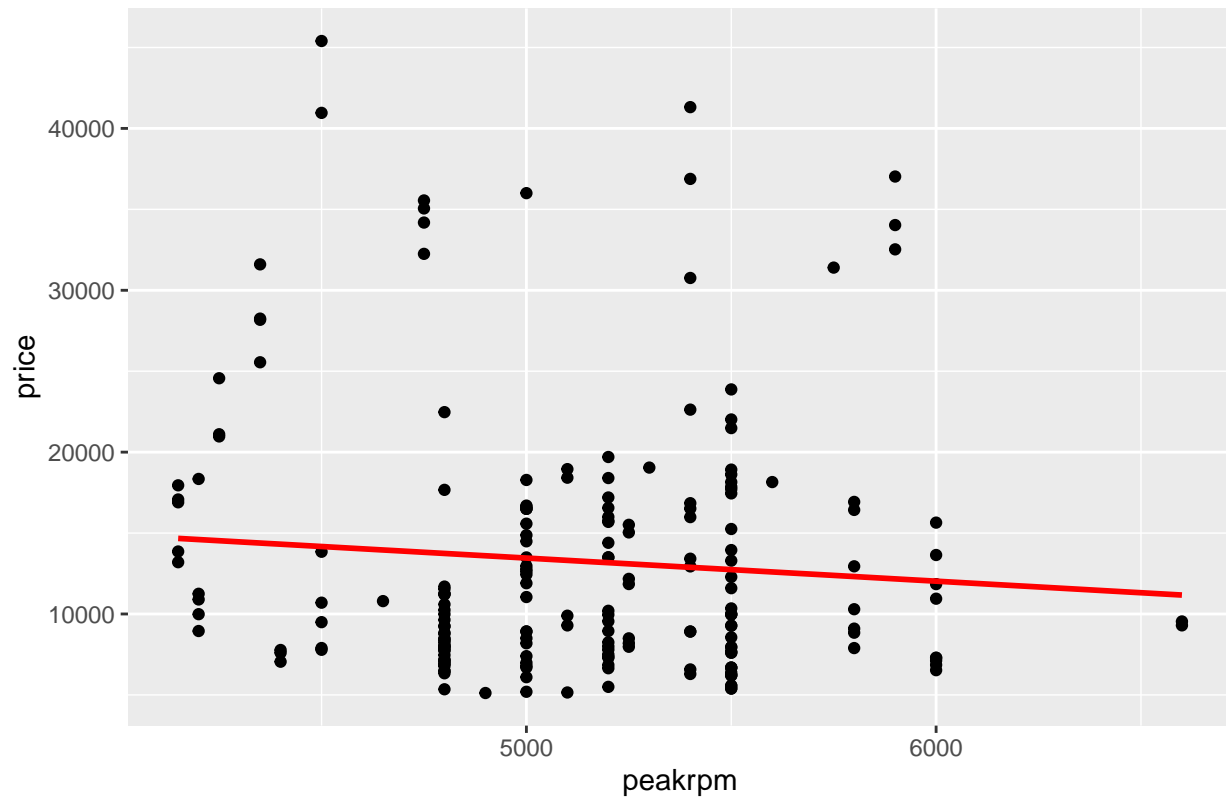
```
## `geom_smooth()` using formula = 'y ~ x'
```

Scatter plot of horsepower vs price



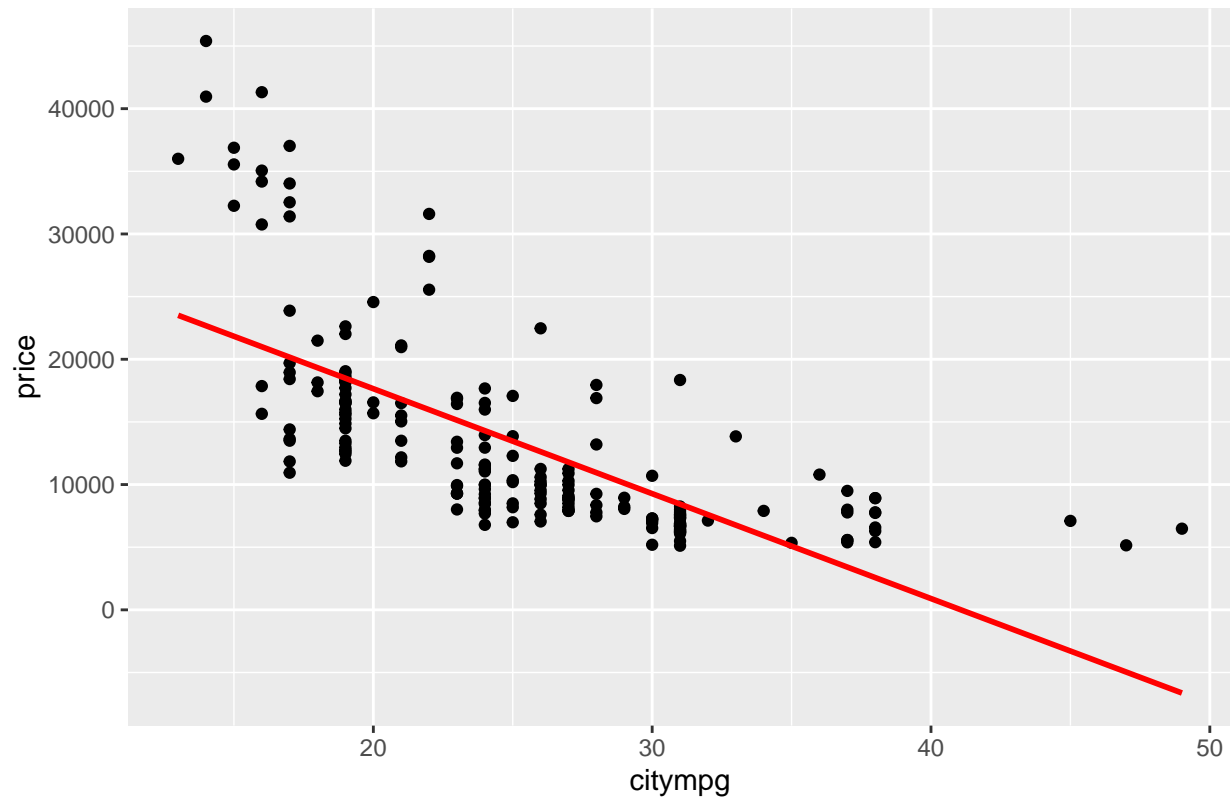
```
## `geom_smooth()` using formula = 'y ~ x'
```

