

# portafoliom1-act01-sem3-1

September 2, 2023

## 0.1 Módulo 1 Técnicas de procesamiento de datos para el análisis estadístico y para la construcción de modelos

### *Adrián Emmanuel Faz Mercado A01570770*

**Instrucciones:** La 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:

- Qué variables son significativas para predecir el precio de un automóvil
- Qué tan bien describen esas variables el precio de un automóvil

## 0.2 1. Exploración y preparación de la base de datos (Portafolio de Análisis)

Para comenzar a explorar y a preparar nuestra base de datos, se comenzará leyendo el archivo de los datos y almacenándolo en una variable como un dataframe.

```
[170]: mydata <- read.csv("precios_autos.csv")
mydata
```

	symboling <int>	CarName <chr>	fueltype <chr>	carbody <chr>	drivewheel <chr>	engine <chr>
	3	alfa-romero giulia	gas	convertible	rwd	front
	3	alfa-romero stelvio	gas	convertible	rwd	front
	1	alfa-romero Quadrifoglio	gas	hatchback	rwd	front
	2	audi 100 ls	gas	sedan	fwd	front
	2	audi 100ls	gas	sedan	4wd	front
	2	audi fox	gas	sedan	fwd	front
	1	audi 100ls	gas	sedan	fwd	front
	1	audi 5000	gas	wagon	fwd	front
	1	audi 4000	gas	sedan	fwd	front
	0	audi 5000s (diesel)	gas	hatchback	4wd	front
	2	bmw 320i	gas	sedan	rwd	front
	0	bmw 320i	gas	sedan	rwd	front
	0	bmw x1	gas	sedan	rwd	front
	0	bmw x3	gas	sedan	rwd	front
	1	bmw z4	gas	sedan	rwd	front
	0	bmw x4	gas	sedan	rwd	front
	0	bmw x5	gas	sedan	rwd	front
	0	bmw x3	gas	sedan	rwd	front
	2	chevrolet impala	gas	hatchback	fwd	front
	1	chevrolet monte carlo	gas	hatchback	fwd	front
	0	chevrolet vega 2300	gas	sedan	fwd	front
	1	dodge rampage	gas	hatchback	fwd	front
	1	dodge challenger se	gas	hatchback	fwd	front
	1	dodge d200	gas	hatchback	fwd	front
	1	dodge monaco (sw)	gas	hatchback	fwd	front
	1	dodge colt hardtop	gas	sedan	fwd	front
	1	dodge colt (sw)	gas	sedan	fwd	front
	1	dodge coronet custom	gas	sedan	fwd	front
	-1	dodge dart custom	gas	wagon	fwd	front
A data.frame: 205 × 21	3	dodge coronet custom (sw)	gas	hatchback	fwd	front
	-1	toyota corona	gas	hatchback	fwd	front
	-1	toyota corolla	gas	sedan	fwd	front
	-1	toyota mark ii	gas	hatchback	fwd	front
	3	toyota corolla liftback	gas	hatchback	rwd	front
	3	toyota corona	gas	hatchback	rwd	front
	-1	toyota starlet	gas	sedan	rwd	front
	-1	toyouta tercel	gas	wagon	rwd	front
	2	vokswagen rabbit	diesel	sedan	fwd	front
	2	volkswagen 1131 deluxe sedan	gas	sedan	fwd	front
	2	volkswagen model 111	diesel	sedan	fwd	front
	2	volkswagen type 3	gas	sedan	fwd	front
	2	volkswagen 411 (sw)	gas	sedan	fwd	front
	2	volkswagen super beetle	diesel	sedan	fwd	front
	2	volkswagen dasher	gas	sedan	fwd	front
	3	vw dasher	gas	convertible	fwd	front
	3	vw rabbit	gas	hatchback	fwd	front
	0	volkswagen rabbit	gas	sedan	fwd	front
	0	volkswagen rabbit custom	diesel	sedan	fwd	front
	0	volkswagen dasher	gas	wagon	fwd	front
	-2	volvo 145e (sw)	gas	sedan	rwd	front

Se crea una lista para nuestras variables categoricas y una lista para nuestras variables numéricas.

```
[171]: variables_cuantitativas <- c("wheelbase", "carlength", "carwidth", "carheight",  
  ↪ "curbweight",  
                                     "enginesize", "stroke", "compressionratio",  
  ↪ "horsepower",  
                                     "peakrpm", "citympg", "highwaympg", "price")  
  
variables_categoricas <- c("Symboling", "CarName", "fueltype", "carbody",  
  ↪ "drivewheel",  
                           "enginelocation", "enginetype", "cylindernumber")
```

Se obtienen las medidas estadísticas principales de todas las variables cuantitativas

```
[172]: for (variable in variables_cuantitativas) {  
  cat("Medidas variable: ", variable, ":\n")  
  print(summary(mydata[[variable]]))  
  
  cat("Desviación estándar: ")  
  print(sd(mydata[[variable]]))  
  cat("\n")  
}
```

```
Medidas variable: wheelbase :  
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
  86.60  94.50  97.00  98.76 102.40 120.90  
Desviación estándar: [1] 6.021776
```

```
Medidas variable: carlength :  
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
 141.1 166.3 173.2 174.0 183.1 208.1  
Desviación estándar: [1] 12.33729
```

```
Medidas variable: carwidth :  
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
  60.30  64.10  65.50  65.91  66.90  72.30  
Desviación estándar: [1] 2.145204
```

```
Medidas variable: carheight :  
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
  47.80  52.00  54.10  53.72  55.50  59.80  
Desviación estándar: [1] 2.443522
```

```
Medidas variable: curbweight :  
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
 1488  2145  2414  2556  2935  4066  
Desviación estándar: [1] 520.6802
```

Medidas variable: enginesize :  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
 61.0 97.0 120.0 126.9 141.0 326.0  
 Desviación estándar: [1] 41.64269

Medidas variable: stroke :  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
 2.070 3.110 3.290 3.255 3.410 4.170  
 Desviación estándar: [1] 0.313597

Medidas variable: compressionratio :  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
 7.00 8.60 9.00 10.14 9.40 23.00  
 Desviación estándar: [1] 3.97204

Medidas variable: horsepower :  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
 48.0 70.0 95.0 104.1 116.0 288.0  
 Desviación estándar: [1] 39.54417

Medidas variable: peakrpm :  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
 4150 4800 5200 5125 5500 6600  
 Desviación estándar: [1] 476.9856

Medidas variable: citympg :  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
 13.00 19.00 24.00 25.22 30.00 49.00  
 Desviación estándar: [1] 6.542142

Medidas variable: highwaympg :  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
 16.00 25.00 30.00 30.75 34.00 54.00  
 Desviación estándar: [1] 6.886443

Medidas variable: price :  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
 5118 7788 10295 13277 16503 45400  
 Desviación estándar: [1] 7988.852

Se calculan las frecuencias de las diferentes variables categóricas presentes de manera numérica.

```
[173]: for (variable in variables_categoricas) {
  cat("Frecuencias de la variable", variable, ":\n")
  print(table(mydata[[variable]]))
  cat("\n")
}
```

Frecuencias de la variable Symboling :  
 < table of extent 0 >

Frecuencias de la variable CarName :

alfa-romero giulia	1	alfa-romero Quadrifoglio	1
alfa-romero stelvio	1	audi 100 ls	1
audi 100ls	2	audi 4000	1
audi 5000	1	audi 5000s (diesel)	1
audi fox	1	bmw 320i	2
bmw x1	1	bmw x3	2
bmw x4	1	bmw x5	1
bmw z4	1	buick century	1
buick century luxus (sw)	1	buick century special	1
buick electra 225 custom	1	buick opel isuzu deluxe	1
buick regal sport coupe (turbo)	1	buick skyhawk	1
buick skylark	1	chevrolet impala	1
chevrolet monte carlo	1	chevrolet vega 2300	1
dodge challenger se	1	dodge colt (sw)	1
dodge colt hardtop	1	dodge coronet custom	1
dodge coronet custom (sw)	1	dodge d200	1
dodge dart custom	1	dodge monaco (sw)	1
dodge rampage	1	honda accord	2
honda accord cvcc	1	honda accord lx	1
honda civic	3	honda civic (auto)	1
honda civic 1300	1	honda civic 1500 gl	1
honda civic cvcc		honda prelude	

2	1
isuzu D-Max	isuzu D-Max V-Cross
2	1
isuzu MU-X	jaguar xf
1	1
jaguar xj	jaguar xk
1	1
maxda glc deluxe	maxda rx3
1	1
mazda 626	mazda glc
3	2
mazda glc 4	mazda glc custom
1	1
mazda glc custom 1	mazda glc deluxe
1	2
mazda rx-4	mazda rx-7 gs
2	2
mazda rx2 coupe	mercury cougar
1	1
mitsubishi g4	mitsubishi lancer
3	1
mitsubishi mirage	mitsubishi mirage g4
1	3
mitsubishi montero	mitsubishi outlander
1	3
mitsubishi pajero	nissan clipper
1	2
nissan dayz	nissan fuga
1	1
nissan gt-r	nissan juke
1	1
nissan kicks	nissan latio
1	2
nissan leaf	nissan note
1	1
nissan nv200	nissan otti
1	1
nissan rogue	nissan teana
2	1
nissan titan	Nissan versa
1	1
peugeot 304	peugeot 504
1	6
peugeot 504 (sw)	peugeot 505s turbo diesel
1	1
peugeot 604sl	plymouth cricket
2	1
plymouth duster	plymouth fury gran sedan

1	1
plymouth fury iii	plymouth satellite custom (sw)
2	1
plymouth valiant	porcshce panamera
1	1
porsche boxter	porsche cayenne
1	2
porsche macan	renault 12tl
1	1
renault 5 gtl	saab 99e
1	2
saab 99gle	saab 99le
2	2
subaru	subaru baja
2	1
subaru brz	subaru dl
1	4
subaru r1	subaru r2
1	1
subaru trezia	subaru tribeca
1	1
toyota carina	toyota celica gt
1	1
toyota celica gt liftback	toyota corolla
1	6
toyota corolla 1200	toyota corolla 1600 (sw)
2	1
toyota corolla liftback	toyota corolla tercel
2	1
toyota corona	toyota corona hardtop
6	1
toyota corona liftback	toyota corona mark ii
1	1
toyota cressida	toyota mark ii
1	3
toyota starlet	toyota tercel
2	1
toyouta tercel	vokswagen rabbit
1	1
volkswagen 1131 deluxe sedan	volkswagen 411 (sw)
1	1
volkswagen dasher	volkswagen model 111
2	1
volkswagen rabbit	volkswagen rabbit custom
1	1
volkswagen super beetle	volkswagen type 3
1	1
volvo 144ea	volvo 145e (sw)

	2		2
volvo 244dl		volvo 245	
	2		1
volvo 246		volvo 264gl	
	1		2
volvo diesel		vw dasher	
	1		1
vw rabbit			
	1		

Frecuencias de la variable fueltype :

diesel	gas
20	185

Frecuencias de la variable carbody :

convertible	hardtop	hatchback	sedan	wagon
6	8	70	96	25

Frecuencias de la variable drivewheel :

4wd fwd rwd
9 120 76

Frecuencias de la variable enginelocation :

front	rear
202	3

Frecuencias de la variable enginetype :

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

Frecuencias de la variable cylindernumber :

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

Antes de comenzar con el análisis y limpieza, es necesario saber si existen registros sin datos o con campos vacíos., para manipularlos y manejarlos adecuadamente

```
[174]: colSums(is.na(mydata))
```

```
symboling 0 CarName 0 fueltype 0 carbody 0 drivewheel 0 enginelocation 0 wheelbase
0 carlength 0 carwidth 0 carheight 0 curbweight 0 enginetype 0 cylindernumber 0
```



```

enginesize  0 stroke    0 compressionratio  0 horsepower  0 peakrpm    0 citympg    0
highwaympg                                0 price                                0

```

En este caso, afortunadamente no hay campos vacíos en ninguno de los registros, por lo que se podemos continuar con el análisis sin necesidad de hacer limpieza en este sentido, pues nuestros datos están listos.

Para analizar la distribución de los datos de las variables cuantitativas, a continuación se realizarán gráficas para mostrar su distribución y para poder definir si se tratan de distribuciones simétricas o no. Además, se realizará un análisis de la relación entre la variable y la variable dependiente, para analizar si podría ser una variable determinante para predecir el precio del auto.

Definimos una variable de precio, que contiene los valores de los precios en el set de datos. Este se usará para saber si hay una relación en las diferentes variables cuantitativas y cualitativas con el mismo.

```

[175]: data_price = mydata[["price"]]
install.packages("e1071")
library(e1071)

```

Installing package into ‘/usr/local/lib/R/site-library’  
(as ‘lib’ is unspecified)

## 0.3 Variables cuantitativas

### 0.3.1 Wheel Base

```

[176]: variable = "wheelbase"
data = mydata[[variable]]

# Histograma
hist(data,col=0,main=paste("Histograma de", variable))

# Diagrama de caja y bigotes
boxplot(data,horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

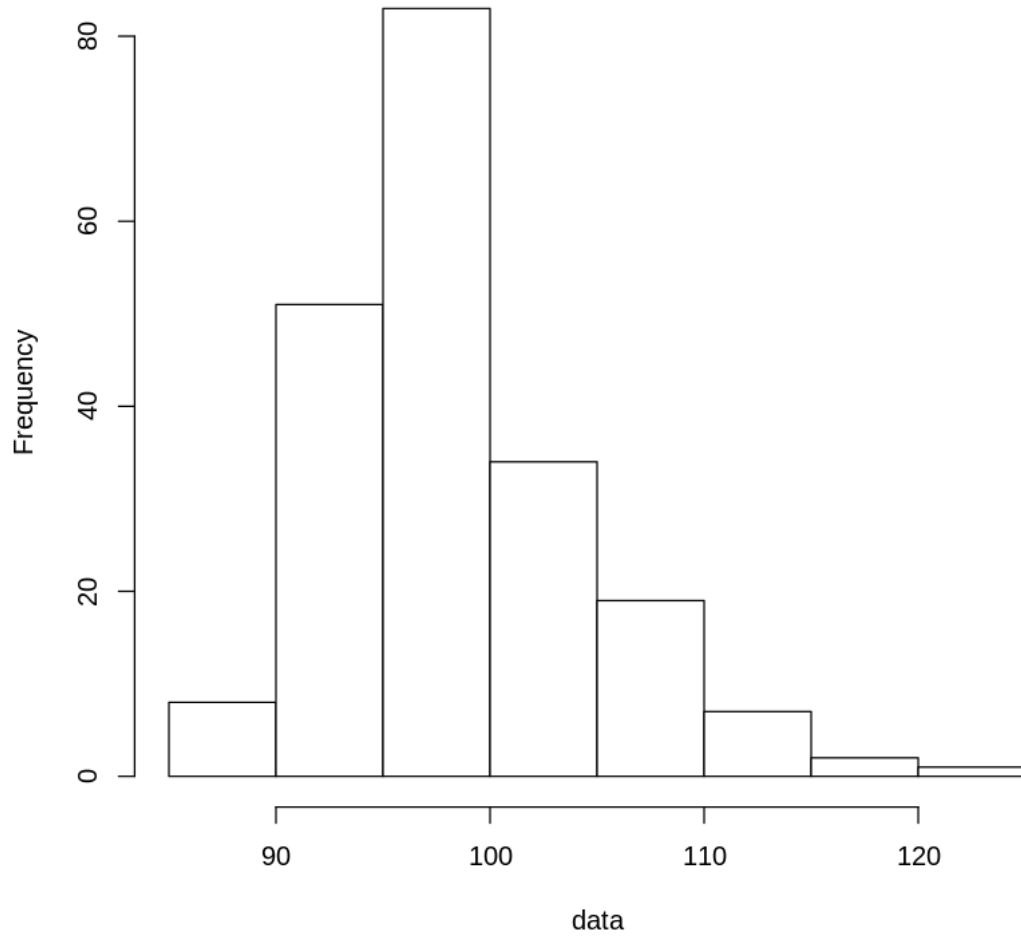
# Diagrama de dispersión
plot(data,data_price, main=paste("Diagrama de dispersión de", variable))

# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

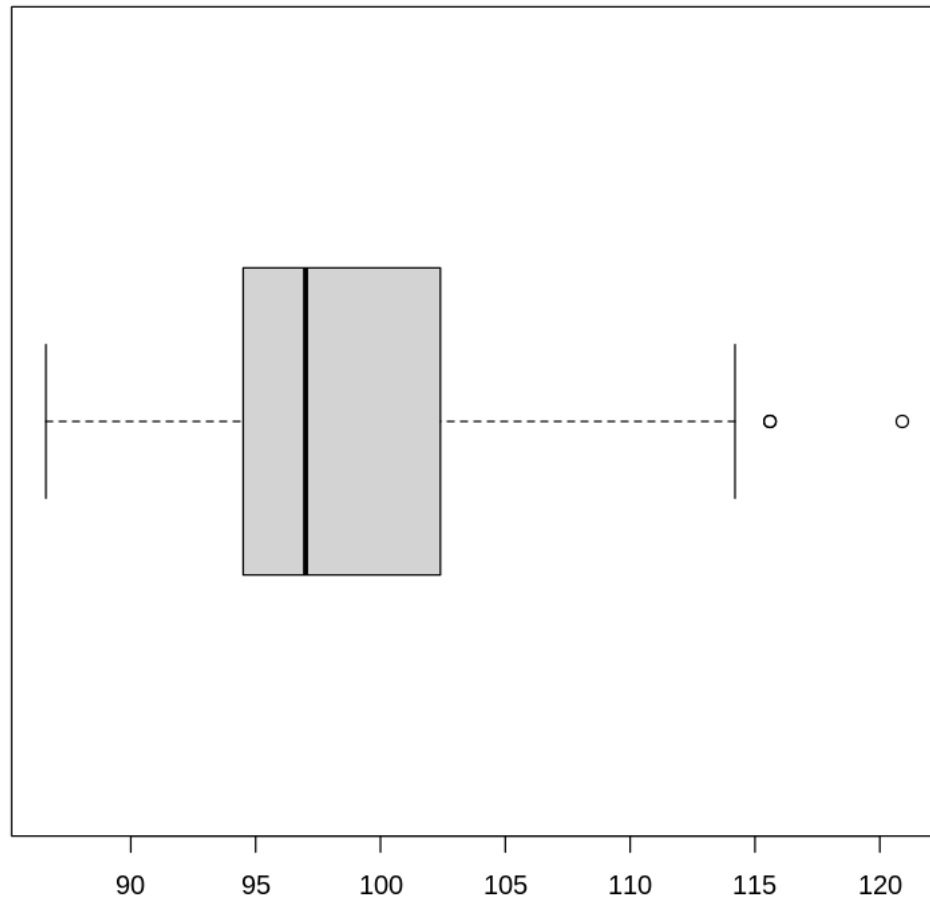
# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```

**Histograma de wheelbase**

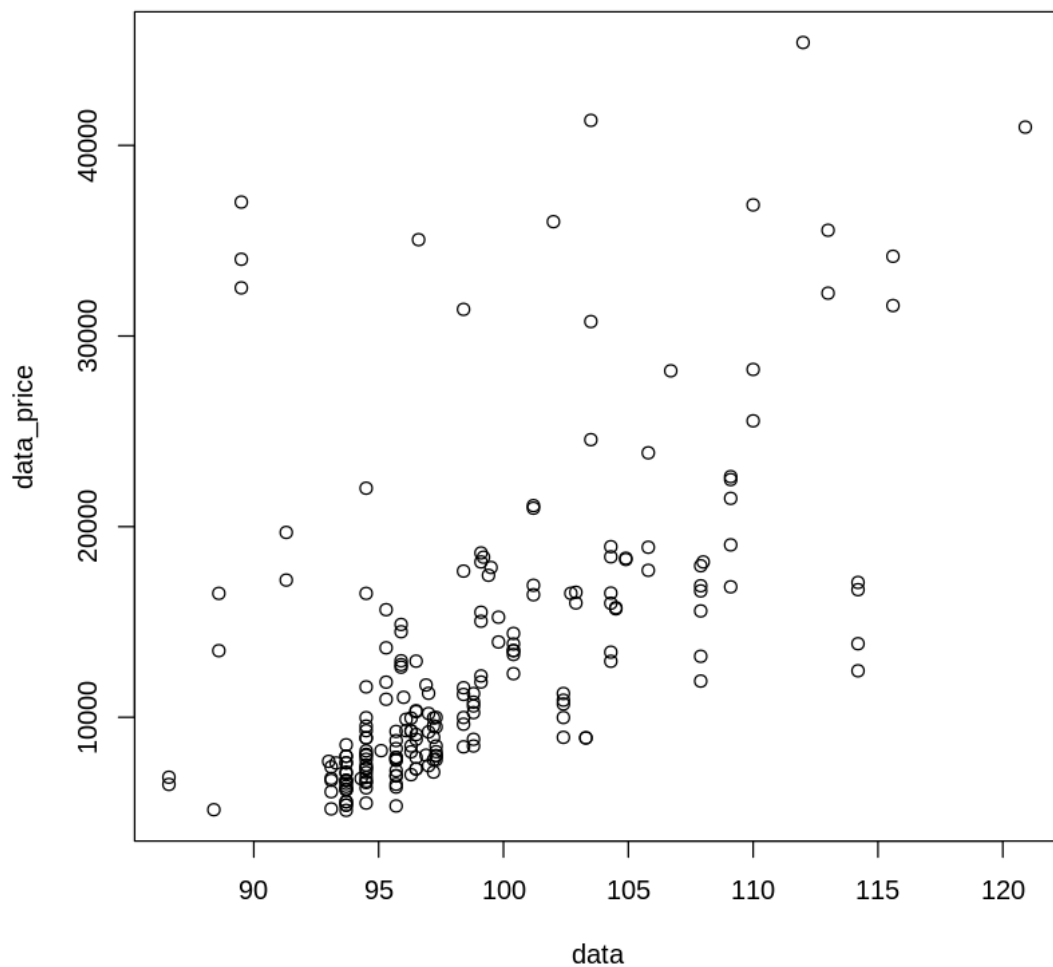


**Diagrama de dispersión de wheelbase**



Coeficiente de correlación entre wheelbase y precio: 0.5778156  
Sesgo: 1.034895

Diagrama de dispersión de wheelbase



Distribución: **Asimétrica**

Se puede observar que el histograma no se encuentra balanceado y que hay una cola más larga a la derecha, por lo que hay una asimetría positiva.

Correlación con precio: 0.57 (Si se observa una cierta relación pero podría mejorar)

###Car Length