Scatter plot of peakrpm vs price



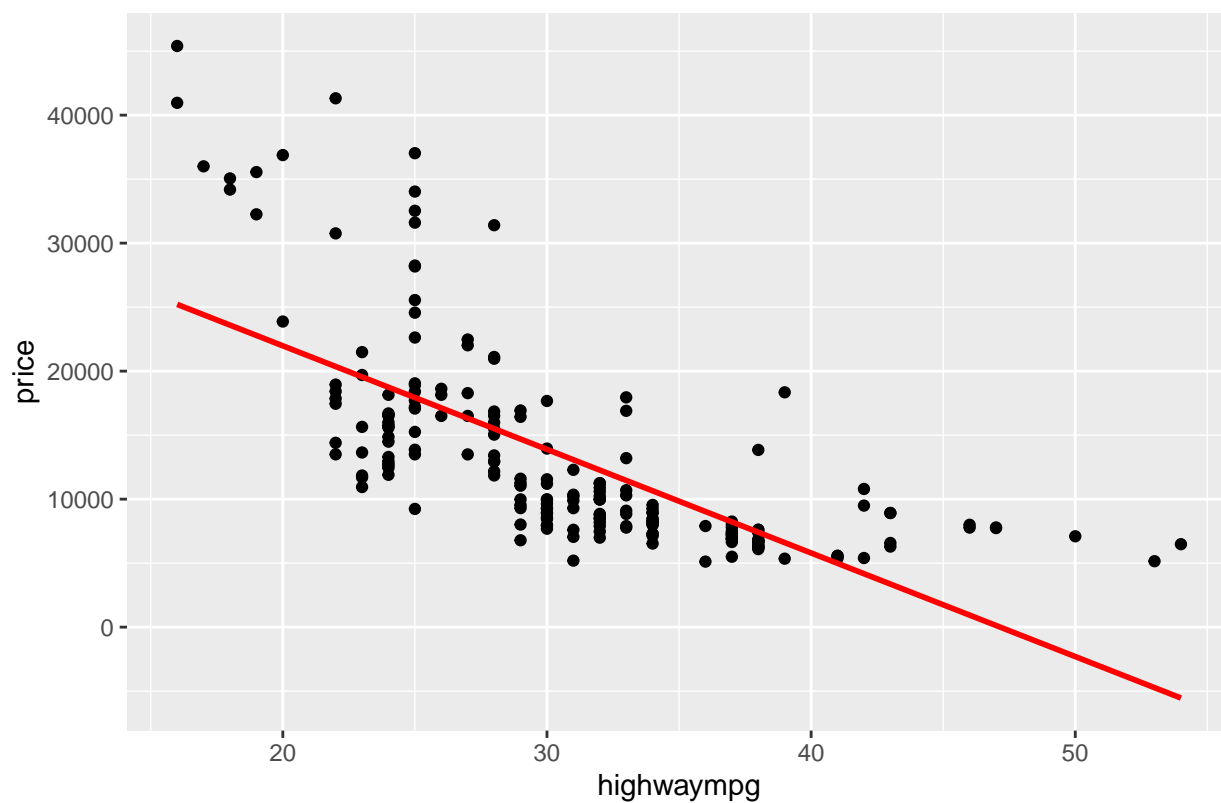
```
## `geom_smooth()` using formula = 'y ~ x'
```

Scatter plot of citympg vs price



```
## `geom_smooth()` using formula = 'y ~ x'
```


Scatter plot of highwaympg vs price



b) Variables categóricas

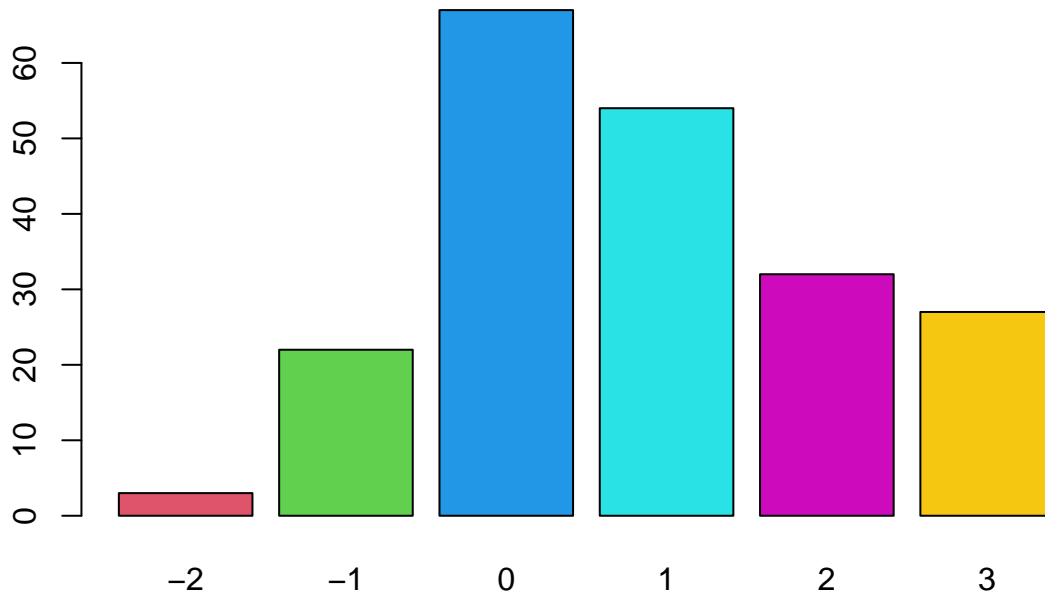
```
T = table(categorical_df$symboling)
T
```

Diagrama de barra

-2	-1	0	1	2	3
3	22	67	54	32	27

```
barplot(T, col = 2:15, main = "Frecuencia en categoría de riesgos por vehiculo")
```

Frecuencia en categoría de riesgos por vehiculo



Se observa que es de utilidad la clase “category”, ya que se acumulan 60 presencias en la categoría “0”, y 50 en la de “1” y se puede interpretar que la mayoría de los valores se acumulan entre esas dos categorías. Por otra parte hay 30 valores para categoría de riesgo “2” y muy pocas para la categoría más baja de riesgo.

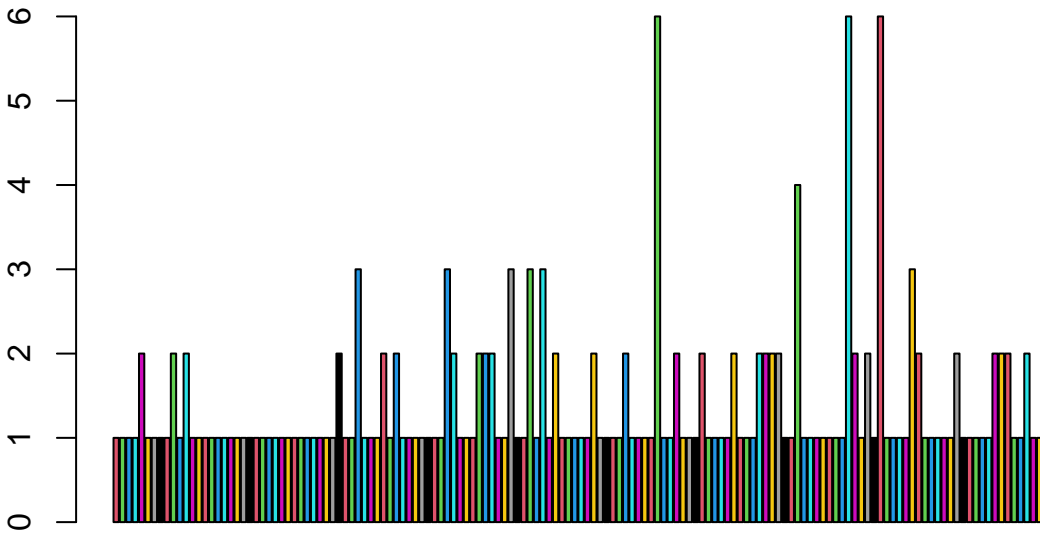
```
T = table(categorical_df$CarName)
```

T

[illegible]

```
barplot(T, col = 2:15, main = "Frecuencia de riesgos por vehiculo")
```

Frecuencia de riesgos por vehiculo



alfa-romero giulia honda civic nissan dayz saab 99e volkswagen type 3

Se observa que no es de utilidad la clase “CarName”, ya que son pocos valores por categoría y no tienen ningún patrón. La mayoría de las marcas tienen frecuencia de 1, y la máxima de 6.