```
[177]: variable = "carlength"
data = mydata[[variable]]

# Histograma
hist(data,col=0,main=paste("Histograma de", variable))
```

```

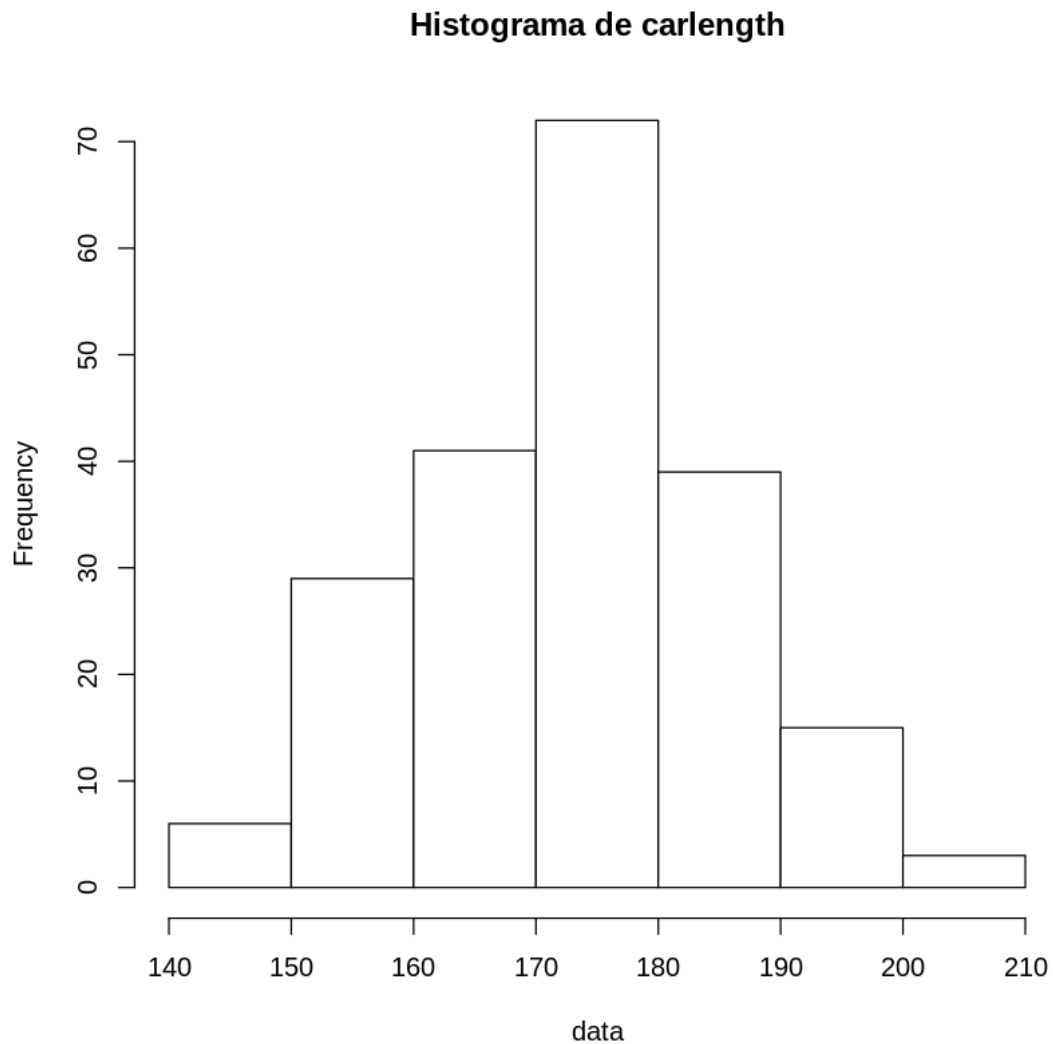
# Diagrama de caja y bigotes
boxplot(data, horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

# Diagrama de dispersión
plot(data, data_price, main=paste("Diagrama de dispersión de", variable))

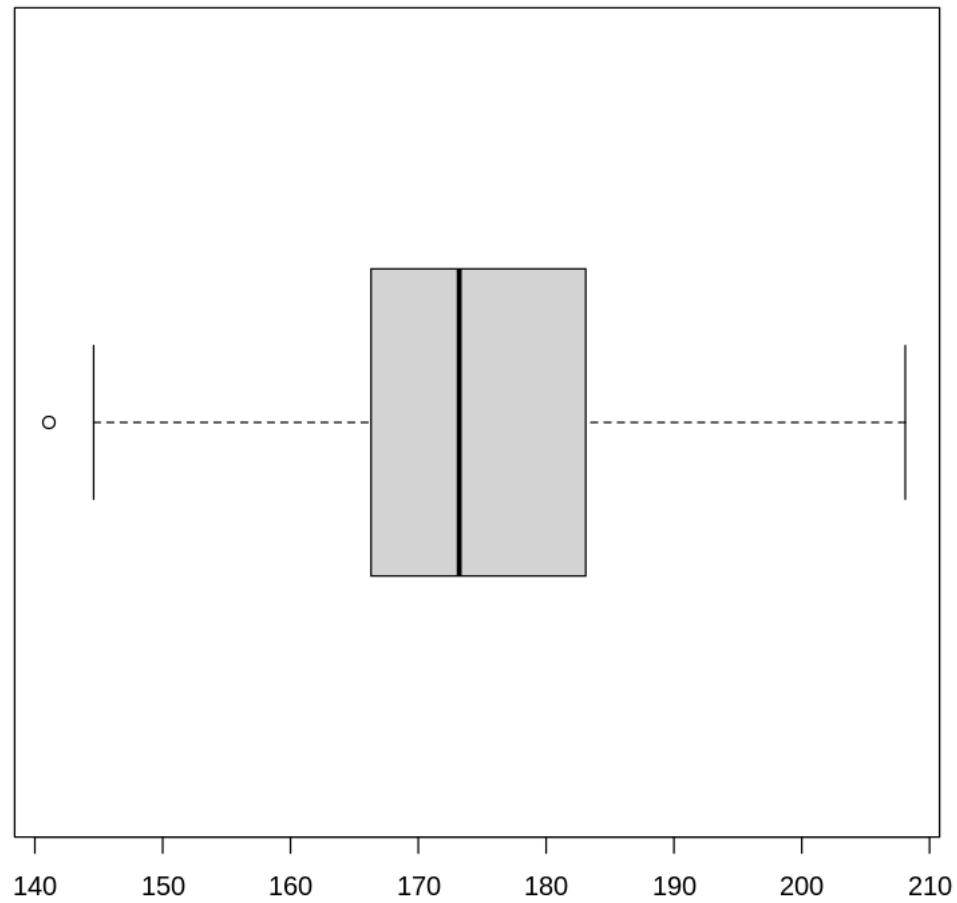
# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```

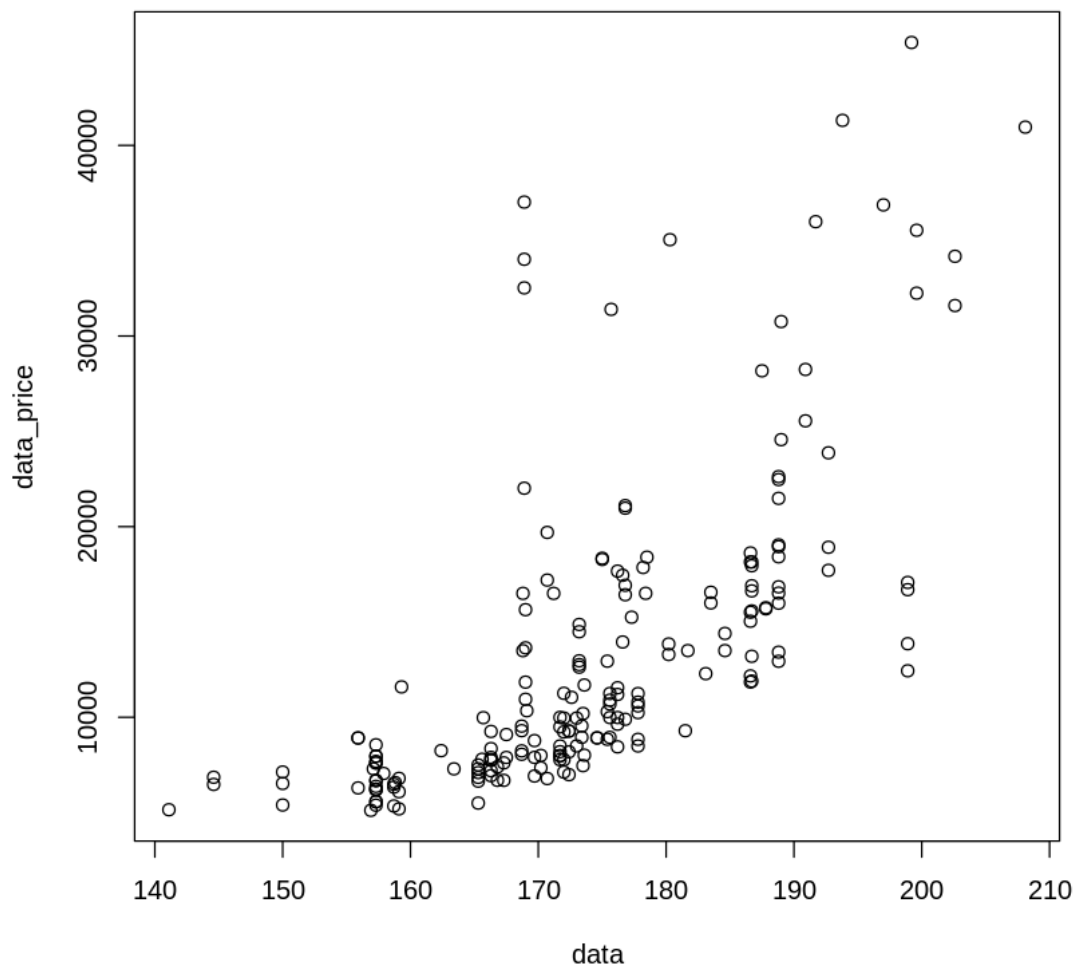


**Diagrama de dispersión de carlength**



Coeficiente de correlación entre carlength y precio: 0.68292  
Sesgo: 0.1536789

Diagrama de dispersión de carlength



Distribución: **Asimétrica**

Se puede observar que el histograma no se encuentra balanceado y que hay una cola más larga a la derecha, por lo que hay una asimetría positiva.

Correlación con precio: 0.57 (Si se observa una cierta relación pero podría mejorar)

###Car Width

```
[178]: variable = "carwidth"
data = mydata[[variable]]

# Histograma
hist(data,col=0,main=paste("Histograma de", variable))
```

```

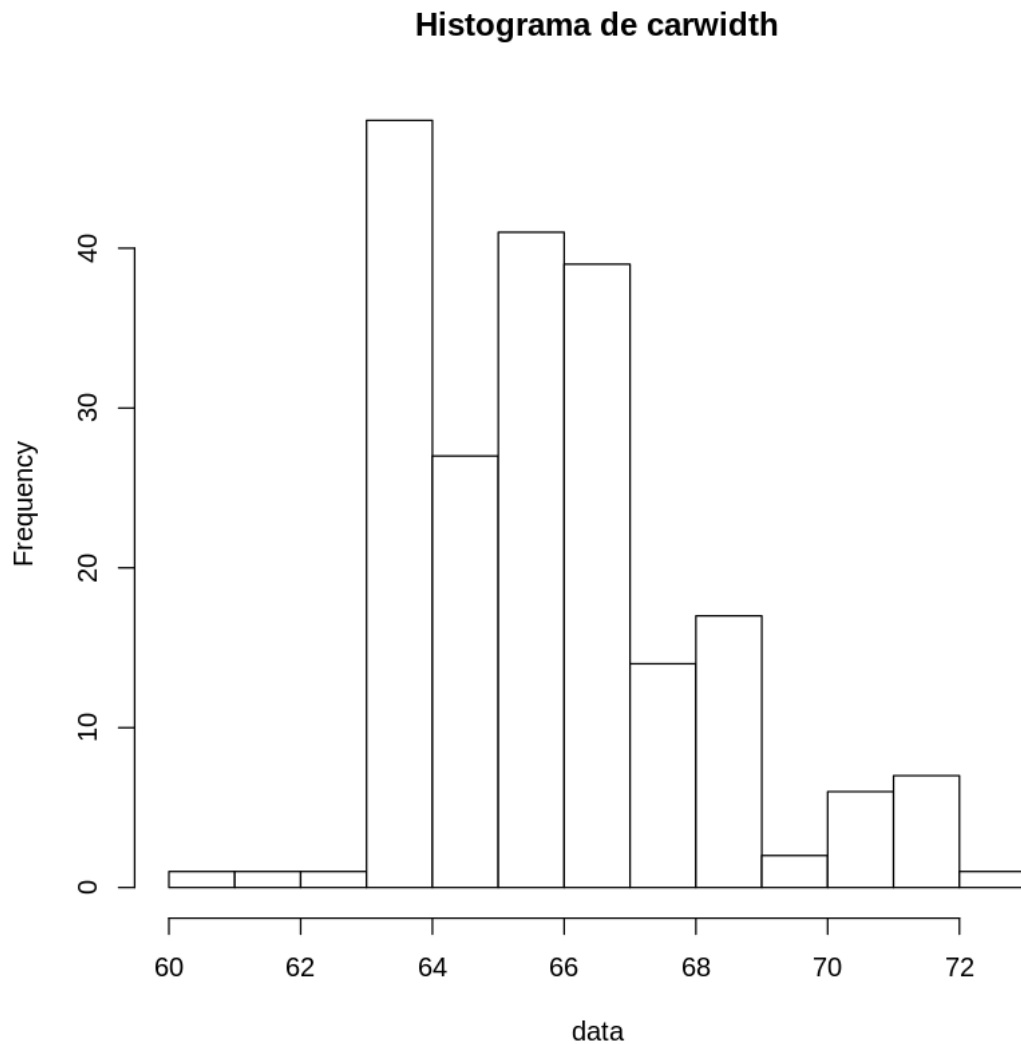
# Diagrama de caja y bigotes
boxplot(data, horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

# Diagrama de dispersión
plot(data, data_price, main=paste("Diagrama de dispersión de", variable))

# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

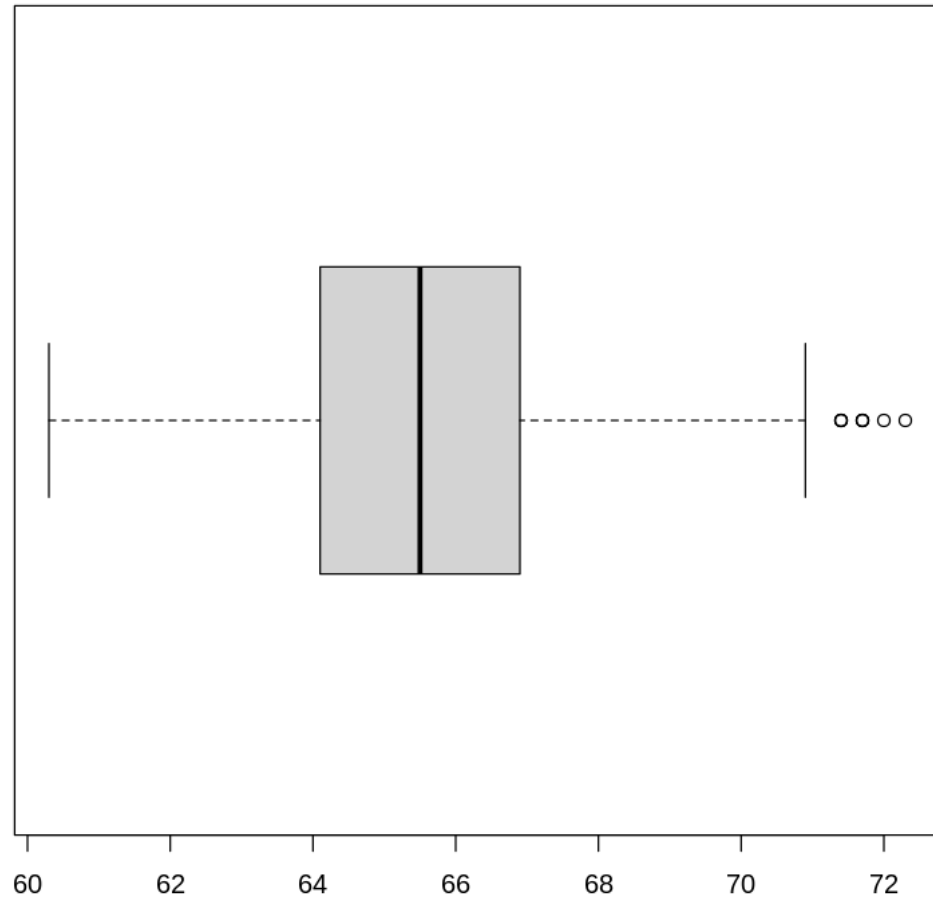
# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```



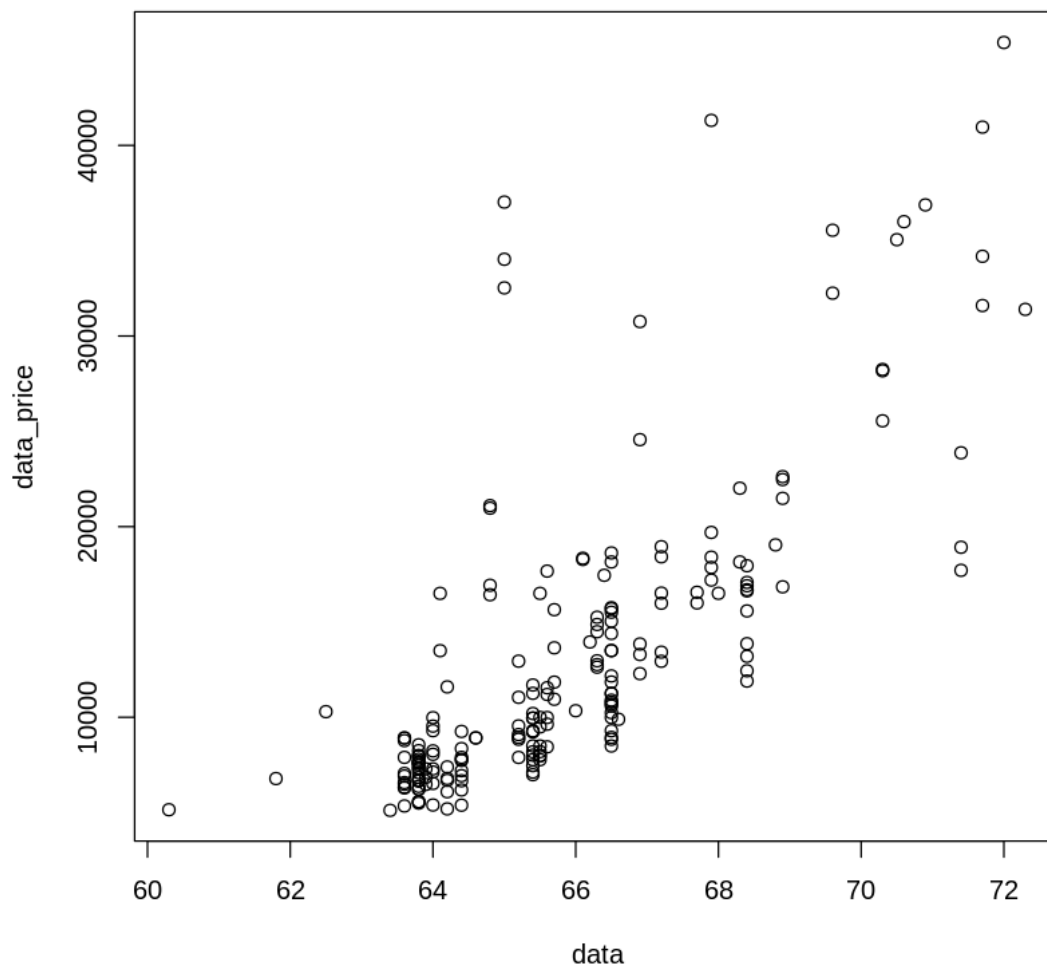


**Diagrama de dispersión de carwidth**



Coeficiente de correlación entre carwidth y precio: 0.7593253  
Sesgo: 0.8908172

Diagrama de dispersión de carwidth



Distribución: **Asimétrica**

Si bien esta dispersión se ve más parecida a una distribución normal, en realidad no lo es completamente, pues igual se ve que existe una cola más grande del lado derecho, y aunque bajó el valor de sesgo a 0.89, aún tiene asimetría positiva.

Correlación con precio: 0.75 (En este caso hay una mayor correlación entre las variables de carwidth y el precio del auto.

###Car Height

```
[179]: variable = "carheight"
      data = mydata[[variable]]

      # Histograma
```

```

hist(data,col=0,main=paste("Histograma de", variable))

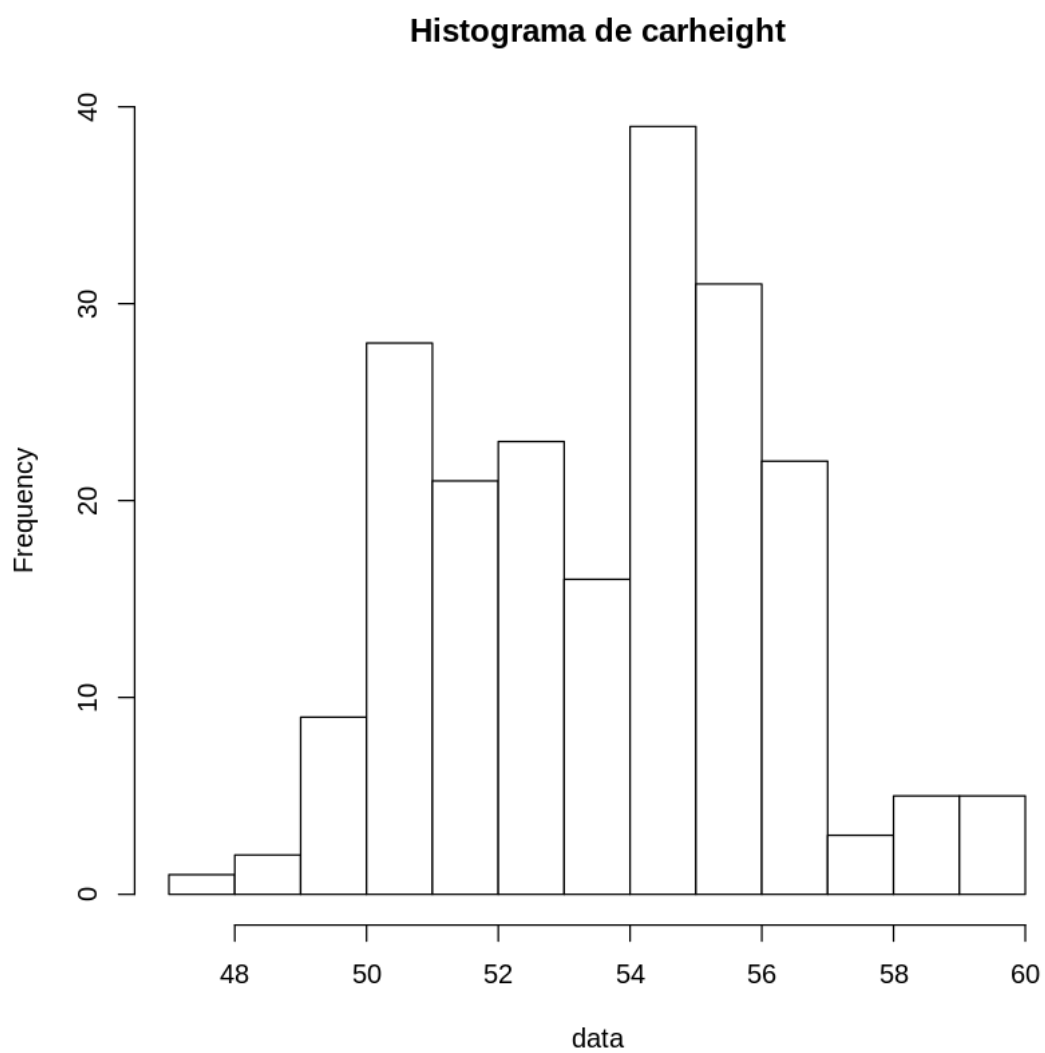
# Diagrama de caja y bigotes
boxplot(data,horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

# Diagrama de dispersión
plot(data,data_price, main=paste("Diagrama de dispersión de", variable))

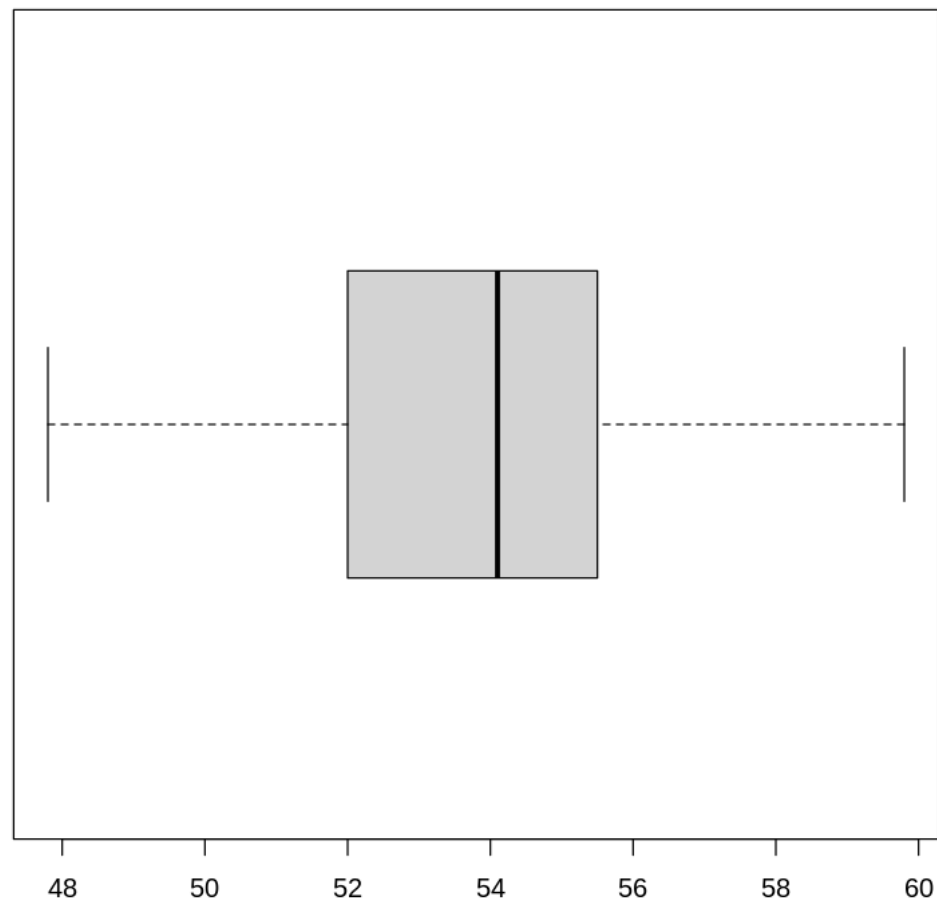
# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```

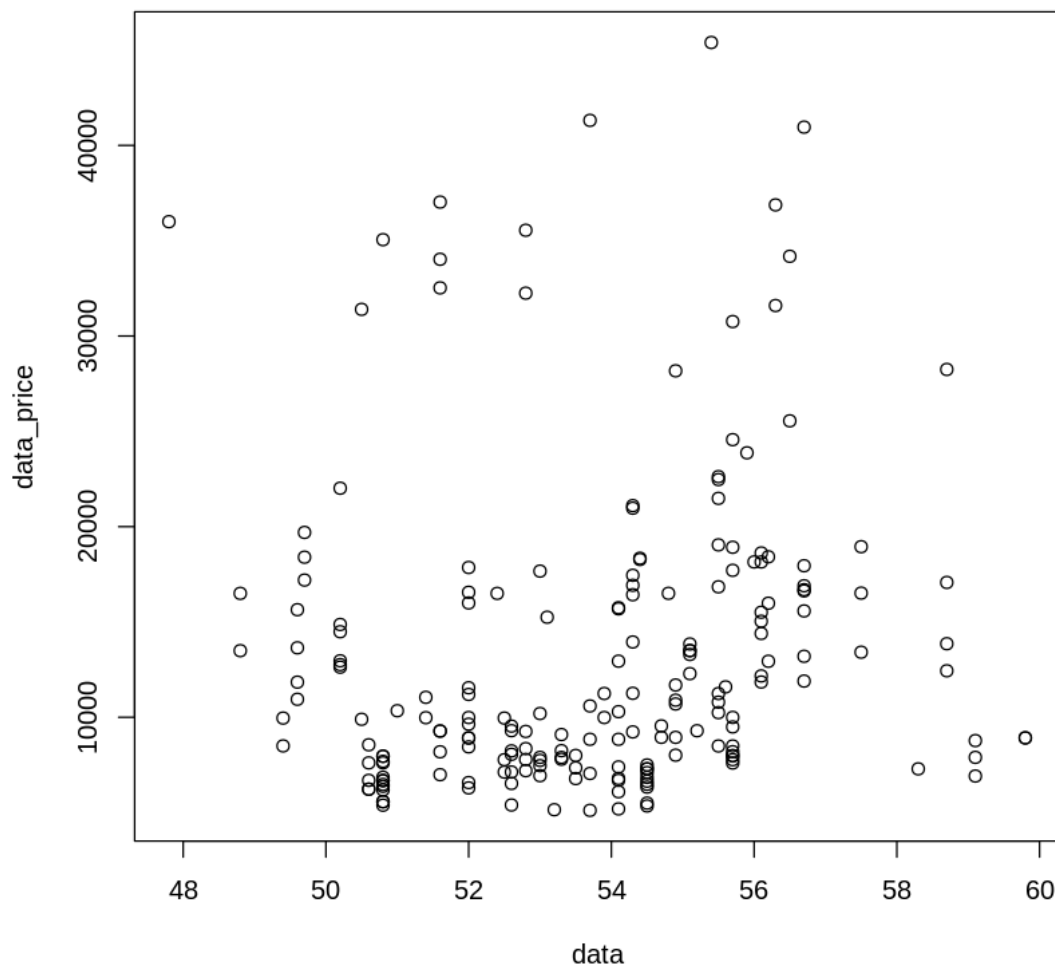


**Diagrama de dispersión de carheight**



Coeficiente de correlación entre carheight y precio: 0.1193362  
Sesgo: 0.06220199

Diagrama de dispersión de carheight



Distribución: **Simétrica**

En este caso, no se observa un espejo en el histograma, pero se puede observar que sí se encuentran distribuidos de una mejor manera. El sesgo nos indica que sí existe simetría en el conjunto de datos, pues el valor es muy cercano a 0.

Correlación con precio: 0.11 (Si bien tuvimos un gran resultado con la distribución de los datos, se puede observar en la gráfica y en el coeficiente que no hay una relación mayor entre el precio y la altura del carro, varía mucho y no sigue una tendencia.

###Curb Weight

```
[180]: variable = "curbweight"  
data = mydata[[variable]]
```

```

# Histograma
hist(data,col=0,main=paste("Histograma de", variable))

# Diagrama de caja y bigotes
boxplot(data,horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

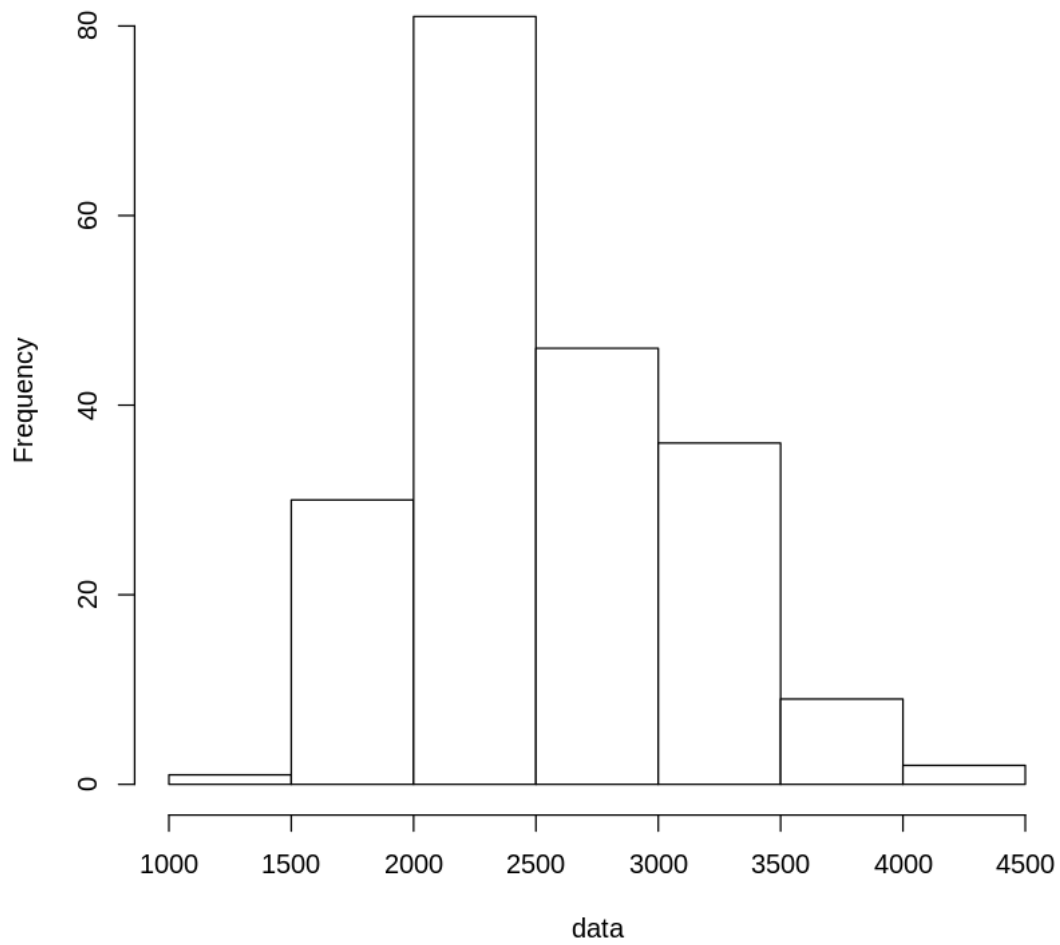
# Diagrama de dispersión
plot(data,data_price, main=paste("Diagrama de dispersión de", variable))

# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

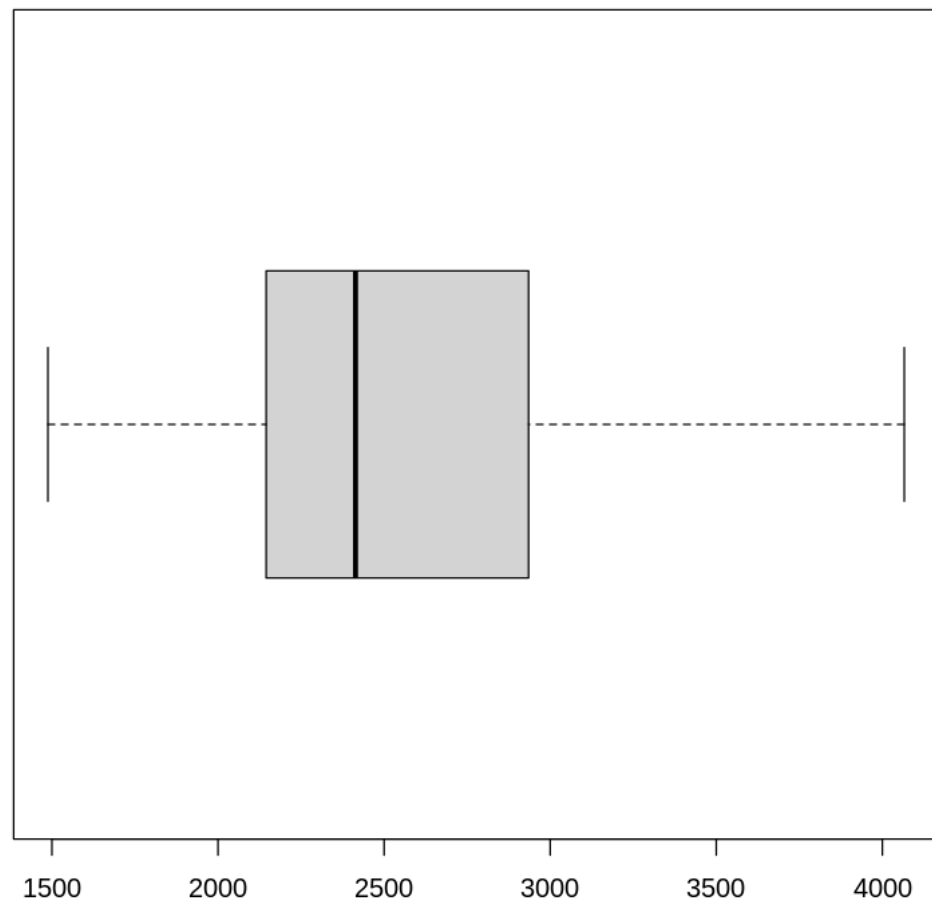
```