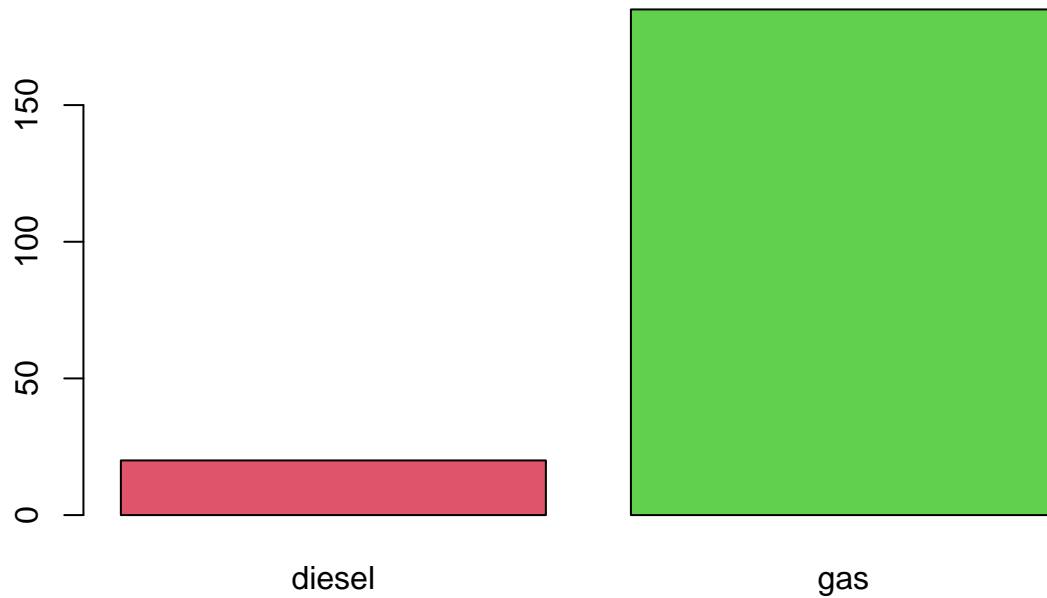
```
T = table(categorical_df$fueltype)
```

T

diesel	gas
20	185

```
barplot(T, col = 2:15, main = "Frecuencia de combustible por vehiculo")
```

Frecuencia de combustible por vehiculo



Aquí es muy notorio que la mayoría de los vehículos utilizan gas, teniendo una frecuencia de más de 150 y de 25 para diésel.

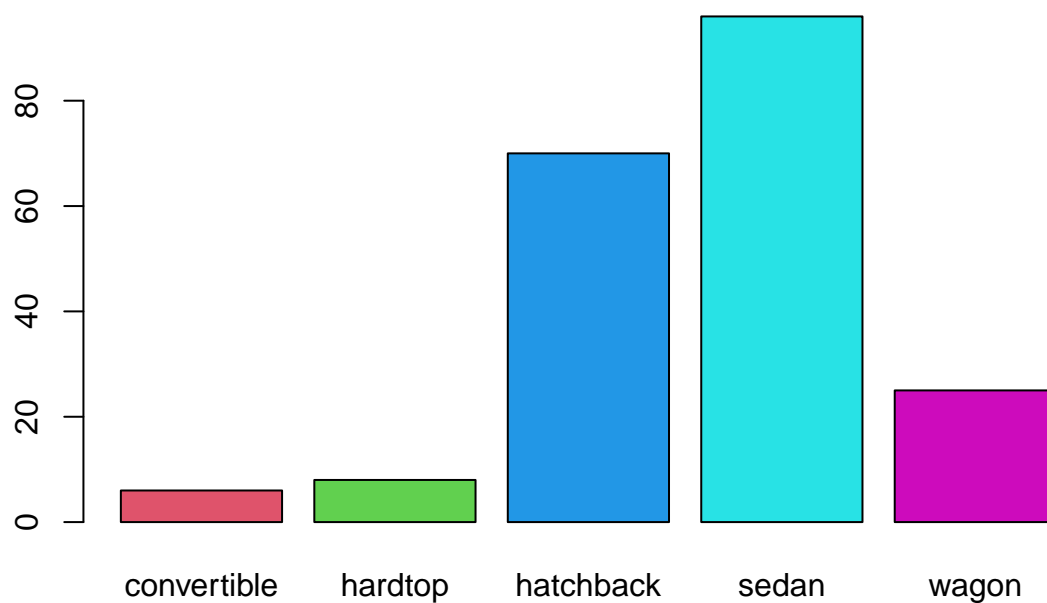
```
T = table(categorical_df$carbody)
```

```
T
```

convertible	hardtop	hatchback	sedan	wagon
6	8	70	96	25

```
barplot(T, col = 2:15, main = "Frecuencia de tipo de vehiculo")
```

Frecuencia de tipo de vehiculo



Se observa que hay más de 80 vehículos sedán, aproximadamente 70 de hatchback, y pocos convertibles, hardtop, y wagon.

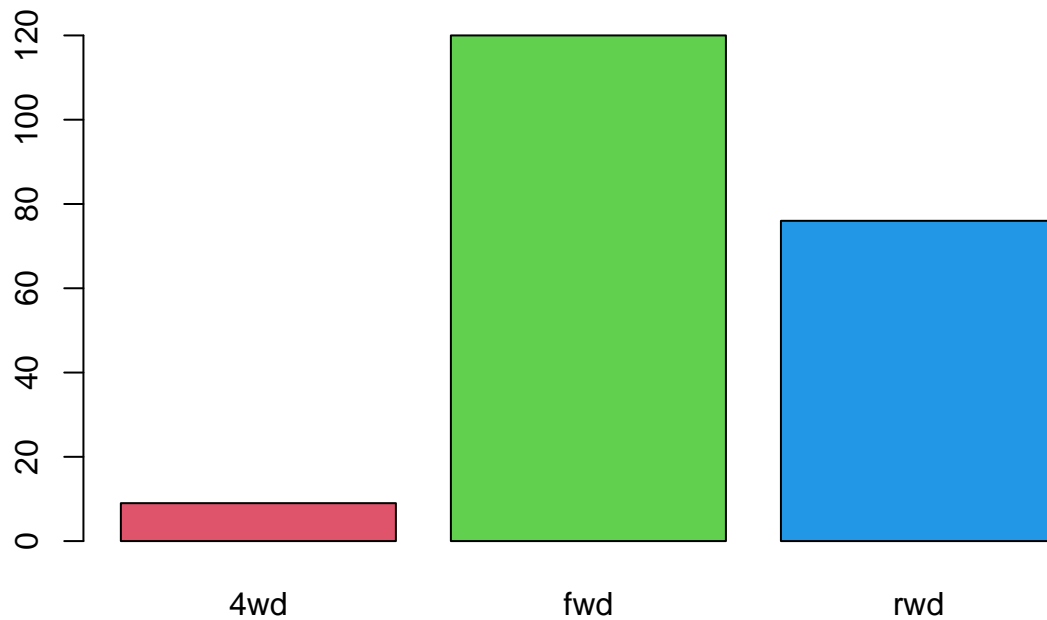
```
T = table(categorical_df$drivewheel)
```

```
T
```

4wd	fwd	rwd
9	120	76

```
barplot(T, col = 2:15, main = "Frecuencia de rueda motriz de vehiculo")
```

Frecuencia de rueda motriz de vehiculo



Se observa que más de 120 vehículos utilizan una rueda motriz “fwd”, la mitad utiliza un rwd y por último casi 10 utilizan “4wd”. Por lo que el “4wd” no es tan significativo en el análisis.