**Histograma de curbweight**



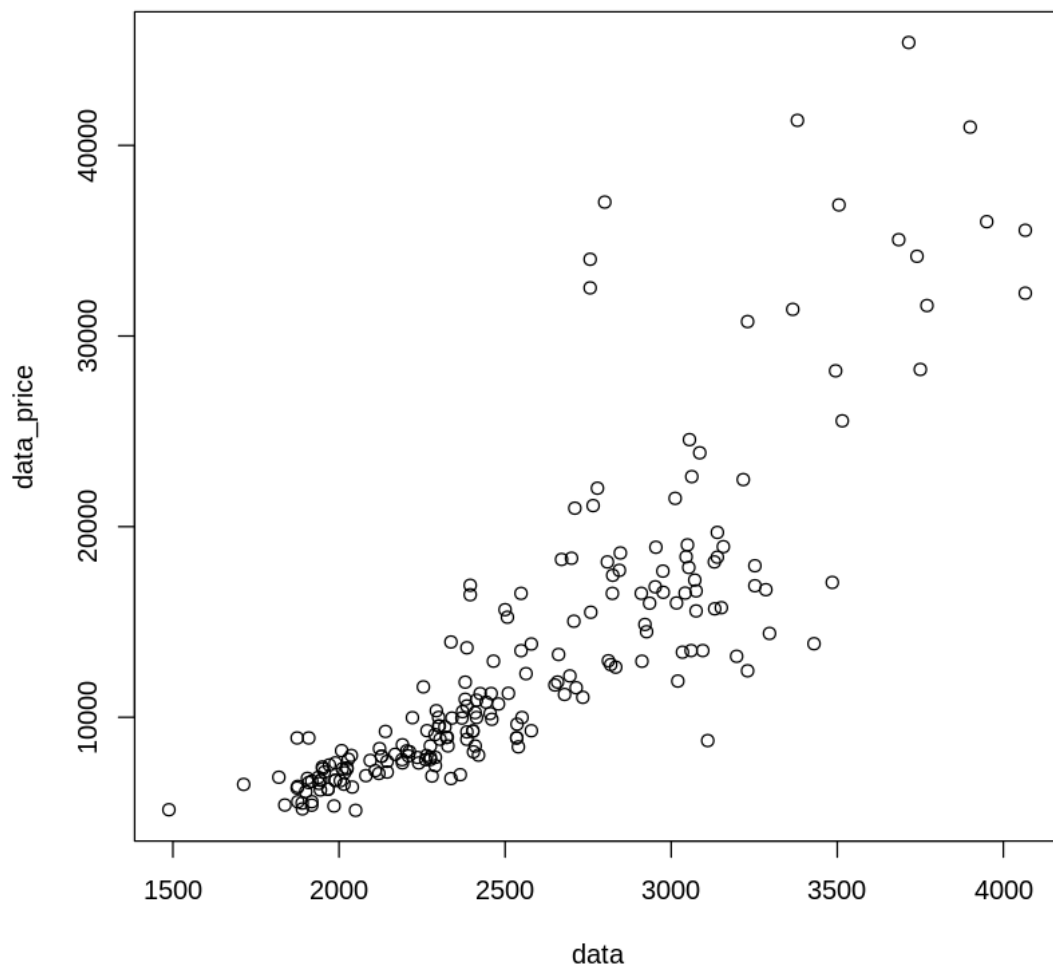


**Diagrama de dispersión de curbweight**



Coeficiente de correlación entre curbweight y precio: 0.8353049  
Sesgo: 0.6714589

Diagrama de dispersión de curbweight



Distribución: **Asimétrica**

La gráfica nos muestra directamente que existe un sesgo a la derecha, pues la cola es mucho más larga. No es asimétrica. Correlación con precio: 0.67 (Es buen resultado de relación, puede que si tengan una relación importante pero habría que investigar y revisar más.)

###Engine Size

```
[181]: variable = "enginesize"
data = mydata[[variable]]

# Histograma
hist(data,col=0,main=paste("Histograma de", variable))
```

```

# Diagrama de caja y bigotes
boxplot(data,horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

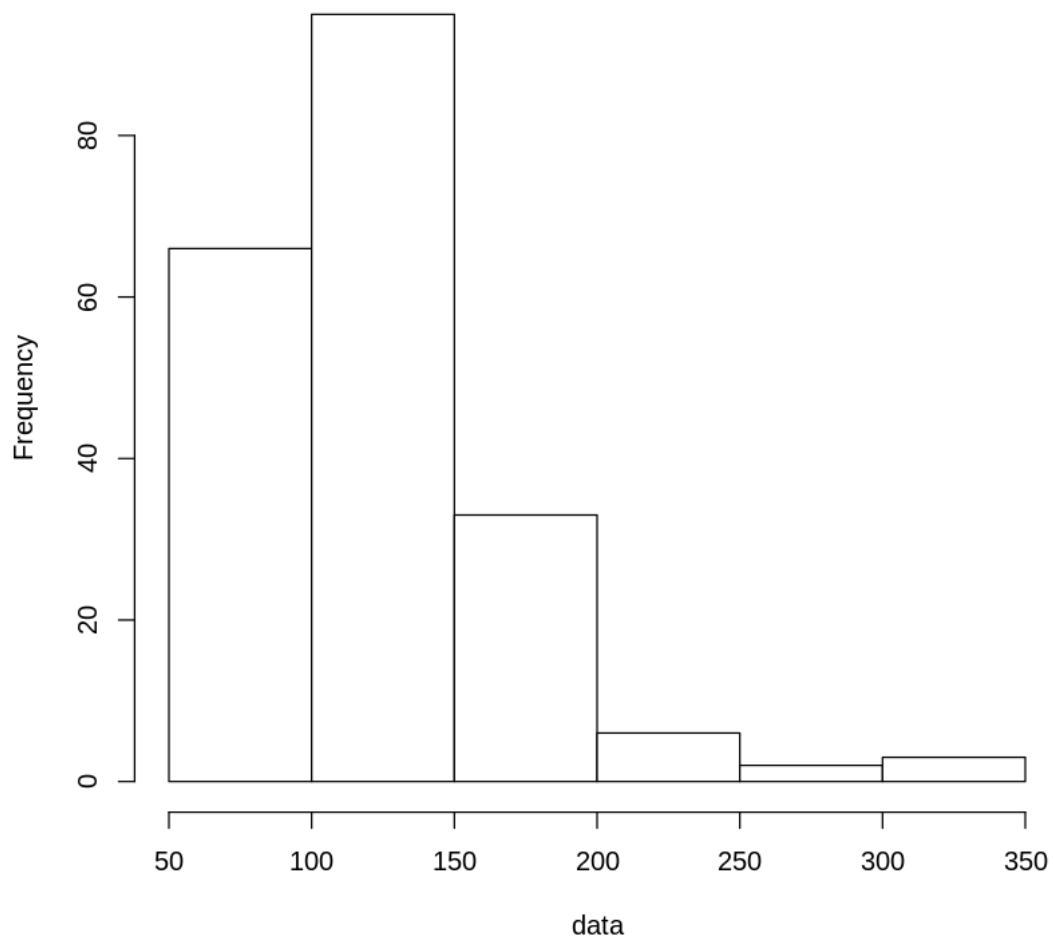
# Diagrama de dispersión
plot(data,data_price, main=paste("Diagrama de dispersión de", variable))

# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

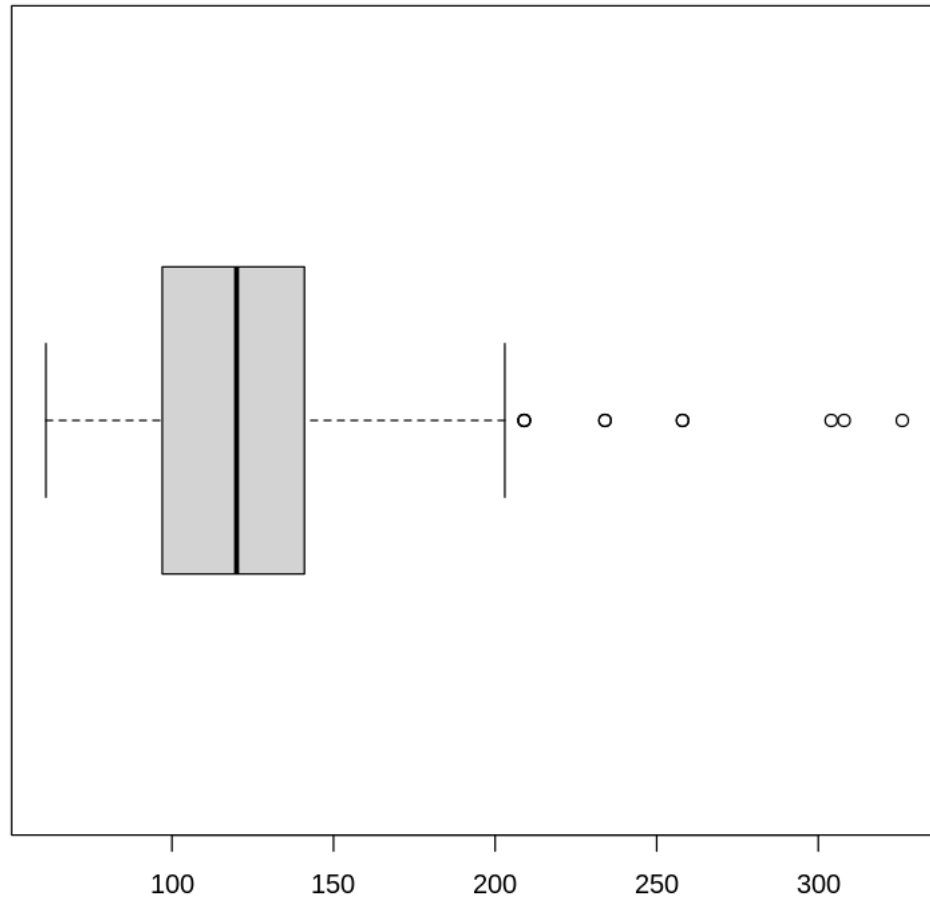
# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```

**Histograma de enginesize**

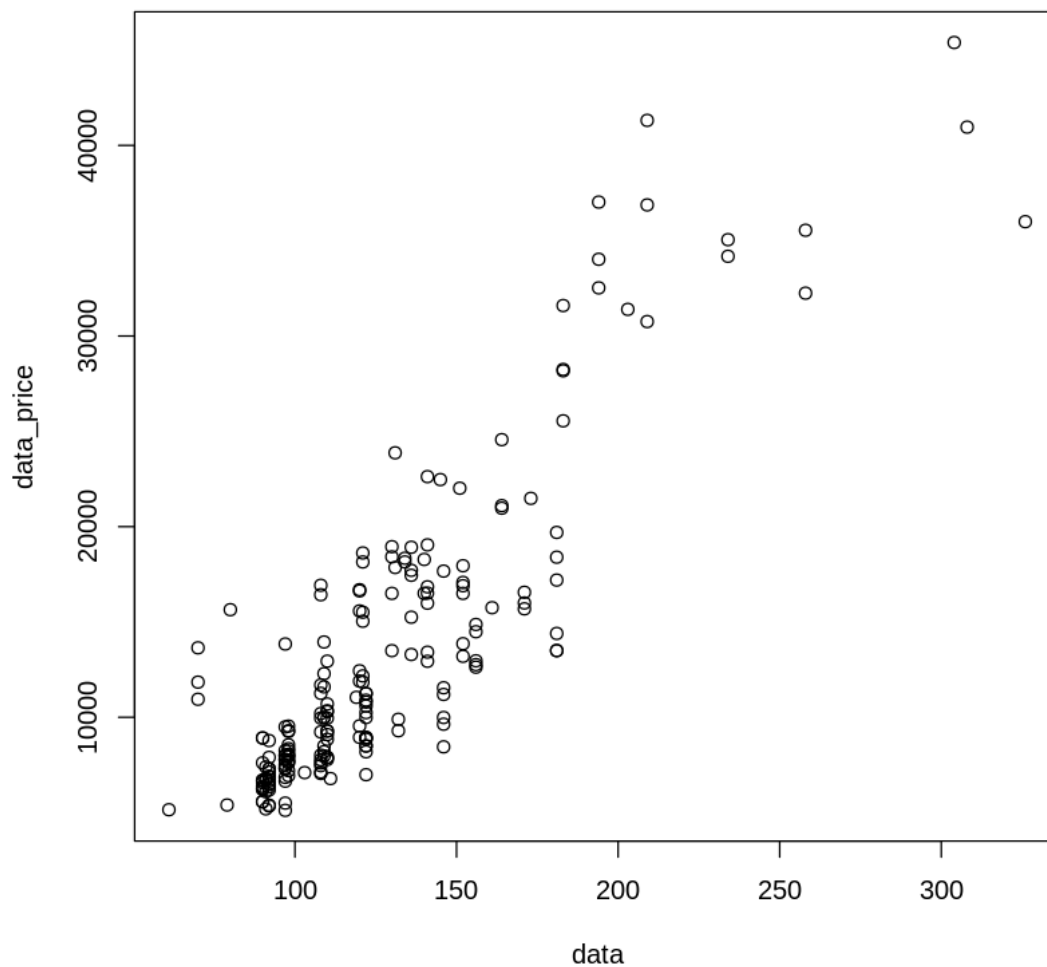


**Diagrama de dispersión de enginesize**



Coeficiente de correlación entre enginesize y precio: 0.8741448  
Sesgo: 1.919245

Diagrama de dispersión de enginesize



Distribución: **Asimétrica**

La gráfica nos muestra directamente que existe un sesgo a la derecha, pues la cola es más larga de la derecha. Sin embargo, en este caso sería bueno intentar quitar los valores de outliers, para ver el efecto en la dispersión de datos.

Correlación con precio: 0.87 (Es un increíble valor de correlación, al parecer es muy probable que mientras mayor sea el tamaño del motor, mayor el precio también).

###Stroke

```
[182]: variable = "stroke"
      data = mydata[[variable]]
```

```
# Histograma
```

```

hist(data,col=0,main=paste("Histograma de", variable))

# Diagrama de caja y bigotes
boxplot(data,horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

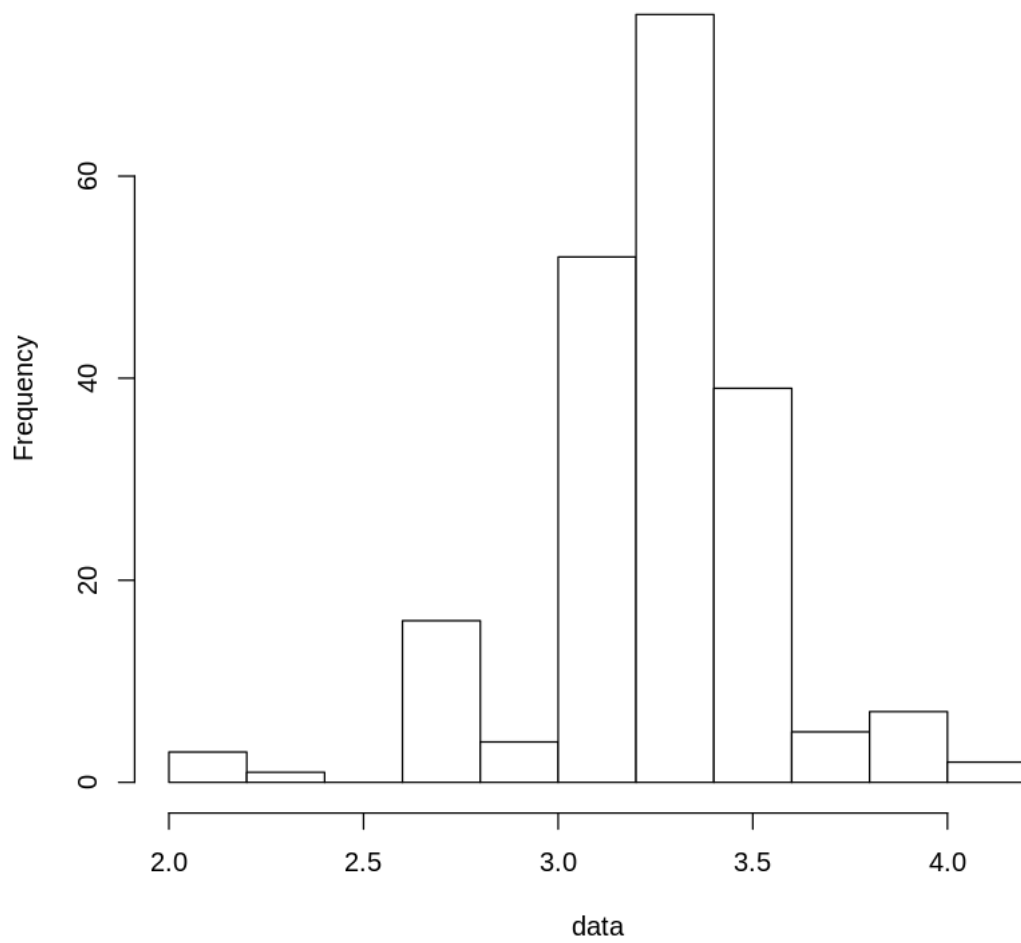
# Diagrama de dispersión
plot(data,data_price, main=paste("Diagrama de dispersión de", variable))

# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

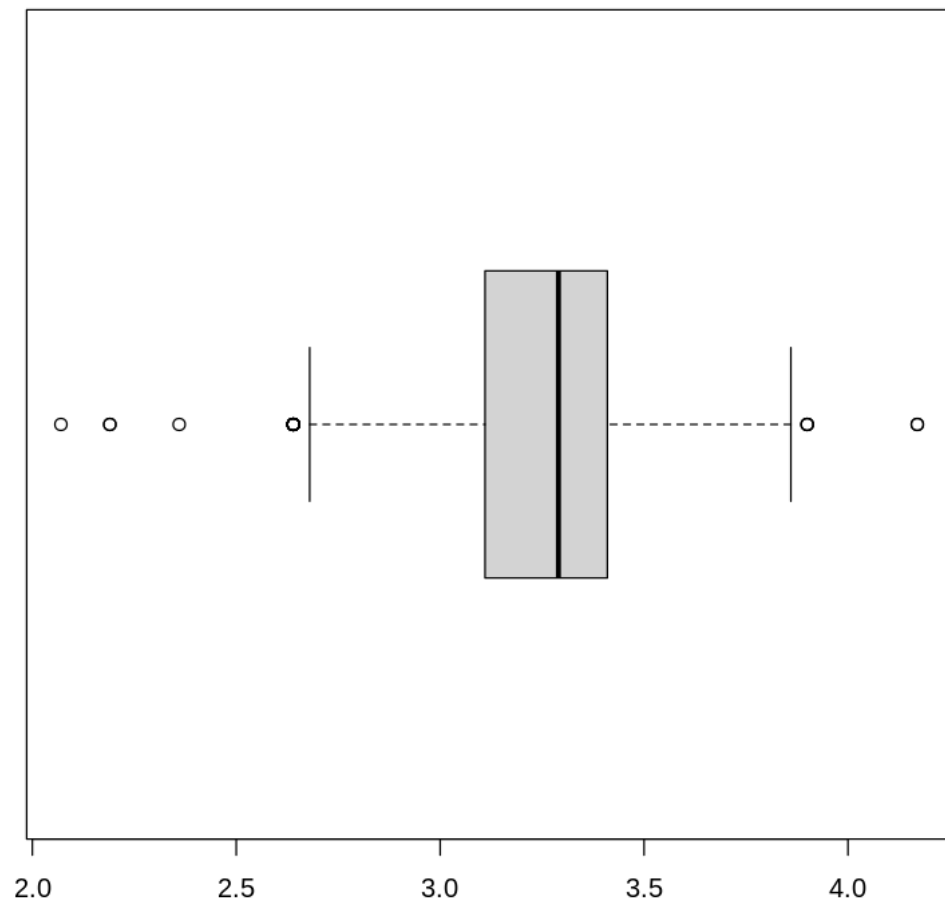
# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```

## Histograma de stroke



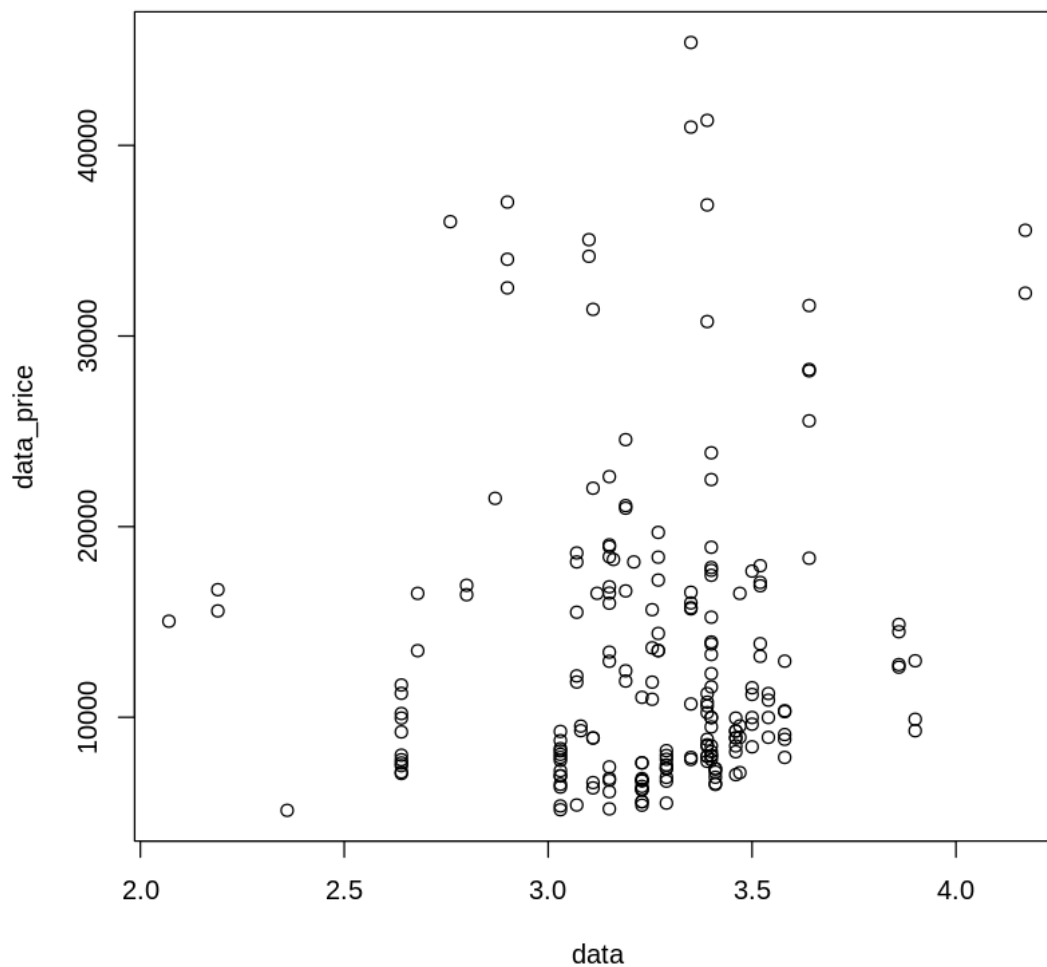
**Diagrama de dispersión de stroke**



Coeficiente de correlación entre stroke y precio: 0.07944308  
Sesgo: -0.6796442



Diagrama de dispersión de stroke



Distribución: **Asimétrica**

La gráfica del histograma nos indica que existe asimetría hacia la izquierda, pues está más grande esa cola, además de que el sesgo es negativo.

Correlación con precio: 0.07 No existe una correlación entre las variables que tenga una evidencia suficiente.

###Compression Ratio

```
[183]: variable = "compressionratio"
data = mydata[[variable]]

# Histograma
hist(data,col=0,main=paste("Histograma de", variable))
```

```

# Diagrama de caja y bigotes
boxplot(data, horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

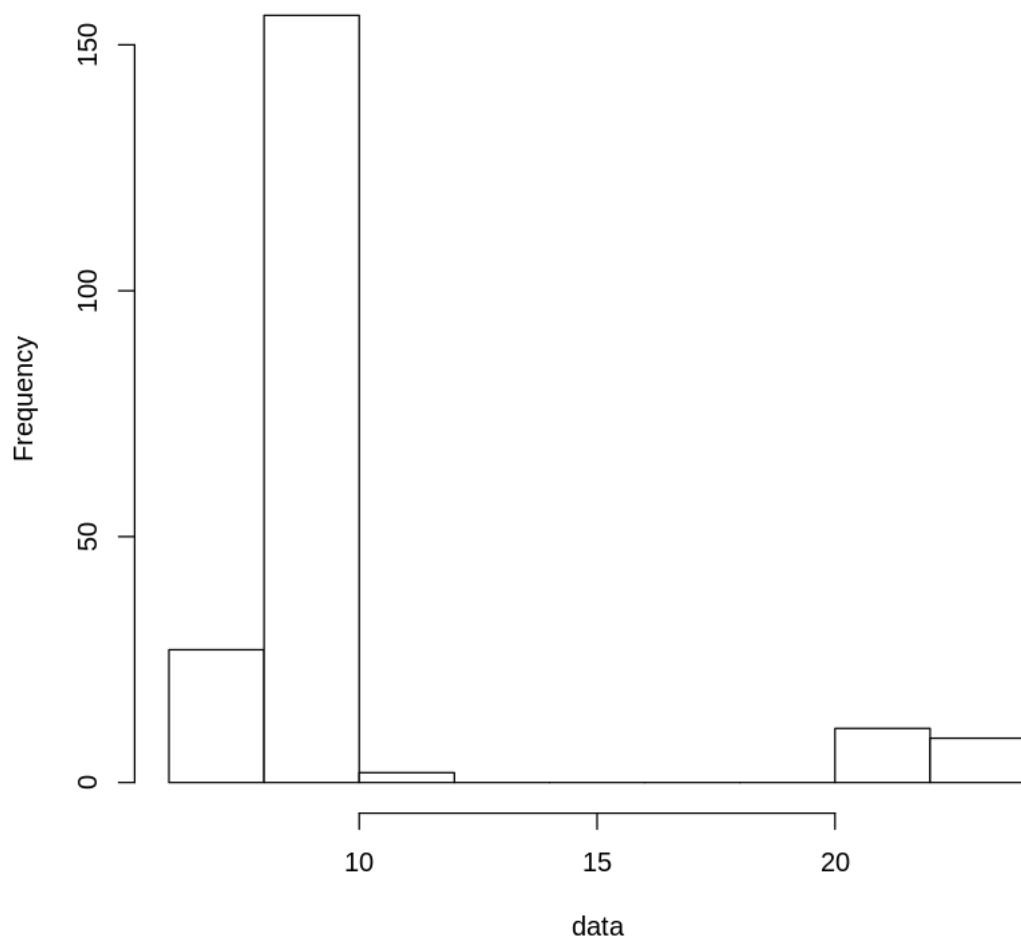
# Diagrama de dispersión
plot(data, data_price, main=paste("Diagrama de dispersión de", variable))

# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

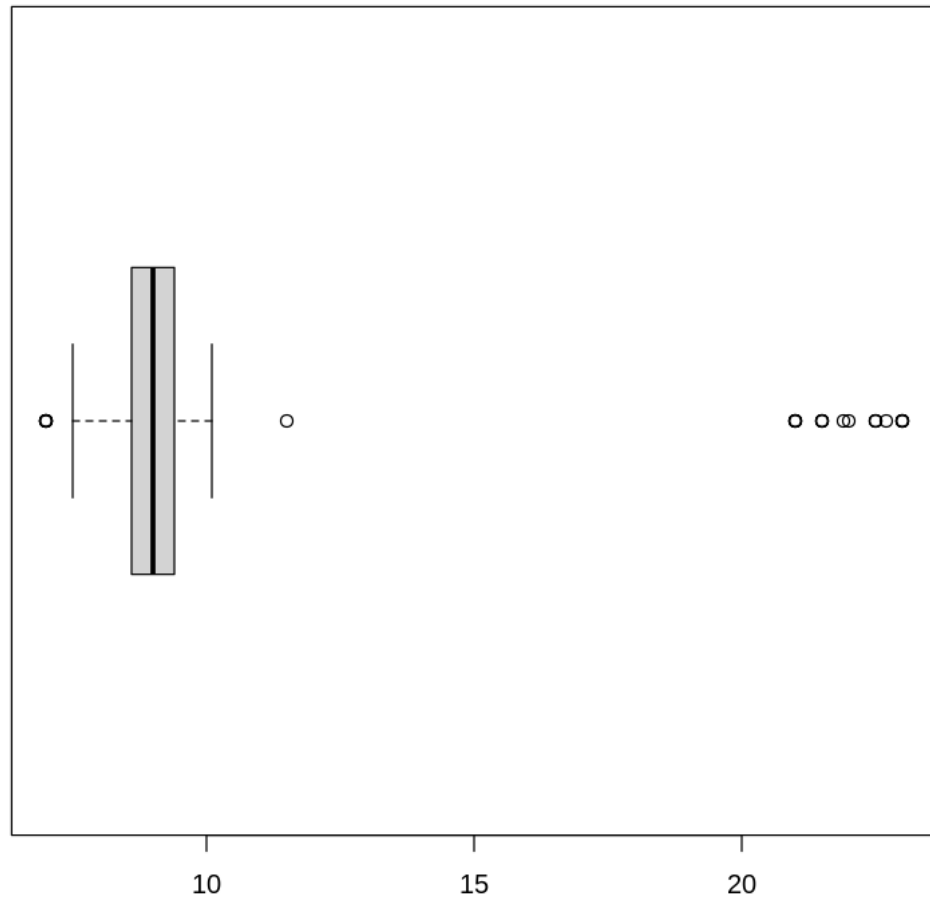
# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```

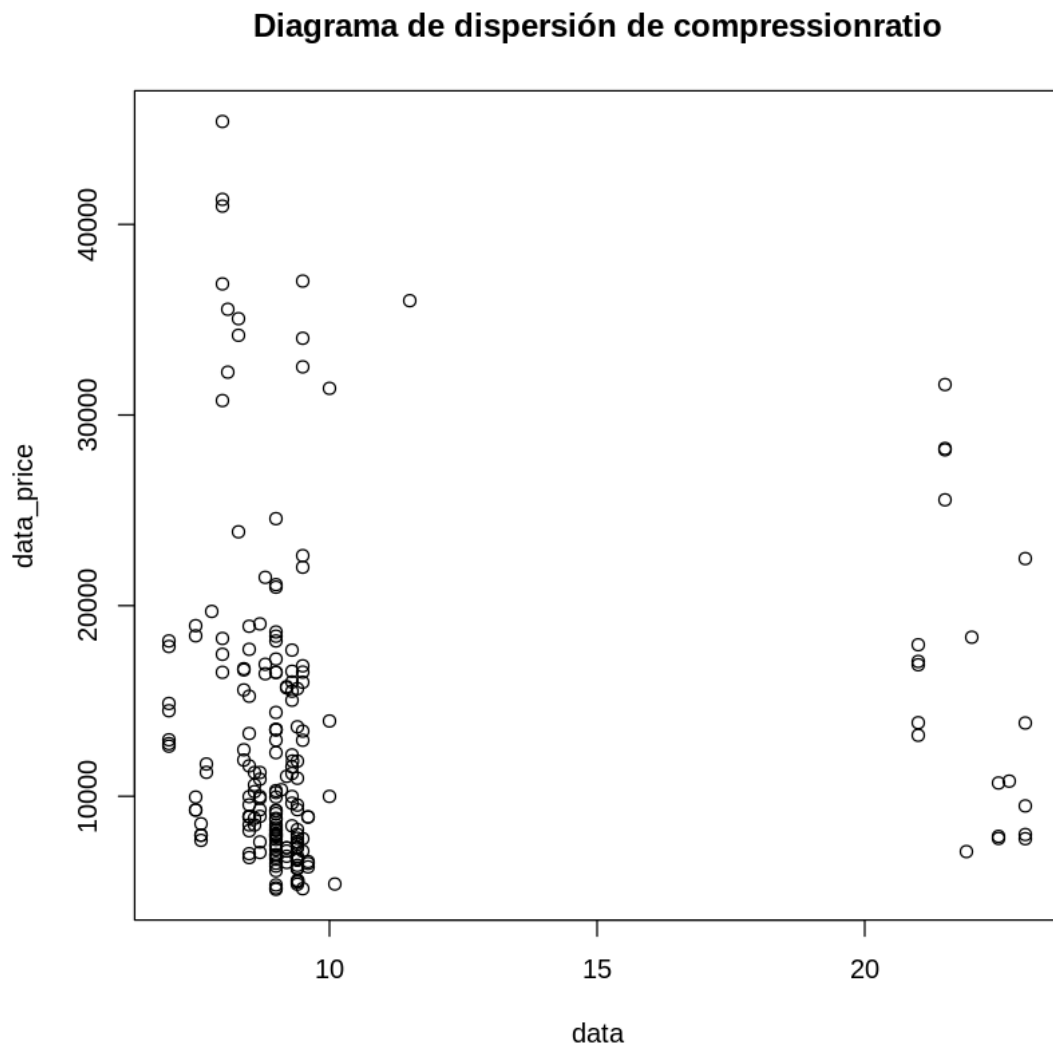
### Histograma de compressionratio



**Diagrama de dispersión de compressionratio**



Coeficiente de correlación entre compressionratio y precio: 0.06798351  
Sesgo: 2.572779



Distribución: **Asimétrica**

La mayoría de los datos se encuentran a la derecha y en un rango entre 0 y 15, mientras que los demás pcoos ya se pueden considerar outliers, y están del otro extremo. No es una distribución normal, habría que probar eliminando outliers si se selecciona.

Correlación con precio: 0.06, no es un buen coeficiente de correlación, por lo que se puede decir que no existe una relación entre ambos probablemente.

###Horsepower

```
[184]: variable = "horsepower"
      data = mydata[[variable]]
```

```
# Histograma
```

```

hist(data,col=0,main=paste("Histograma de", variable))

# Diagrama de caja y bigotes
boxplot(data,horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

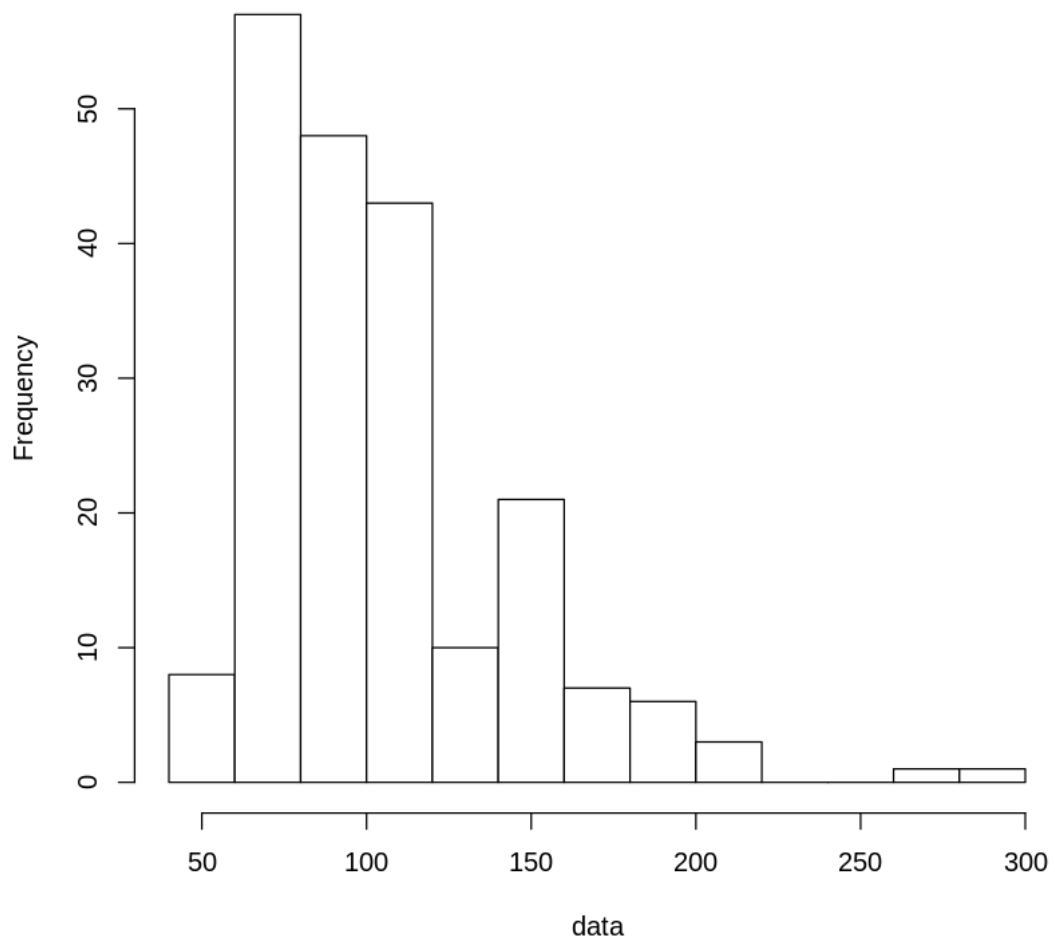
# Diagrama de dispersión
plot(data,data_price, main=paste("Diagrama de dispersión de", variable))

# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

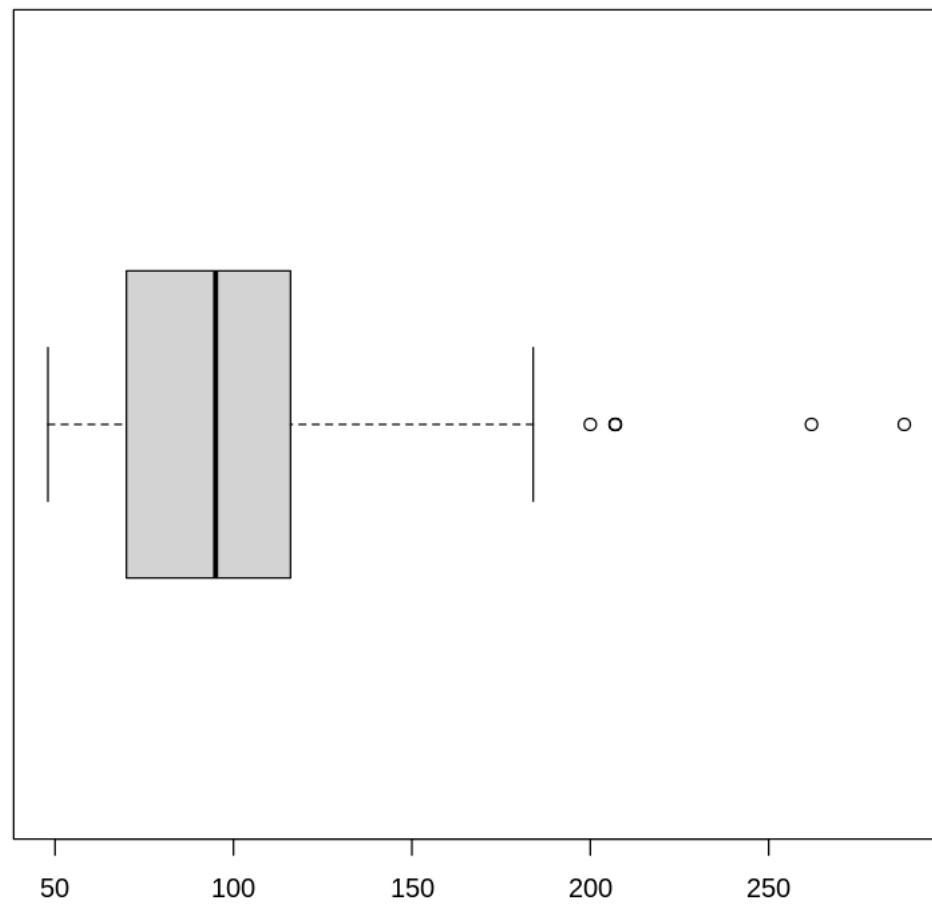
# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```

**Histograma de horsepower**

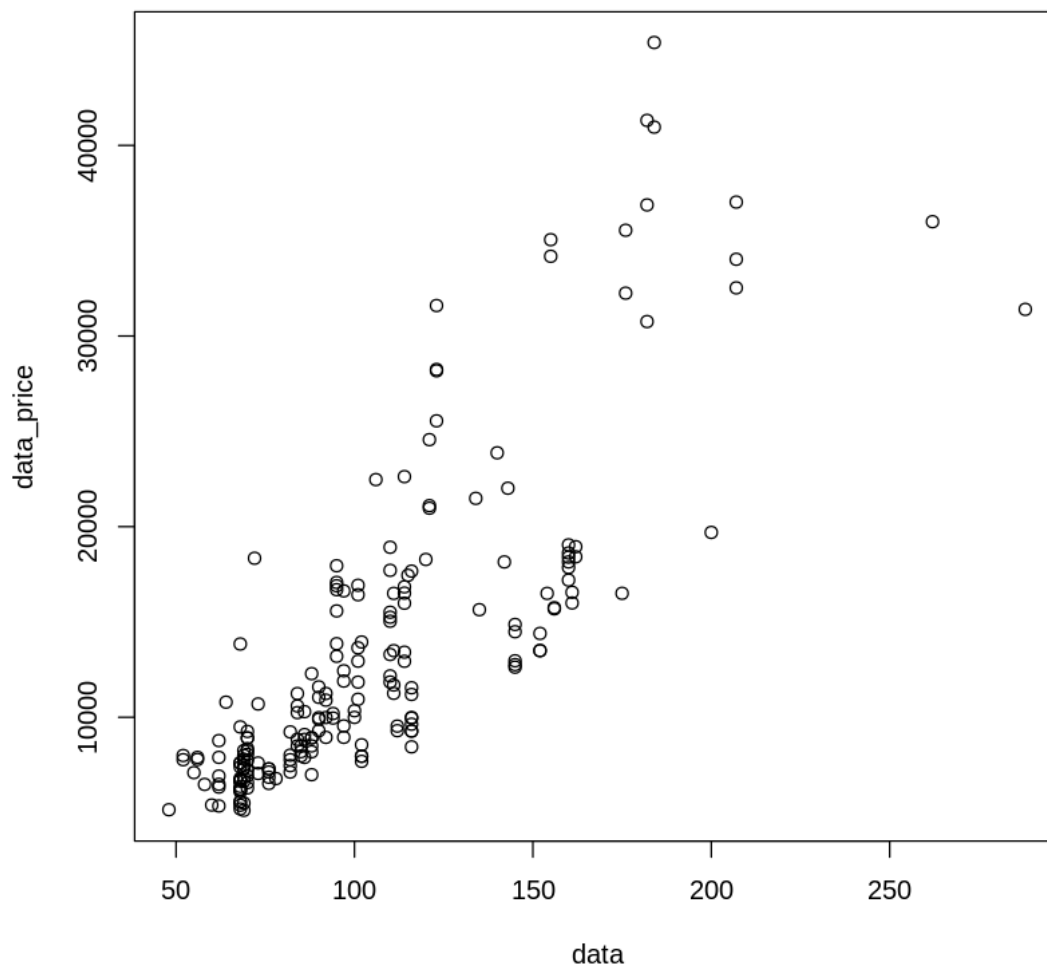


**Diagrama de dispersión de horsepower**



Coeficiente de correlación entre horsepower y precio: 0.8081388  
Sesgo: 1.384812

Diagrama de dispersión de horsepower



Distribución: **Asimétrica**

La gráfica del histograma nos indica que existe asimetría hacia la derecha, pues está más grande esa cola, además de que el sesgo es positivo y es un valor alto.

Correlación con precio: 0.80 (Es un valor alto de correlación entre las variables, aunque se ve que a partir de los 200, ya hay muy pocos datos, pero los demás se nota que sí tienen una relación directa con el precio.

###Peak rpm

```
[185]: variable = "peakrpm"  
data = mydata[[variable]]
```

```
# Histograma
```



```

hist(data,col=0,main=paste("Histograma de", variable))

# Diagrama de caja y bigotes
boxplot(data,horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

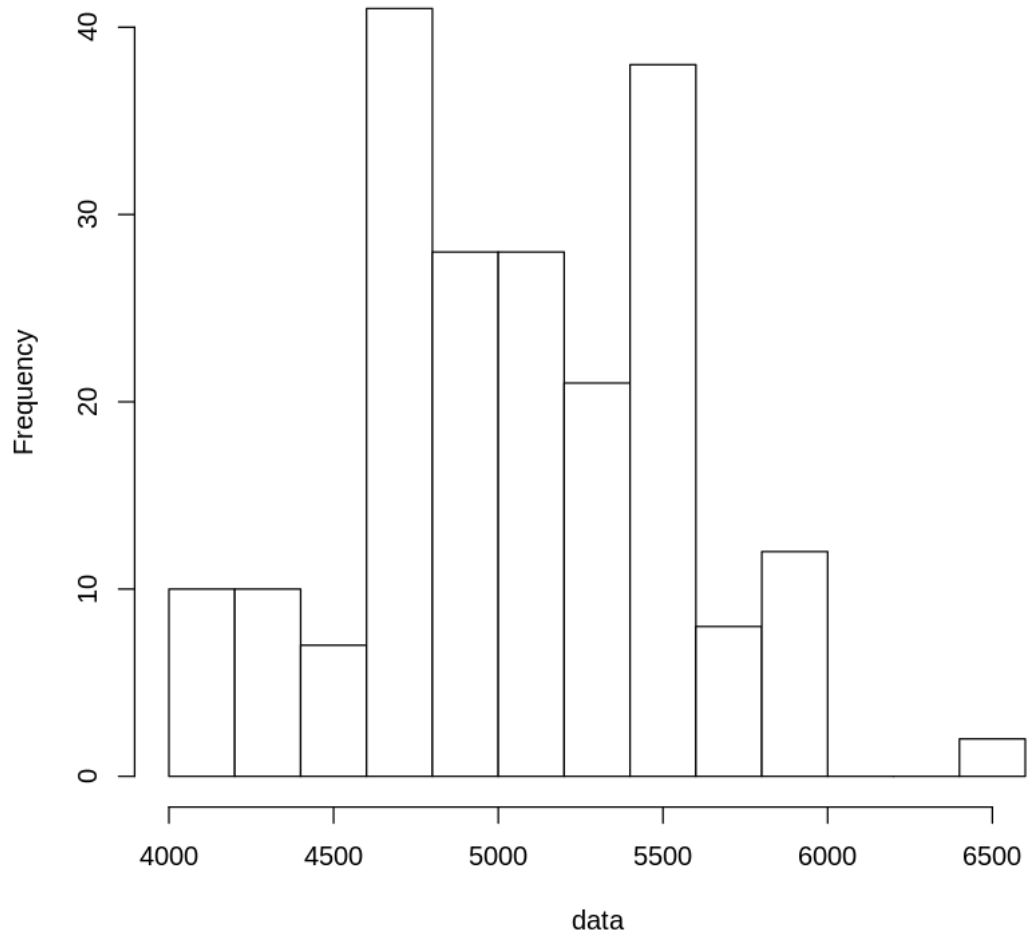
# Diagrama de dispersión
plot(data,data_price, main=paste("Diagrama de dispersión de", variable))

# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

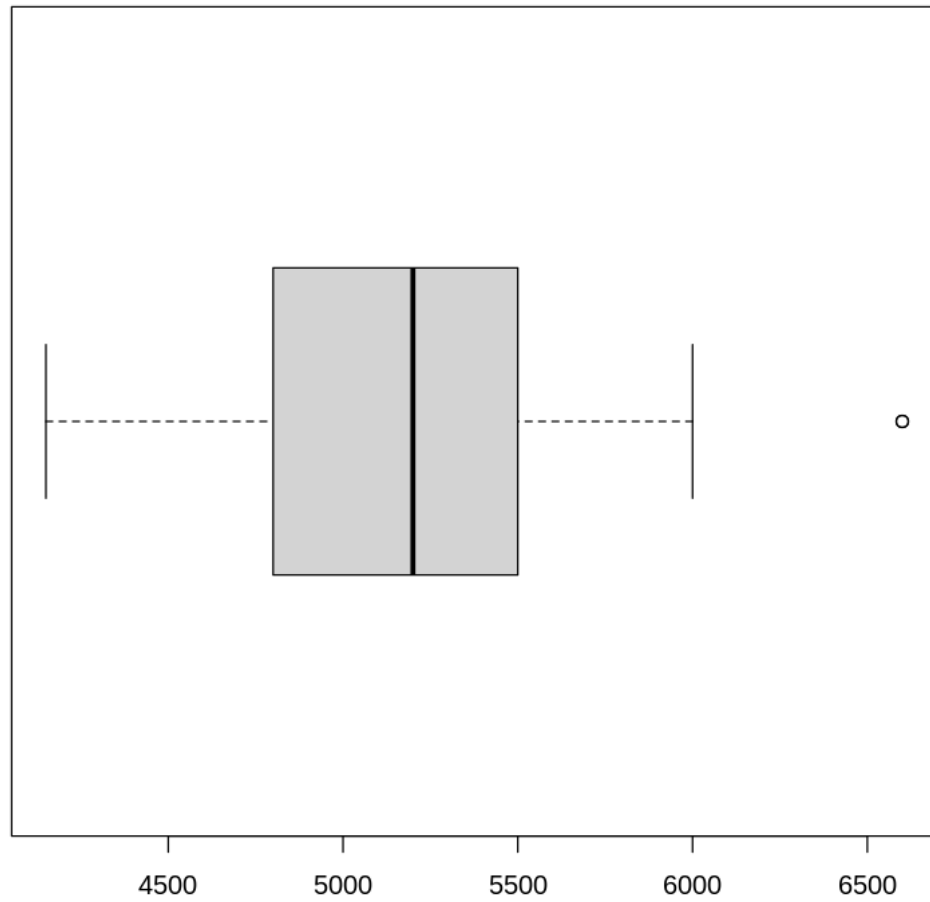
# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```

**Histograma de peakrpm**



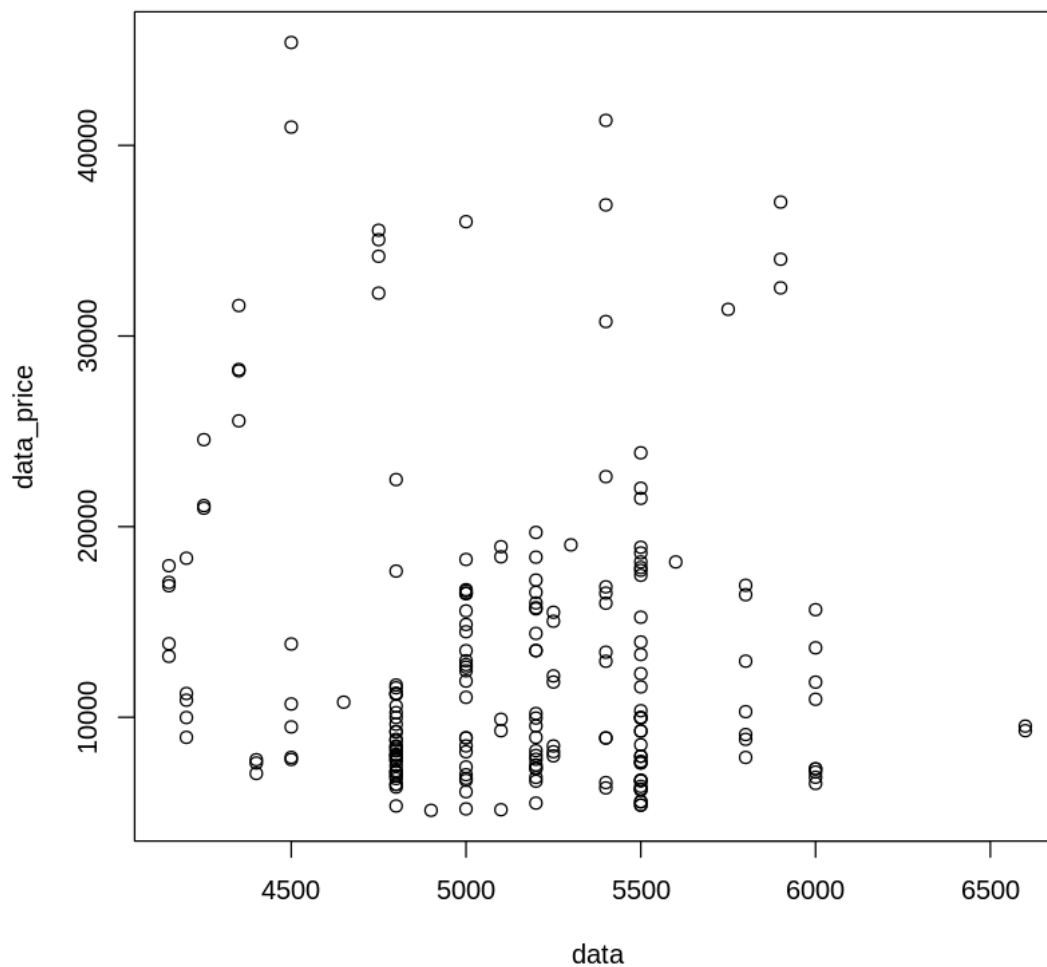
**Diagrama de dispersión de peakrpm**



Coeficiente de correlación entre peakrpm y precio: -0.08526715

Sesgo: 0.07406242

Diagrama de dispersión de peakrpm



Distribución: **Simétrica**

Se puede observar que sí hay de cierta forma una distribución normal en la gráfica, pues se ve parecida la cantidad de datos en ambos lados, y además, el sesgo dio un valor muy pequeño.

Si bien tiene distribución normal, no tiene una relación significativa, pues el coeficiente de correlación fue de -0.08.

###City mpg

```
[186]: variable = "citympg"
data = mydata[[variable]]

# Histograma
hist(data,col=0,main=paste("Histograma de", variable))
```

```

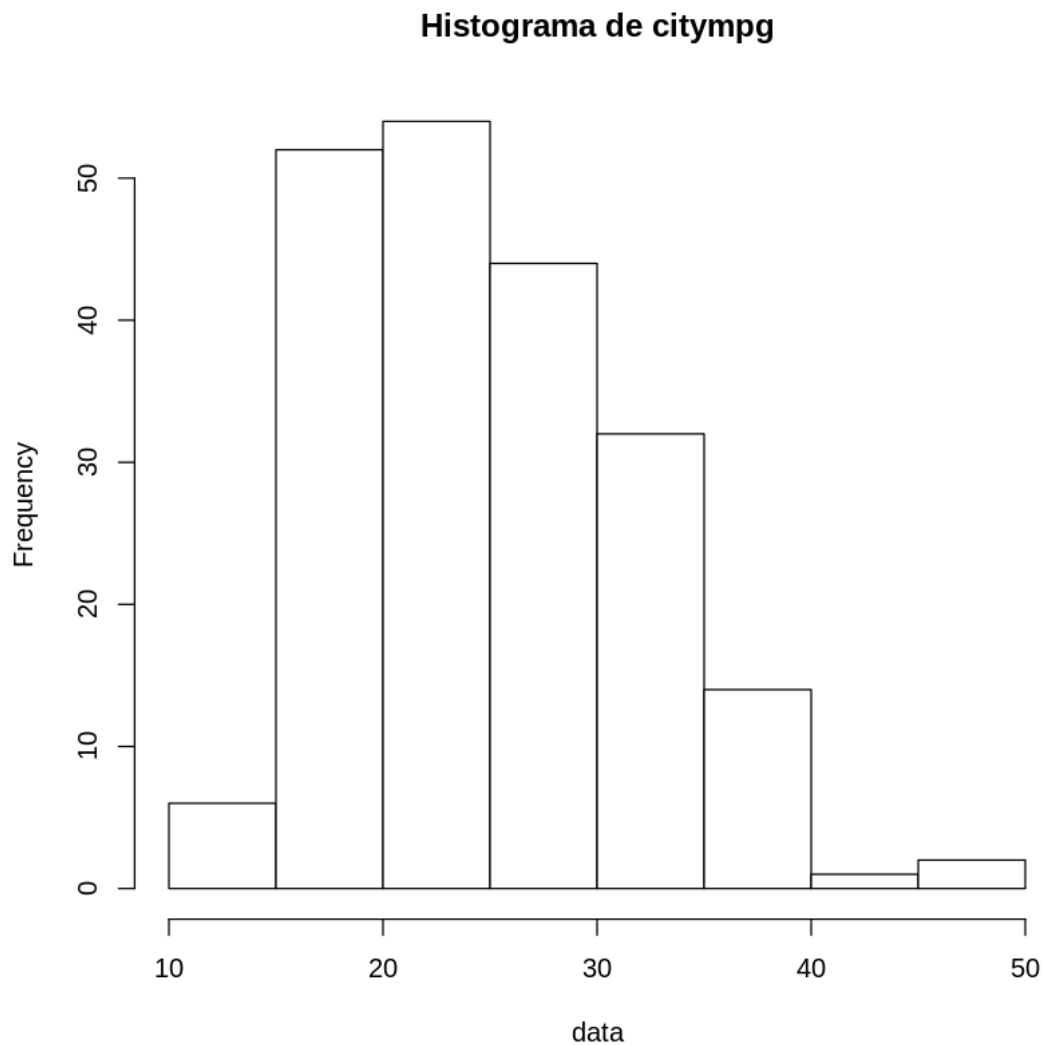
# Diagrama de caja y bigotes
boxplot(data, horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