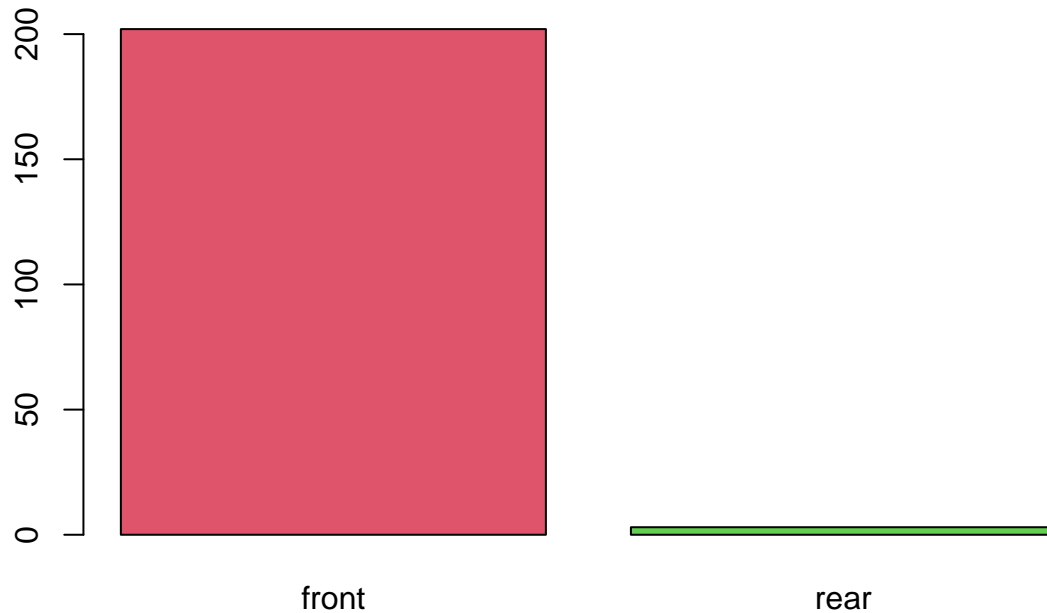
```
T = table(categorical_df$enginelocation)
```

```
T
```

front	rear
202	3

```
barplot(T, col = 2:15, main = "Frecuencia de ubicación del motor de vehiculo")
```

Frecuencia de ubicación del motor de vehículo



Es muy evidente que 200 vehículos ubican su motor al frente, y menos de 50 en la parte trasera por lo que se tomará como más significativo el los motores al frente del vehículo.

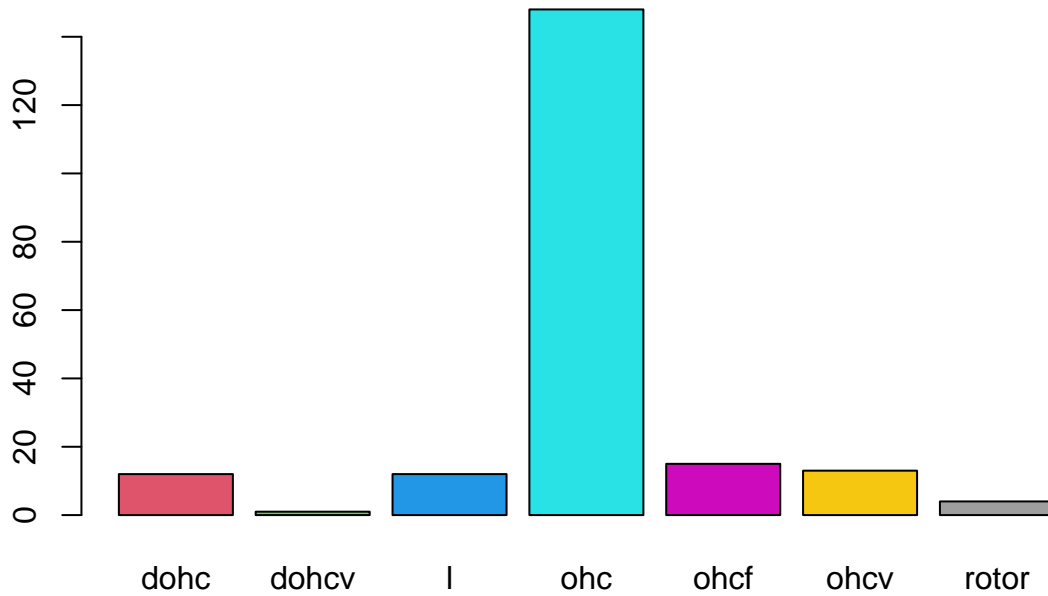
```
T = table(categorical_df$enginetype)
```

T

dohc	dohcv	l	ohc	ohcf	ohcv	rotor
12	1	12	148	15	13	4

```
barplot(T, col = 2:15, main = "Frecuencia de tipo de motor del vehiculo")
```

Frecuencia de tipo de motor del vehiculo



La mayoría de los vehículos utilizan el motor “ohc”, los valores “dohcv” y “rotor” son casi nulos por lo que serán no significativos.

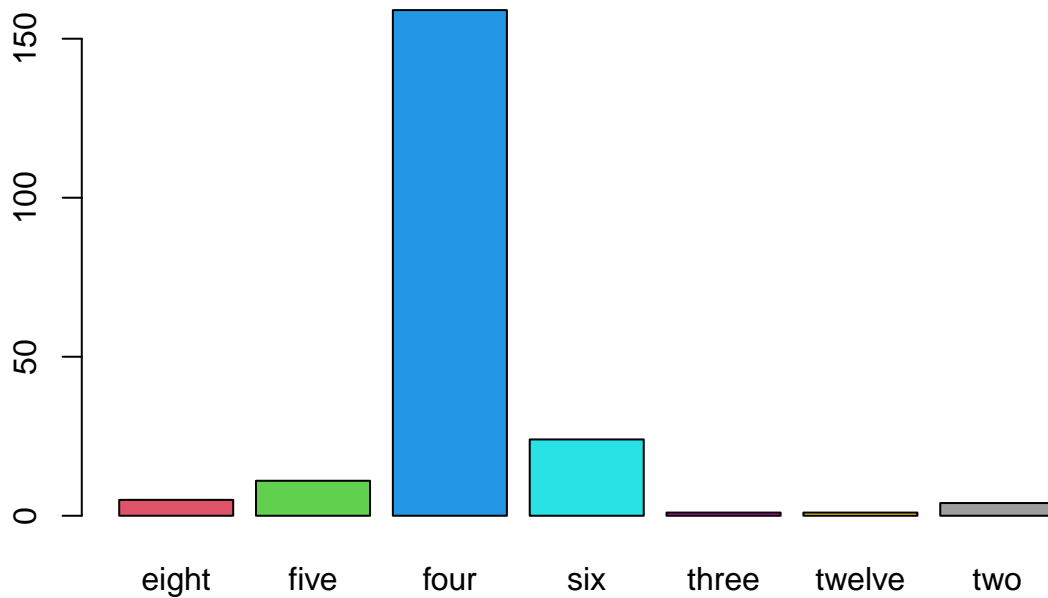
```
T = table(categorical_df$cylindernumber)
```

T

eight	five	four	six	three	twelve	two
5	11	159	24	1	1	4

```
barplot(T, col = 2:15, main = "Frecuencia en categoría de número de cilindros del motor de vehiculo")
```