# Diagrama de dispersión
plot(data, data_price, main=paste("Diagrama de dispersión de", variable))

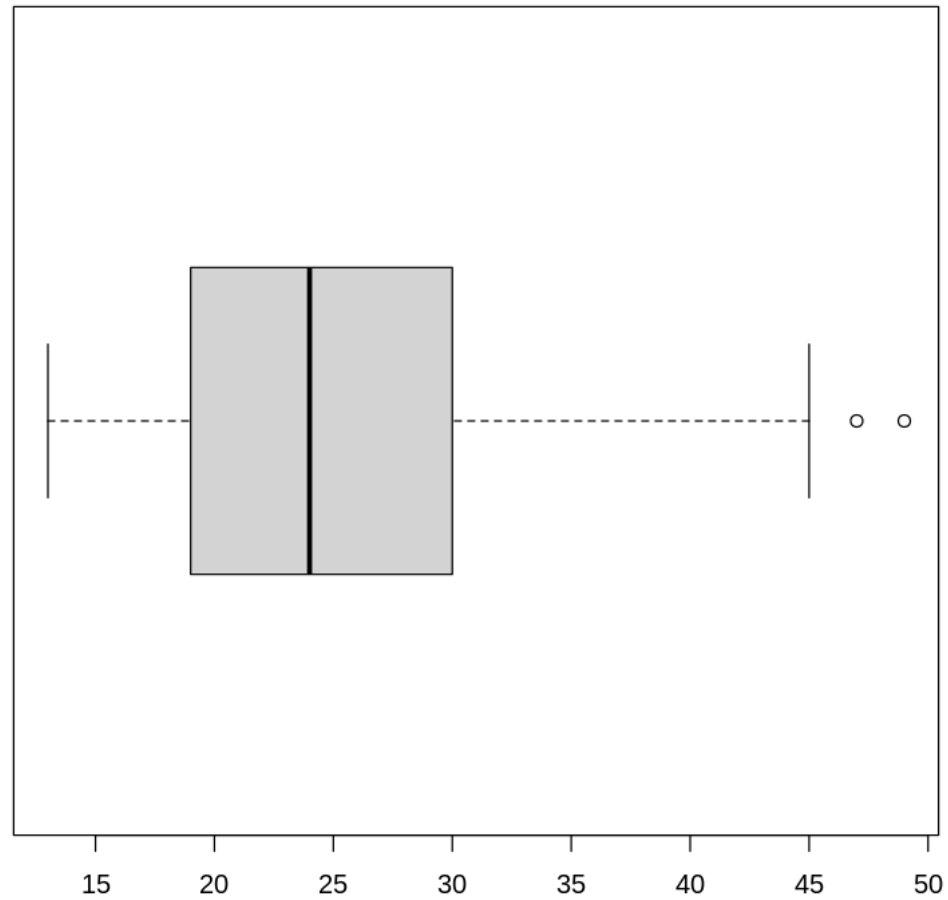
# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```

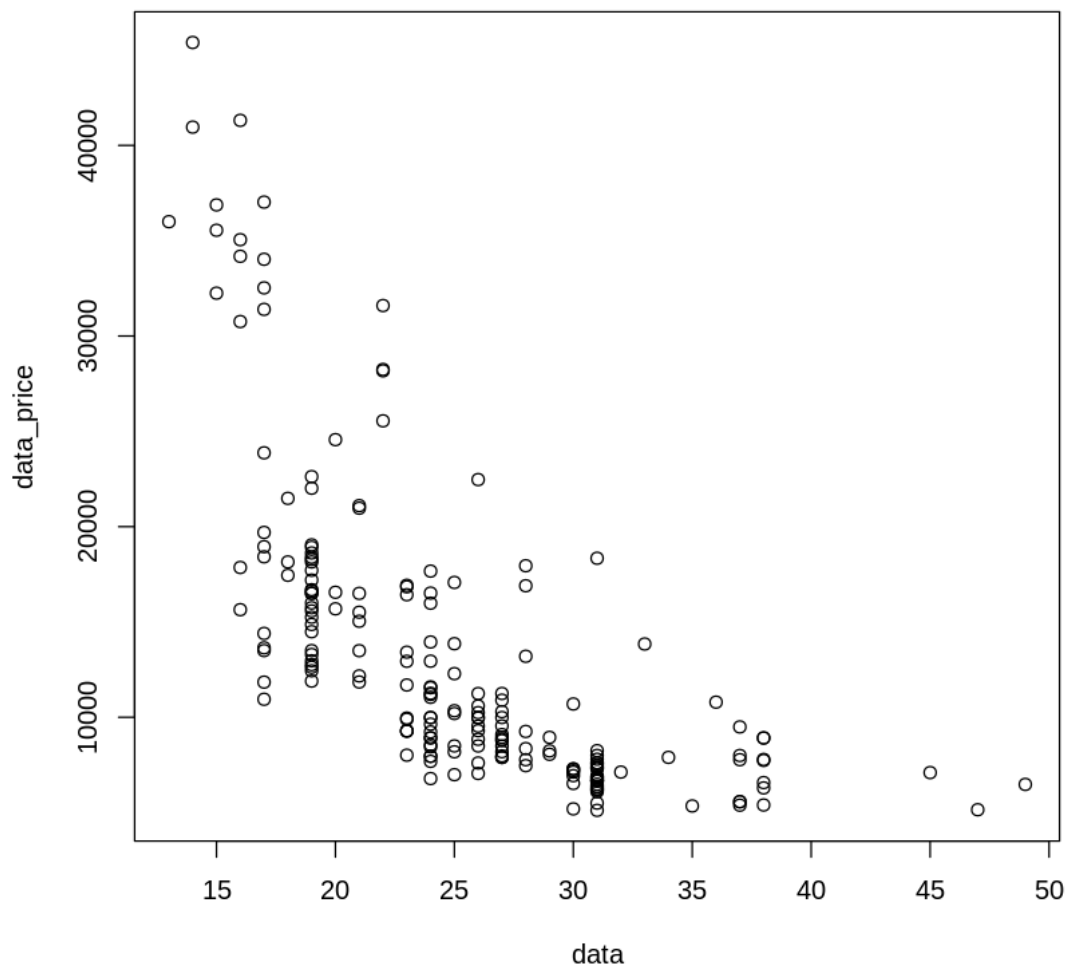


**Diagrama de dispersión de citympg**



Coeficiente de correlación entre citympg y precio: -0.6857513  
Sesgo: 0.6540229

Diagrama de dispersión de citympg



Distribución: **Asimétrica**

Hay asimetría en el histograma pues hay una mayor cola a la derecha y el sesgo es positivo.

Hay una relación relativamente alta entre las variables, pues el coeficiente de correlación fue de 0.65, podría ser buena opción para analizar.

###Highway mpg

```
[187]: variable = "highwaympg"
data = mydata[[variable]]

# Histograma
hist(data,col=0,main=paste("Histograma de", variable))
```

```

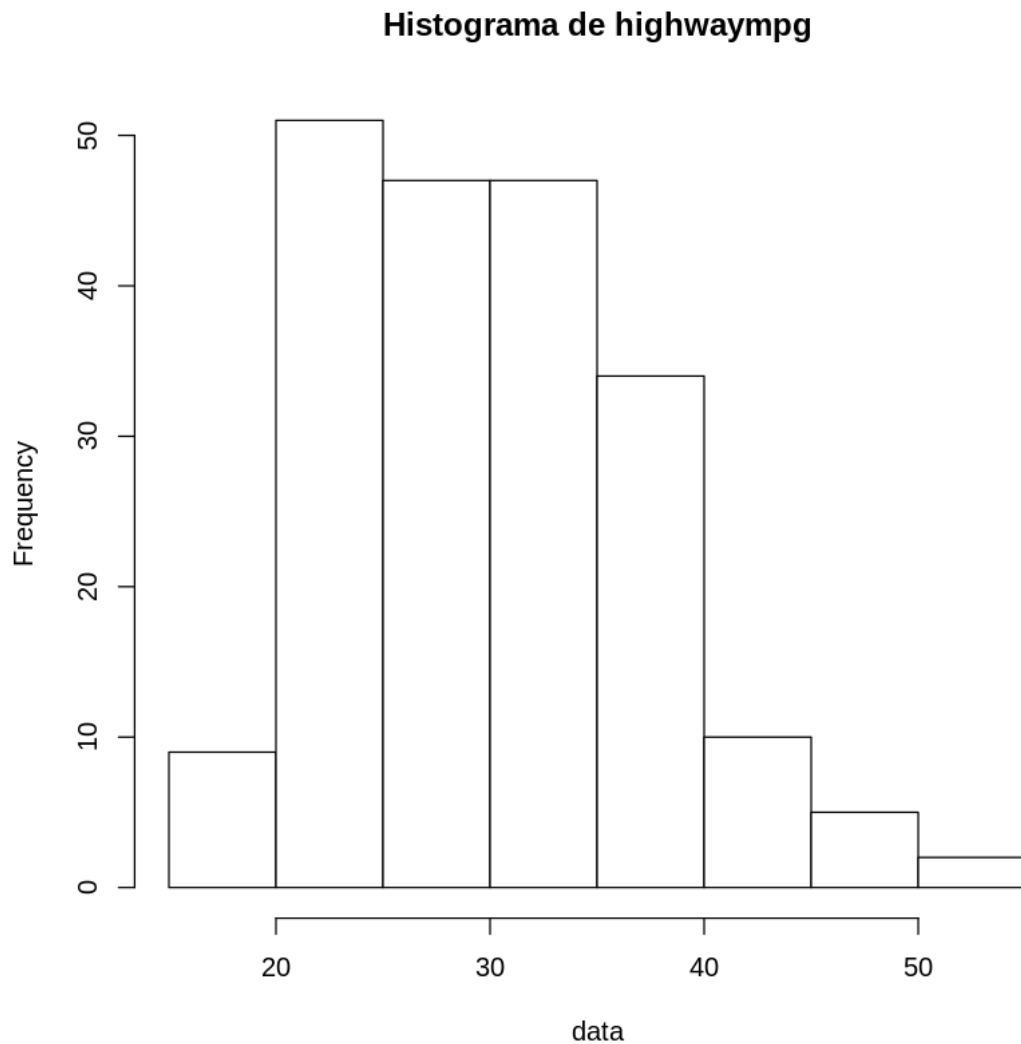
# Diagrama de caja y bigotes
boxplot(data, horizontal=TRUE, main=paste("Diagrama de dispersión de", variable))

# Diagrama de dispersión
plot(data, data_price, main=paste("Diagrama de dispersión de", variable))

# Coeficiente de correlación
coef_corr <- cor(mydata[[variable]], mydata[["price"]], use = "complete.obs")
cat("Coeficiente de correlación entre", variable, "y precio:", coef_corr, "\n")

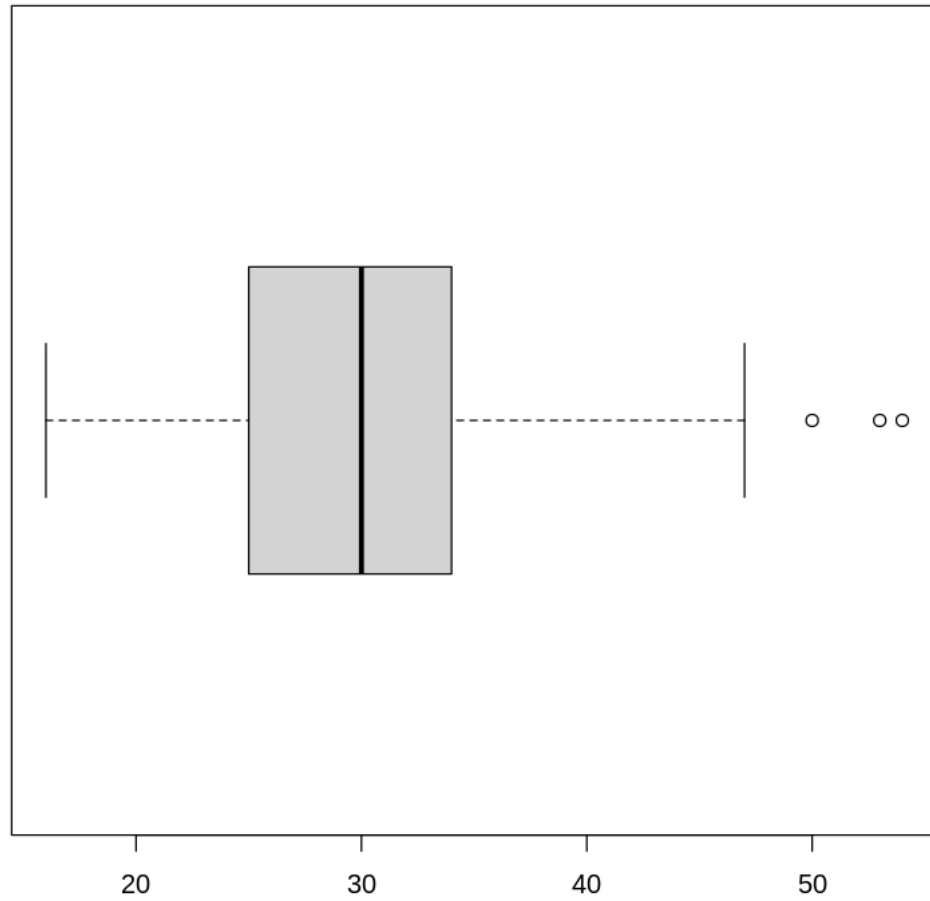
# Coeficiente de sesgo
sesgo = skewness(data)
cat("Sesgo: ", sesgo, "\n")

```



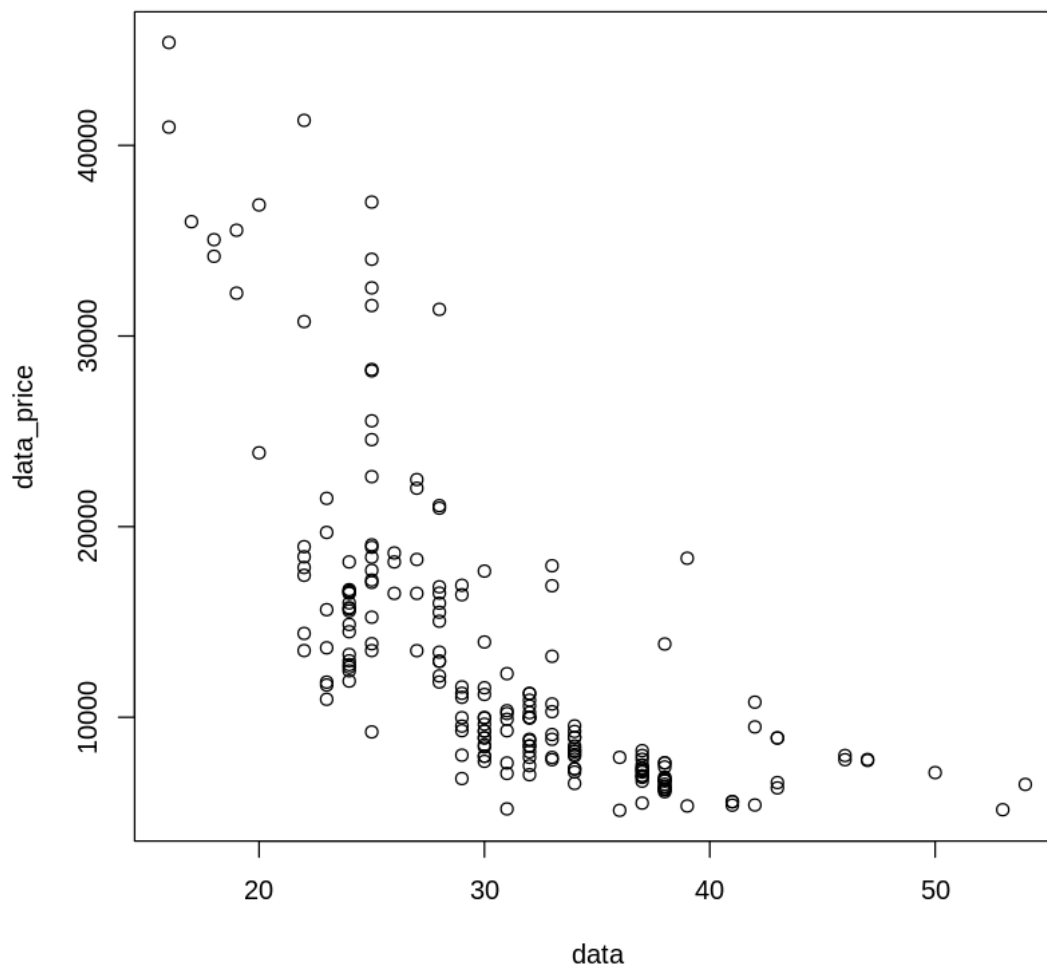


**Diagrama de dispersión de highwaympg**



Coeficiente de correlación entre highwaympg y precio: -0.6975991  
Sesgo: 0.5321205

Diagrama de dispersión de highwaympg



Distribución: **Asimétrica**

Existe asimetría hacia el lado izquierdo, debido a que la cola es ligeramente más larga de ese lado, sin embargo, no es tan grande el valor a comparación de otros.

Existe una relación negativa en la gráfica, mientras mayor es el highwaympg, menor el precio por lo visto, por lo que sería algo interesante de analizar. Es a la izquierda debido también a que el sesgo es negativo.