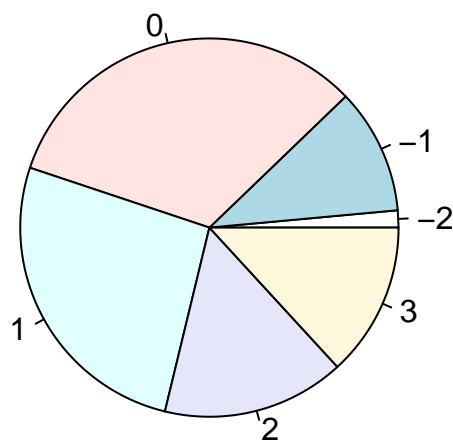

Frecuencia en categoría de número de cilindros del motor de vehicu



Se observa que la mayoría de los motores utilizan 4 cilindros, y los demás son tan pequeños que no se les consideran significativos para el análisis.

```
Tabla = table(categorical_df$symboling)
Tabla=prop.table(Tabla)
names(Tabla)=c("-2", "-1", "0", "1", "2", "3")
pie(Tabla, main = "Categoría de riesgo", labels = c("-2", "-1", "0", "1", "2", "3"))
```

Categoría de riesgo



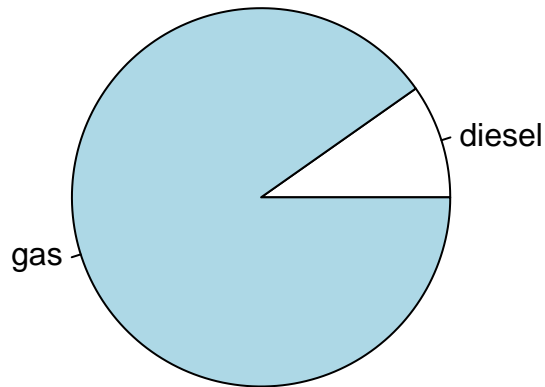
Diagramas de pastel

Se observa que hay una mayor cantidad de vehículos en la categoría de riesgo 0 y una muy baja en la -2.

```
Tabla = table(categorical_df$fueltype)
Tabla=prop.table(Tabla)
names(Tabla)=c("diesel", "gas")
```

```
pie(Tabla, main = "Tipo de combustible", labels = c("diesel","gas"))
```

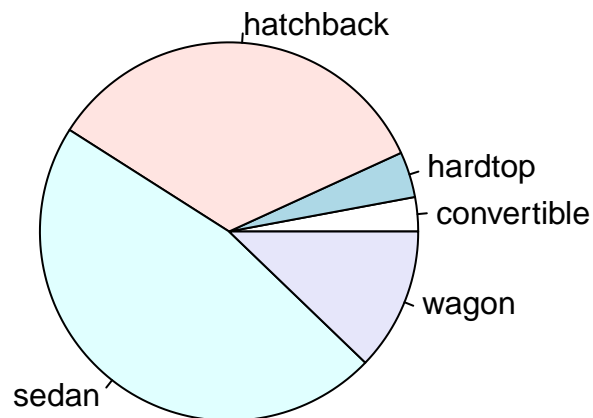
Tipo de combustible



Así como se observó en la gráfica de barras, el uso o consumo de gas por vehículos predomina sobre el uso del diésel.

```
Tabla = table(categorical_df$carbody)
Tabla=prop.table(Tabla)
names(Tabla)=c("convertible", "hardtop", "hatchback", "sedan", "wagon")
pie(Tabla, main = "Tipo de vehículo", labels = c("convertible", "hardtop", "hatchback", "sedan", "wagon"))
```

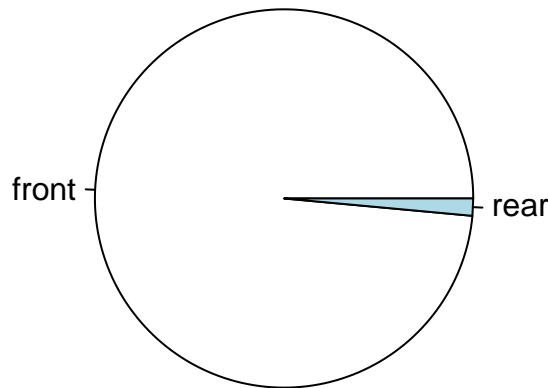
Tipo de vehículo



Así como en el gráfico de barras, se observa que predomina el sedán.

```
Tabla = table(categorical_df$enginelocation)
Tabla=prop.table(Tabla)
names(Tabla)=c("front", "rear")
pie(Tabla, main = "Ubicación del motor", labels = c("front", "rear"))
```

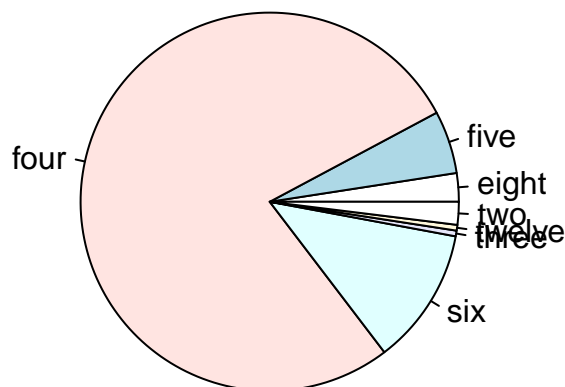
Ubicación del motor



Así como en el gráfico de barras, se observa que la mayoría de los autos tienen su motor en el frente.

```
Tabla = table(categorical_df$cylindernumber)
Tabla=prop.table(Tabla)
names(Tabla)=c("eight", "five", "four", "six", "three", "twelve", "two")
pie(Tabla, main = "Número de cilindros del motor", labels = c("eight", "five", "four", "six", "three"))
```

Número de cilindros del motor



Así como en el gráfico de barras, se observa que la mayoría de los autos tienen un motor de 4 cilindros.

Diagramas de caja y bigote de precio por categoría y barras por categoría

3. Identificación de problemas de calidad de datos

a) Valores faltantes

```
# Se verifica si hay NaN en el dataframe numerico
any(apply(numerical_df, 2, function(x) any(is.nan(x))))
```

Se evaluará si existen valores faltantes en el dataframe:

```
## [1] FALSE
```

No existen NaN en las variables numéricas

```
#Se verifica si hay NA en el dataframe categorico  
any(is.na(categorical_df))
```

```
## [1] FALSE
```

No existen NA en las variables numéricas

Se puede observar que no existen valores faltantes en el data frame por lo que el análisis puede ser significativo.

b) Outliers

Se observan los datos atípicos para las variables numéricas Teniendo en cuenta los boxplots que se realizaron y analizaron previamente se puede observar que existen muchos datos atípicos en la mayoría de las clases, pero, principalmente en la relación de la compresión de los automóviles y la distribución del tamaño de los motores para los automóviles. Por otra parte, no existe ningún dato atípico en los pesos en vacío de los automóviles. Sin embargo se hizo un mayor énfasis en el análisis de cada gráfica para explicar esto de una mejor manera.

Se observan los datos atípicos para las variables categóricas Como se observó en el gráfico de barras realizado y analizado previamente, para la la clase de número de cilindros por vehículo se tienen como valores atípicos tres cilindros y veinte cilindros ya que su presencia es muy poca y es insignificante. Además, para el tipo de motor de vehículo se consideran como outliers el motor “dohcv”, y “rotor”. Sin embargo se hizo un mayor énfasis en el análisis de cada gráfica para explicar esto de una mejor manera.

4. Variables importantes para el análisis de las características de los automóviles que determinan su precio.

A partir de todo lo mencionado anteriormente, como las gráficas, las frecuencias de cada variable, el resumen de sus medidas estadísticas y correlaciones con la variable dependiente “precio”, las variables que se consideran importantes para determinar el precio de un automóvil son:

1. Curbweight
2. Horsepower
3. Carwidth
4. enginesize
5. Citympg
6. Highwaympg
7. Symboling
8. Cylinder number
9. Engine type
10. Carbody

Estas variables fueron seleccionadas debido a que tienen una correlación o positiva o negativa con el precio del vehículo, y valores que son significativos para el análisis.

- Preparación de la base de datos

1. Selecciona el conjunto de datos a utilizar.

a) **Maneja datos categóricos: transforma a variables dummy si es necesario.** Primeramente eliminaré las variables que no serán de utilidad para mi análisis.

```
columns_to_keep <- c("curbweight", "horsepower", "carwidth", "enginesize", "citympg", "highwaympg", "symboling")

# Crear un nuevo dataframe solo con las columnas seleccionadas
df_selected <- M[, columns_to_keep]
head(df_selected)
```

curbweight	horsepower	carwidth	enginesize	citympg	highwaympg	symboling	cylindernumber	price	engine	type	carbody
2548	111	64.1	130	21	27	3	four	13495	dohc		convertible
2548	111	64.1	130	21	27	3	four	16500	dohc		convertible
2823	154	65.5	152	19	26	1	six	16500	ohcv		hatchback
2337	102	66.2	109	24	30	2	four	13950	ohc		sedan
2824	115	66.4	136	18	22	2	five	17450	ohc		sedan
2507	110	66.3	136	19	25	2	five	15250	ohc		sedan

b) Maneja apropiadamente datos atípicos. Posteriormente, eliminaré los datos atípicos de las variables numéricas ya que eliminar los outliers de las variables categóricas realmente no influiría mucho en el análisis de los valores.

Para esto, primeramente definí las variables numéricas a las que se les quieren extraer los outliers.

```
# Definir las variables numericas
num_vars <- c("curbweight", "horsepower", "carwidth", "enginesize", "citympg", "highwaympg", "symboling")

#remover outliers
remove_outliers <- function(df, num_vars) {
  # Se registran los valores atipicos
  keep_rows <- rep(TRUE, nrow(df))

  # Ciclo for para cada variable numerica
  for (var in num_vars) {
    # Se calculan los valores atipicos de cada variable numerica
    outliers <- boxplot.stats(df[[var]])$out

    # Se remueven las filas con outliers
    keep_rows <- keep_rows & !df[[var]] %in% outliers
  }

  # Se regresa un dataframe sin los valores atipicos
  df[keep_rows, ]
}

# Se remueven los outliers y se crea un nuevo dataframe con los valores sin outliers
df_selected_no_outliers <- remove_outliers(df_selected, num_vars)

head(df_selected_no_outliers)
```

curbweight	horsepower	carwidth	enginesize	citympg	highwaympg	symboling	cylindernumber	price	engine	type	carbody
2548	111	64.1	130	21	27	3	four	13495	dohc		convertible
2548	111	64.1	130	21	27	3	four	16500	dohc		convertible
2823	154	65.5	152	19	26	1	six	16500	ohcv		hatchback
2337	102	66.2	109	24	30	2	four	13950	ohc		sedan
2824	115	66.4	136	18	22	2	five	17450	ohc		sedan
2507	110	66.3	136	19	25	2	five	15250	ohc		sedan

```
curbweight,horsepower,carwidth,engine_size,citympg,highwaympg,year,engine_type,carbody
```

Por último se crean variables dummy para el mejor análisis de los datos, volviendo las variables categóricas a numéricas en un rango entre 0 o 1. Si están presentes, estas tendrán un valor de 1, en caso contrario, su valor será de 0.

```
data <- df_selected_no_outliers
# Convertir variables categoricas a variables dummy
dummy_columns <- character()

for (col in names(data)) {

  if (is.factor(data[[col]]) || is.character(data[[col]])) {

    dummy_mat <- model.matrix(~ data[[col]] - 1, data = data)

    colnames(dummy_mat) <- sub("^data\\\[([\"'\"]|['\"])\$\"", "", colnames(dummy_mat), perl = TRUE)

    # Agregar columnas dummy a dataframe sin outliers
    data <- cbind(data, dummy_mat)

    # Almacenar nombres de las columnas dummy para cada variable categorica
    dummy_columns <- c(dummy_columns, colnames(dummy_mat))
  }
}

# Crear un nuevo dataframe solo con las columnas numéricas y las variables dummy, sin categoricas
numeric_columns <- c("curbweight", "horsepower", "carwidth", "enginesize", "citympg", "highwaympg", "syndr")
selected_columns <- c(dummy_columns, numeric_columns)
df_cleaned <- data[selected_columns]
head(df_cleaned)
```

data	[data]	[data]	[data]	[data]	[data]	[data]	[data]	[data]	[data]	[data]	[data]	[data]	[data]	[data]	[data]	weight	moving	in	size	high	sampling
0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	2548111	64.1130	21	27	3	13495	
0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	2548111	64.1130	21	27	3	16500	
0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	2823154	65.5152	19	26	1	16500	
0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	2337102	66.2109	24	30	2	13950	
1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	2824115	66.4136	18	22	2	17450	
1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	2507110	66.3136	19	25	2	15250	

2. Transforma los datos en caso necesario.

a) **Revisa si es necesario discretizar los datos** Considero que no es necesario discretizar los datos ya que al predecir el precio de un automóvil es preferible mantener las variables continuas en lugar de discretizarlas para conservar la información detallada en los datos sin perder información importante sobre la relación entre las variables de los automóviles y su precio. Al tener una gran cantidad de variables continuas creo que perdería valores reales al discretizarlos y no reflejaría la realidad del contexto, lo cual no es conveniente para el cliente.

b) **Revisa si es necesario escalar y normalizar los datos** Teniendo en cuenta que las distintas variables tienen diferentes sesgos, y están distribuidos hacia la derecha o hacia la izquierda, considero prudente para

el análisis el normalizar los datos. Además, si en el futuro decido usar algún algoritmo de optimización, la normalización puede mejorar la eficiencia del entrenamiento y tener una mejor predicción.

```
# Lista de variables numéricas
numeric_columns <- c("curbweight", "horsepower", "carwidth", "enginesize", "citympg", "highwaympg", "symboling")

# Normalizar variables numéricas
normalize_zscore <- function(column) {
  (column - mean(column, na.rm = TRUE)) / sd(column, na.rm = TRUE)
}

# Crear un nuevo dataframe para los datos normalizados
df_normalized <- df_cleaned

# Normalizar las columnas numéricas
df_normalized[numeric_columns] <- lapply(df_normalized[numeric_columns], normalize_zscore)

head(df_normalized)
```

data	city	highway	curbweight	horsepower	carwidth	enginesize	citympg	highwaympg	symboling	price	engine	carbody	carcolor	carname	carid	caryear	carmonth	carweek	carhour	carmin	carsec	carms	carus	carother	carinfo	carphoto	carvideo	caraudio	carradio	carstereo	carcd	carcd2	carcd3	carcd4	carcd5	carcd6	carcd7	carcd8	carcd9	carcd10	carcd11	carcd12	carcd13	carcd14	carcd15	carcd16	carcd17	carcd18	carcd19	carcd20	carcd21	carcd22	carcd23	carcd24	carcd25	carcd26	carcd27	carcd28	carcd29	carcd30	carcd31	carcd32	carcd33	carcd34	carcd35	carcd36	carcd37	carcd38	carcd39	carcd40	carcd41	carcd42	carcd43	carcd44	carcd45	carcd46	carcd47	carcd48	carcd49	carcd50	carcd51	carcd52	carcd53	carcd54	carcd55	carcd56	carcd57	carcd58	carcd59	carcd60	carcd61	carcd62	carcd63	carcd64	carcd65	carcd66	carcd67	carcd68	carcd69	carcd70	carcd71	carcd72	carcd73	carcd74	carcd75	carcd76	carcd77	carcd78	carcd79	carcd80	carcd81	carcd82	carcd83	carcd84	carcd85	carcd86	carcd87	carcd88	carcd89	carcd90	carcd91	carcd92	carcd93	carcd94	carcd95	carcd96	carcd97	carcd98	carcd99	carcd100	carcd101	carcd102	carcd103	carcd104	carcd105	carcd106	carcd107	carcd108	carcd109	carcd110	carcd111	carcd112	carcd113	carcd114	carcd115	carcd116	carcd117	carcd118	carcd119	carcd120	carcd121	carcd122	carcd123	carcd124	carcd125	carcd126	carcd127	carcd128	carcd129	carcd130	carcd131	carcd132	carcd133	carcd134	carcd135	carcd136	carcd137	carcd138	carcd139	carcd140	carcd141	carcd142	carcd143	carcd144	carcd145	carcd146	carcd147	carcd148	carcd149	carcd150	carcd151	carcd152	carcd153	carcd154	carcd155	carcd156	carcd157	carcd158	carcd159	carcd160	carcd161	carcd162	carcd163	carcd164	carcd165	carcd166	carcd167	carcd168	carcd169	carcd170	carcd171	carcd172	carcd173	carcd174	carcd175	carcd176	carcd177	carcd178	carcd179	carcd180	carcd181	carcd182	carcd183	carcd184	carcd185	carcd186	carcd187	carcd188	carcd189	carcd190	carcd191	carcd192	carcd193	carcd194	carcd195	carcd196	carcd197	carcd198	carcd199	carcd200	carcd201	carcd202	carcd203	carcd204	carcd205	carcd206	carcd207	carcd208	carcd209	carcd210	carcd211	carcd212	carcd213	carcd214	carcd215	carcd216	carcd217	carcd218	carcd219	carcd220	carcd221	carcd222	carcd223	carcd224	carcd225	carcd226	carcd227	carcd228	carcd229	carcd230	carcd231	carcd232	carcd233	carcd234	carcd235	carcd236	carcd237	carcd238	carcd239	carcd240	carcd241	carcd242	carcd243	carcd244	carcd245	carcd246	carcd247	carcd248	carcd249	carcd250	carcd251	carcd252	carcd253	carcd254	carcd255	carcd256	carcd257	carcd258	carcd259	carcd260	carcd261	carcd262	carcd263	carcd264	carcd265	carcd266	carcd267	carcd268	carcd269	carcd270	carcd271	carcd272	carcd273	carcd274	carcd275	carcd276	carcd277	carcd278	carcd279	carcd280	carcd281	carcd282	carcd283	carcd284	carcd285	carcd286	carcd287	carcd288	carcd289	carcd290	carcd291	carcd292	carcd293	carcd294	carcd295	carcd296	carcd297	carcd298	carcd299	carcd300	carcd301	carcd302	carcd303	carcd304	carcd305	carcd306	carcd307	carcd308	carcd309	carcd310	carcd311	carcd312	carcd313	carcd314	carcd315	carcd316	carcd317	carcd318	carcd319	carcd320	carcd321	carcd322	carcd323	carcd324	carcd325	carcd326	carcd327	carcd328	carcd329	carcd330	carcd331	carcd332	carcd333	carcd334	carcd335	carcd336	carcd337	carcd338	carcd339	carcd340	carcd341	carcd342	carcd343	carcd344	carcd345	carcd346	carcd347	carcd348	carcd349	carcd350	carcd351	carcd352	carcd353	carcd354	carcd355	carcd356	carcd357	carcd358	carcd359	carcd360	carcd361	carcd362	carcd363	carcd364	carcd365	carcd366	carcd367	carcd368	carcd369	carcd370	carcd371	carcd372	carcd373	carcd374	carcd375	carcd376	carcd377	carcd378	carcd379	carcd380	carcd381	carcd382	carcd383	carcd384	carcd385	carcd386	carcd387	carcd388	carcd389	carcd390	carcd391	carcd392	carcd393	carcd394	carcd395	carcd396	carcd397	carcd398	carcd399	carcd400	carcd401	carcd402	carcd403	carcd404	carcd405	carcd406	carcd407	carcd408	carcd409	carcd410	carcd411	carcd412	carcd413	carcd414	carcd415	carcd416	carcd417	carcd418	carcd419	carcd420	carcd421	carcd422	carcd423	carcd424	carcd425	carcd426	carcd427	carcd428	carcd429	carcd430	carcd431	carcd432	carcd433	carcd434	carcd435	carcd436	carcd437	carcd438	carcd439	carcd440	carcd441	carcd442	carcd443	carcd444	carcd445	carcd446	carcd447	carcd448	carcd449	carcd450	carcd451	carcd452	carcd453	carcd454	carcd455	carcd456	carcd457	carcd458	carcd459	carcd460	carcd461	carcd462	carcd463	carcd464	carcd465	carcd466	carcd467	carcd468	carcd469	carcd470	carcd471	carcd472	carcd473	carcd474	carcd475	carcd476	carcd477	carcd478	carcd479	carcd480	carcd481	carcd482	carcd483	carcd484	carcd485	carcd486	carcd487	carcd488	carcd489	carcd490	carcd491	carcd492	carcd493	carcd494	carcd495	carcd496	carcd497	carcd498	carcd499	carcd500	carcd501	carcd502	carcd503	carcd504	carcd505	carcd506	carcd507	carcd508	carcd509	carcd510	carcd511	carcd512	carcd513	carcd514	carcd515	carcd516	carcd517	carcd518	carcd519	carcd520	carcd521	carcd522	carcd523	carcd524	carcd525	carcd526	carcd527	carcd528	carcd529	carcd530	carcd531	carcd532	carcd533	carcd534	carcd535	carcd536	carcd537	carcd538	carcd539	carcd540	carcd541	carcd542	carcd543	carcd544	carcd545	carcd546	carcd547	carcd548	carcd549	carcd550	carcd551	carcd552	carcd553	carcd554	carcd555	carcd556	carcd557	carcd558	carcd559	carcd560	carcd561	carcd562	carcd563	carcd564	carcd565	carcd566	carcd567	carcd568	carcd569	carcd570	carcd571	carcd572	carcd573	carcd574	carcd575	carcd576	carcd577	carcd578	carcd579	carcd580	carcd581	carcd582	carcd583	carcd584	carcd585	carcd586	carcd587	carcd588	carcd589	carcd590	carcd591	carcd592	carcd593	carcd594	carcd595	carcd596	carcd597	carcd598	carcd599	carcd600	carcd601	carcd602	carcd603	carcd604	carcd605	carcd606	carcd607	carcd608	carcd609	carcd610	carcd611	carcd612	carcd613	carcd614	carcd615	carcd616	carcd617	carcd618	carcd619	carcd620	carcd621	carcd622	carcd623	carcd624	carcd625	carcd626	carcd627	carcd628	carcd629	carcd630	carcd631	carcd632	carcd633	carcd634	carcd635	carcd636	carcd637	carcd638	carcd639	carcd640	carcd641	carcd642	carcd643	carcd644	carcd645	carcd646	carcd647	carcd648	carcd649	carcd650	carcd651	carcd652	carcd653	carcd654	carcd655	carcd656	carcd657	carcd658	carcd659	carcd660	carcd661	carcd662	carcd663	carcd664	carcd665	carcd666	carcd667	carcd668	carcd669	carcd670	carcd671	carcd672	carcd673	carcd674	carcd675	carcd676	carcd677	carcd678	carcd679	carcd680	carcd681	carcd682	carcd683	carcd684	carcd685	carcd686	carcd687	carcd688	carcd689	carcd690	carcd691	carcd692	carcd693	carcd694	carcd695	carcd696	carcd697	carcd698	carcd699	carcd700	carcd701	carcd702	carcd703	carcd704	carcd705	carcd706	carcd707	carcd708	carcd709	carcd710	carcd711	carcd712	carcd713	carcd714	carcd715	carcd716	carcd717	carcd718	carcd719	carcd720	carcd721	carcd722	carcd723	carcd724	carcd725	carcd726	carcd727	carcd728	carcd729	carcd730	carcd731	carcd732	carcd733	carcd734	carcd735	carcd736	carcd737	carcd738	carcd739	carcd740	carcd741	carcd742	carcd743	carcd744	carcd745	carcd746	carcd747	carcd748	carcd749	carcd750	carcd751	carcd752	carcd753	carcd754	carcd755	carcd756	carcd757	carcd758	carcd759	carcd760	carcd761	carcd762	carcd763	carcd764	carcd765	carcd766	carcd767	carcd768	carcd769	carcd770	carcd771	carcd772	carcd773	carcd774	carcd775	carcd776	carcd777	carcd778	carcd779	carcd780	carcd781	carcd782	carcd783	carcd784	carcd785	carcd786	carcd787	carcd788	carcd789	carcd790	carcd791	carcd792	carcd793	carcd794	carcd795	carcd796	carcd797	carcd798	carcd799	carcd800	carcd801	carcd802	carcd803	carcd804	carcd805	carcd806	carcd807	carcd808	carcd809	carcd810	carcd811	carcd812	carcd813	carcd814	carcd815	carcd816	carcd817	carcd818	carcd819	carcd820	carcd821	carcd822	carcd823	carcd824	carcd825	carcd826	carcd827	carcd828	carcd829	carcd830	carcd831	carcd832	carcd833	carcd834	carcd835	carcd836	carcd837	carcd838	carcd839	carcd840	carcd841	carcd842	carcd843	carcd844	carcd845	carcd846	carcd847	carcd848	carcd849	carcd850	carcd851	carcd852	carcd853	carcd854	carcd855	carcd856	carcd857	carcd858	carcd859	carcd860	carcd861	carcd862	carcd863	carcd864	carcd865	carcd866	carcd867	carcd868	carcd869	carcd870	carcd871	carcd872	carcd873	carcd874	carcd875	carcd876	carcd877	carcd878	carcd879	carcd880	carcd881	carcd882	carcd883	carcd884	carcd885	carcd886	carcd887	carcd888	carcd889	carcd890	carcd891	carcd892	carcd893	carcd894	carcd895	carcd896	carcd897	carcd898	carcd899	carcd900	carcd901	carcd902	carcd903	carcd904	carcd905	carcd906	carcd907	carcd908	carcd909	carcd910	carcd911	carcd912	carcd913	carcd914	carcd915	carcd916	carcd917	carcd918	carcd919	carcd920	carcd921	carcd922	carcd923	carcd924	carcd925	carcd926	carcd927	carcd928	carcd929	carcd930	carcd931	carcd932	carcd933	carcd934	carcd935	carcd936	carcd937	carcd938	carcd939	carcd940	carcd941	carcd942	carcd943	carcd944	carcd945	carcd946	carcd947	carcd948	carcd949	carcd950	carcd951	carcd952	carcd953	carcd954	carcd955	carcd956	carcd957	carcd958	carcd959	carcd960	carcd961	carcd962	carcd963	carcd964	carcd965	carcd966	carcd967	carcd968	carcd969	carcd970	carcd971	carcd972	carcd973	carcd974	carcd975	carcd976	carcd977	carcd978	carcd979	carcd980	carcd981	carcd982	carcd983	carcd984	carcd985	carcd986	carcd987	carcd988	carcd989	carcd990	carcd991	carcd992	carcd993	carcd994	carcd995	carcd996	carcd997	carcd998	carcd999	carcd1000	carcd1001	carcd1002	carcd1003	carcd1004	carcd1005	carcd1006	carcd1007	carcd1008	carcd1009	carcd1010	carcd1011	carcd1012	carcd1013	carcd1014	carcd1015	carcd1016	carcd1017	carcd1018	carcd1019	carcd1020	carcd1021	carcd1022	carcd1023	carcd1024	carcd1025	carcd1026	carcd1027	carcd1028	carcd1029	carcd1030	carcd1031	carcd1032	carcd1033	carcd1034	carcd1035	carcd1036	carcd1037	carcd1038	carcd1039	carcd1040	carcd1041	carcd1042	carcd1043	carcd1044	carcd1045	carcd1046	carcd1047	carcd1048	carcd1049	carcd1050	carcd1051	carcd1052	carcd1053	carcd1054	carcd1055	carcd1056	carcd1057	carcd1058	carcd1059	carcd1060	carcd1061	carcd1062	carcd1063	carcd1064	carcd1065	carcd1066	carcd1067	carcd1068	carcd1069	carcd1070	carcd1071	carcd1072	carcd1073	carcd1074	carcd1075	carcd1076	carcd1077	carcd1078	carcd1079	carcd1080	carcd1081	carcd1082	carcd1083	carcd1084	carcd1085	carcd1086	carcd1087	carcd1088	carcd1089	carcd1090	carcd1091	carcd1092	carcd1093	carcd1094	carcd1095	carcd1096	carcd1097	carcd1098	carcd1099	carcd1100	carcd1101	carcd1102	carcd1103	carcd1104	carcd1105	carcd1106	carcd1107	carcd1108	carcd1109	carcd1110	carcd1111	carcd1112	carcd1113	carcd1114	carcd1115	carcd1116	carcd1117	carcd1118	carcd1119	carcd1120	carcd1121	carcd1122	carcd1123	carcd1124	carcd1125	carcd1126	carcd1127	carcd1128	carcd1129	carcd1130	carcd1131	carcd1132	carcd1133	carcd1134	carcd1135	carcd1136	carcd1137	carcd1138	carcd1139	carcd1140	carcd1141	carcd1142	carcd1143	carcd1144	carcd1145	carcd1146	carcd1147	carcd1148	carcd1149	carcd1150	carcd1151	carcd1152	carcd1153	carcd1154	carcd1155	carcd1156	carcd1157	carcd1158	carcd1159	carcd1160	carcd1161	carcd1162	carcd1163	carcd1164	carcd1165	carcd1166	carcd1167	carcd1168	carcd1169	carcd1170	carcd1171	carcd1172	carcd1173	carcd1174	carcd1175	carcd1176	carcd1177	carcd1178	carcd1179	carcd1180
------	------	---------	------------	------------	----------	------------	---------	------------	-----------	-------	--------	---------	----------	---------	-------	---------	----------	---------	---------	--------	--------	-------	-------	----------	---------	----------	----------	----------	----------	-----------	-------	--------	--------	--------	--------	--------	--------	--------	--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------