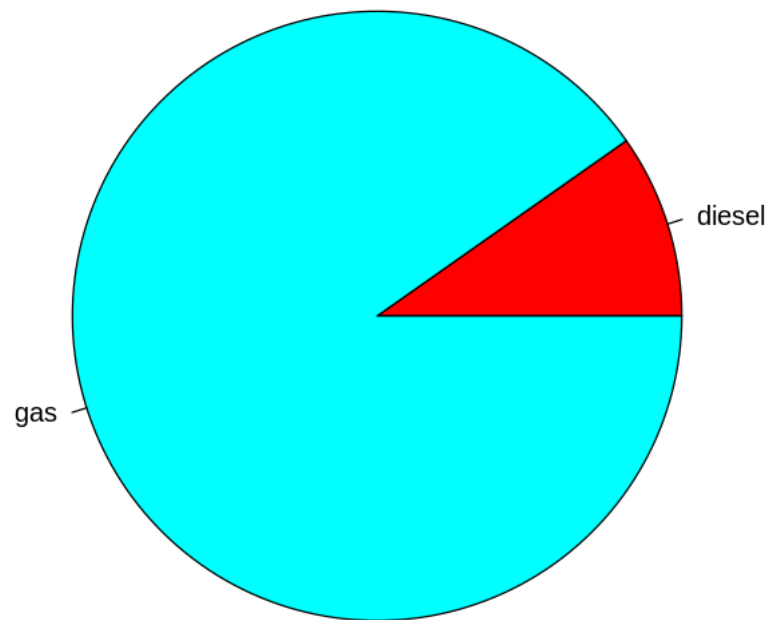
## Variables cualitativas

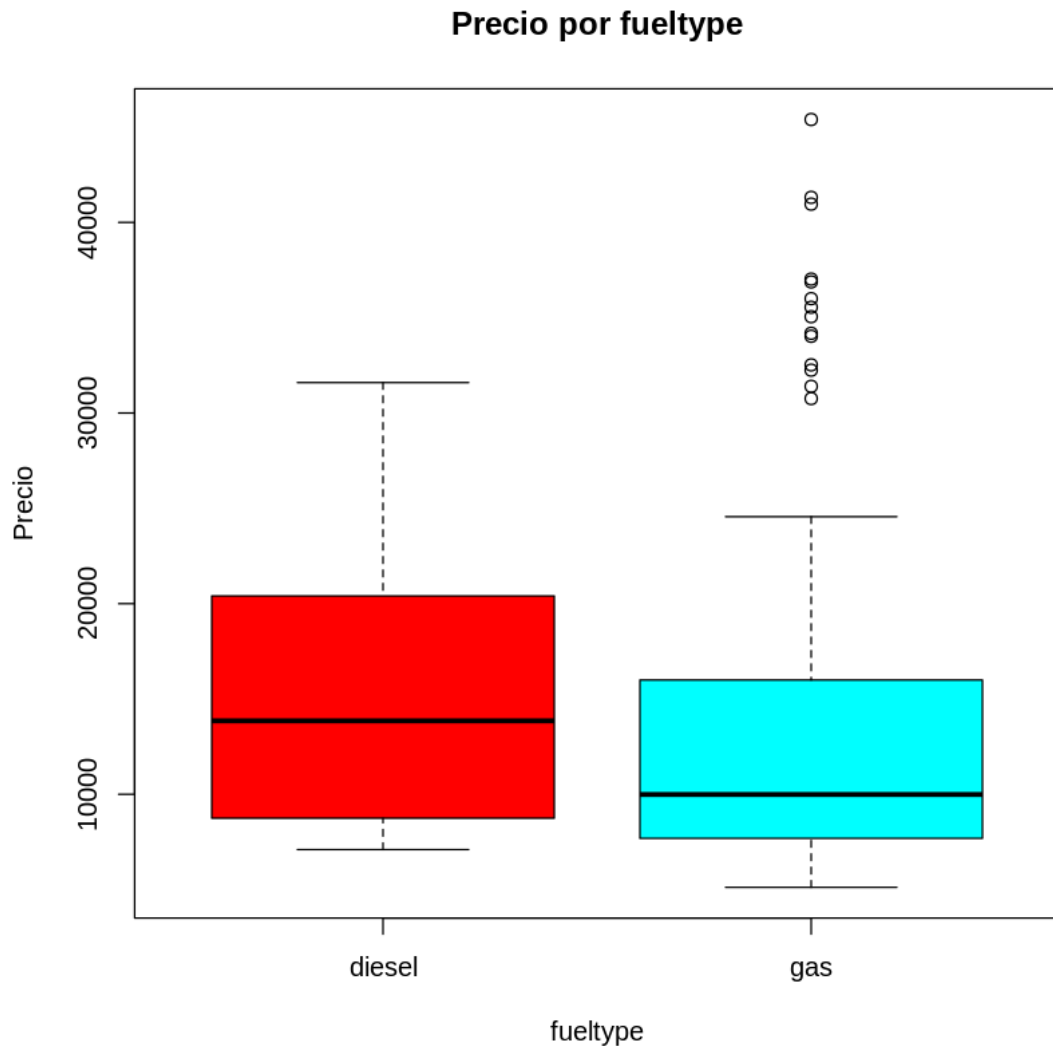
### Fuel type

```
[188]: variable = "fueltype"
      data = table(mydata[[variable]])
```

```
pie(data, main=paste("Distribución de", variable), col=rainbow(length(data)))
boxplot(mydata$price ~ mydata[[variable]], main=paste("Precio por", variable),
  ↳ylab="Precio", xlab=variable,
  ↳col=rainbow(length(unique(mydata[[variable]]))))
```

### Distribución de fueltype





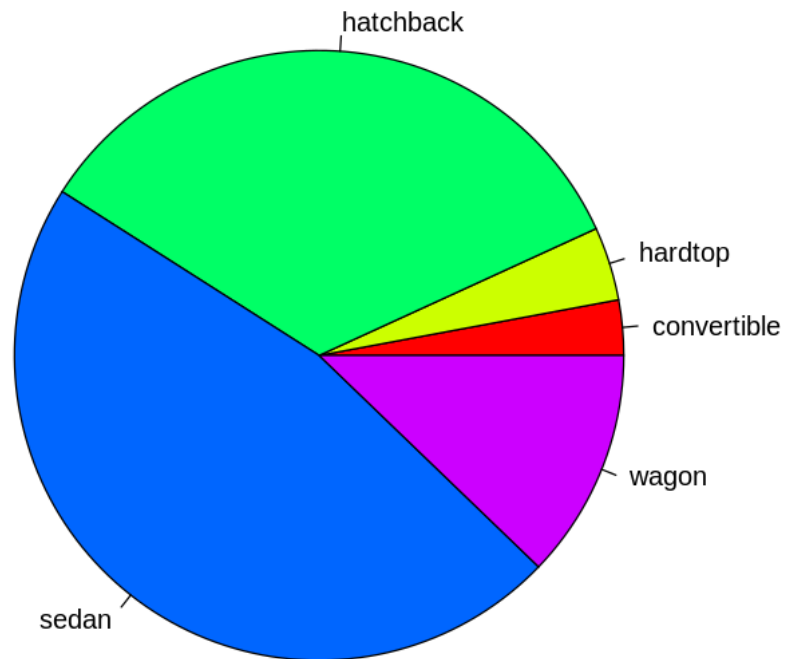
Las gráficas nos indican que es más probable que si el fueltype es de diesel, es más probable que sea más caro, sin embargo, no existe una gran diferencia entre ambos y no hay mucho punto de diferencia en el límite inferior.

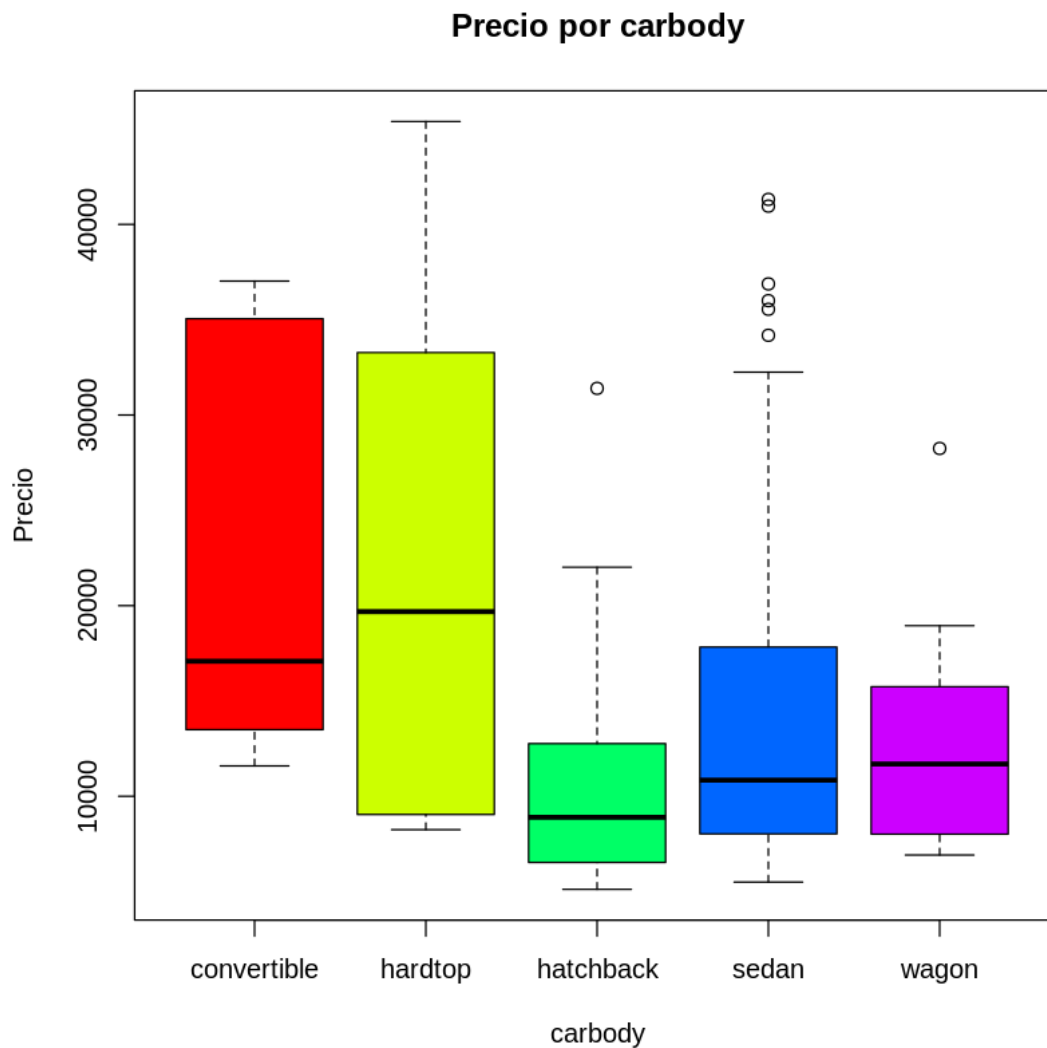
###Car body

```
[189]: variable = "carbody"
data = table(mydata[[variable]])

pie(data, main=paste("Distribución de", variable), col=rainbow(length(data)))
boxplot(mydata$price ~ mydata[[variable]], main=paste("Precio por", variable),
        ylab="Precio", xlab=variable,
        col=rainbow(length(unique(mydata[[variable]]))))
```

## Distribución de carbody



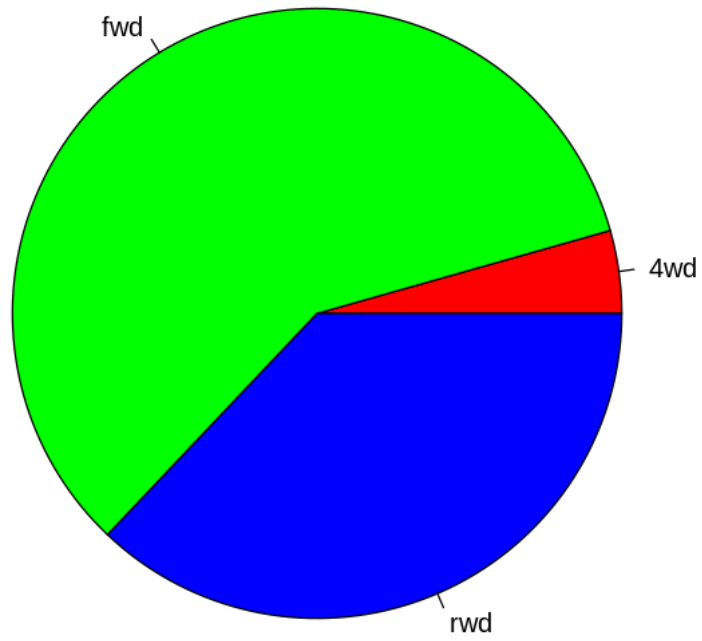


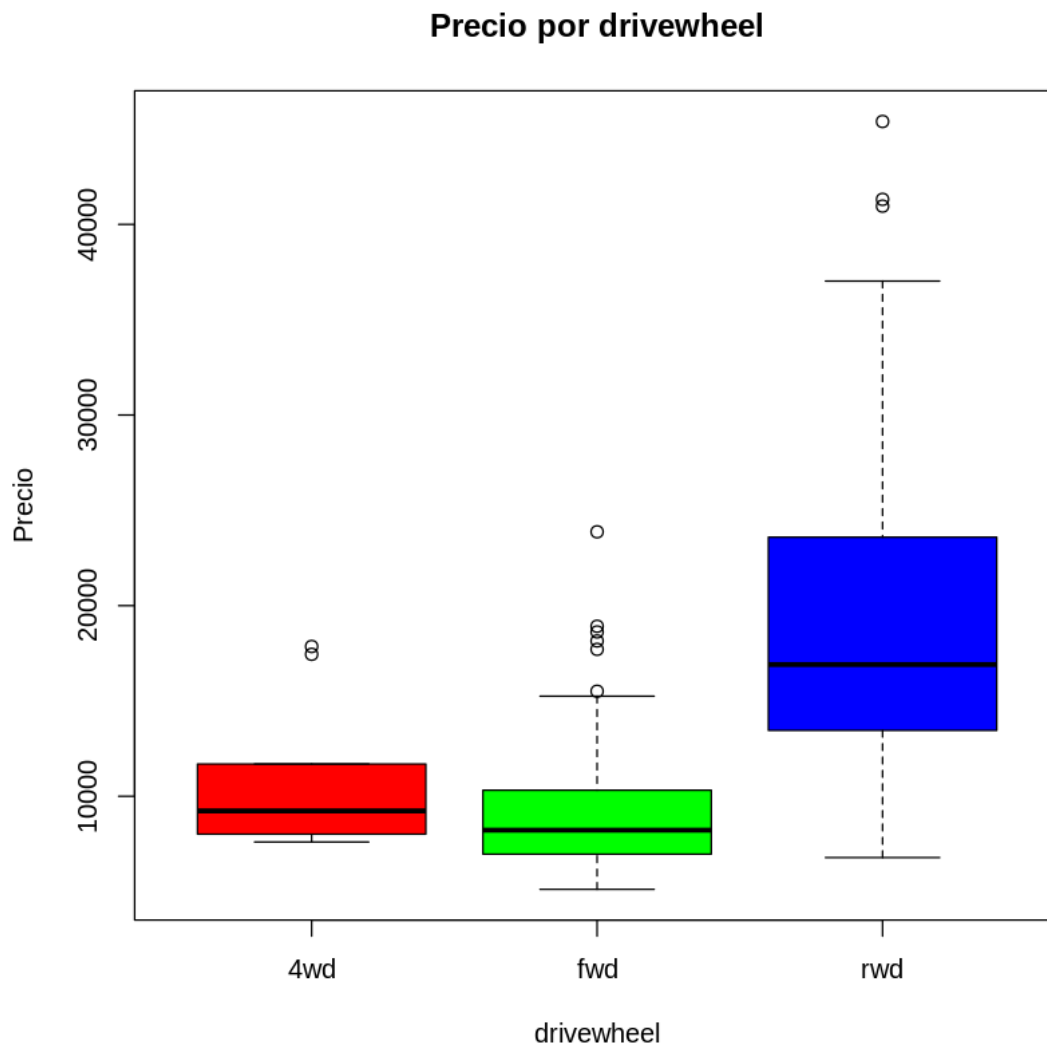
###Drive wheel

```
[190]: variable = "drivewheel"
data = table(mydata[[variable]])

pie(data, main=paste("Distribución de", variable), col=rainbow(length(data)))
boxplot(mydata$price ~ mydata[[variable]], main=paste("Precio por", variable),
        ylab="Precio", xlab=variable,
        col=rainbow(length(unique(mydata[[variable]]))))
```

## Distribución de drivewheel





Se puede observar que sí existe una diferencia significativa si la drivewheel es rwd, a comparación de una 4wd o fwd, pues entra en un rango mayor.

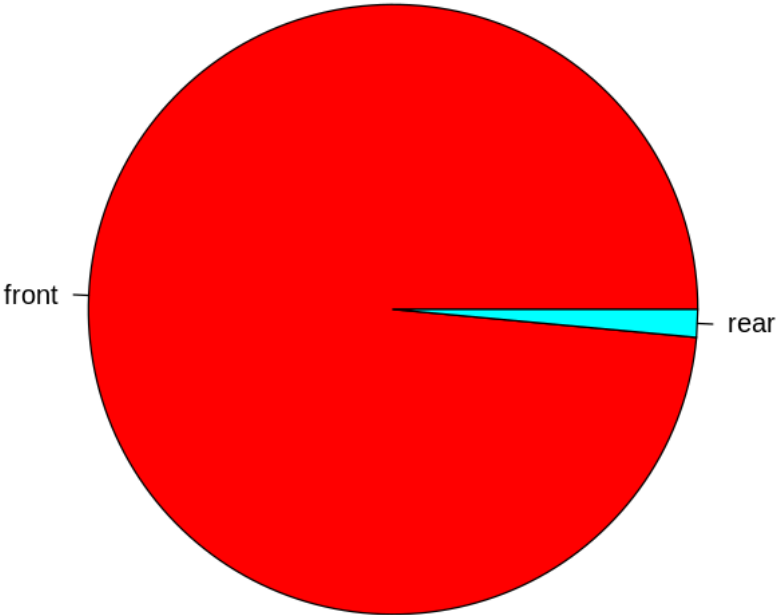
### Engine location

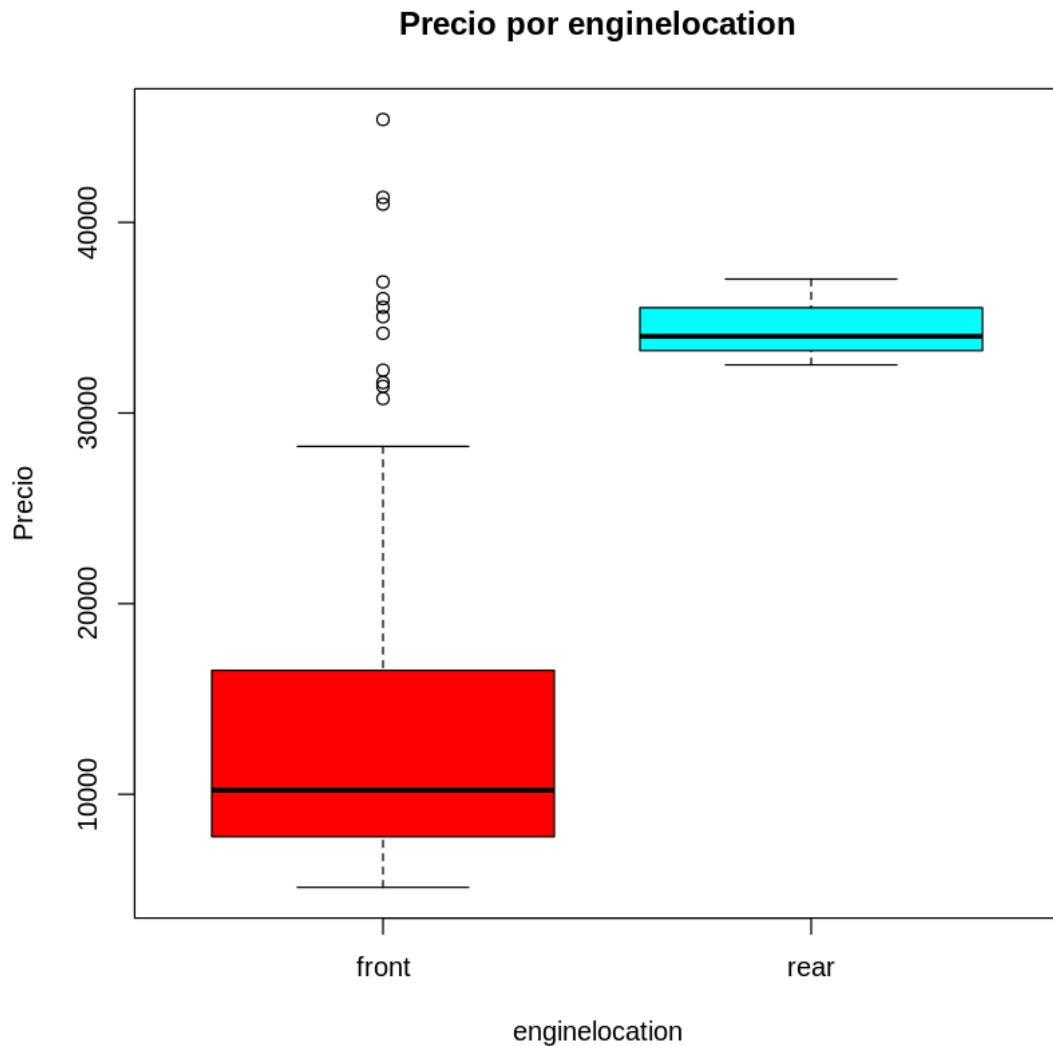
```
[191]: variable = "engine.location"
data = table(mydata[[variable]])

pie(data, main=paste("Distribución de", variable), col=rainbow(length(data)))
boxplot(mydata$price ~ mydata[[variable]], main=paste("Precio por", variable),
        ylab="Precio", xlab=variable,
        col=rainbow(length(unique(mydata[[variable]]))))
```



Distribución de enginelocation





Hay pocos datos para analizar esta variable, y son tan pocos que no tenemos suficiente evidencia para encontrar una relación.

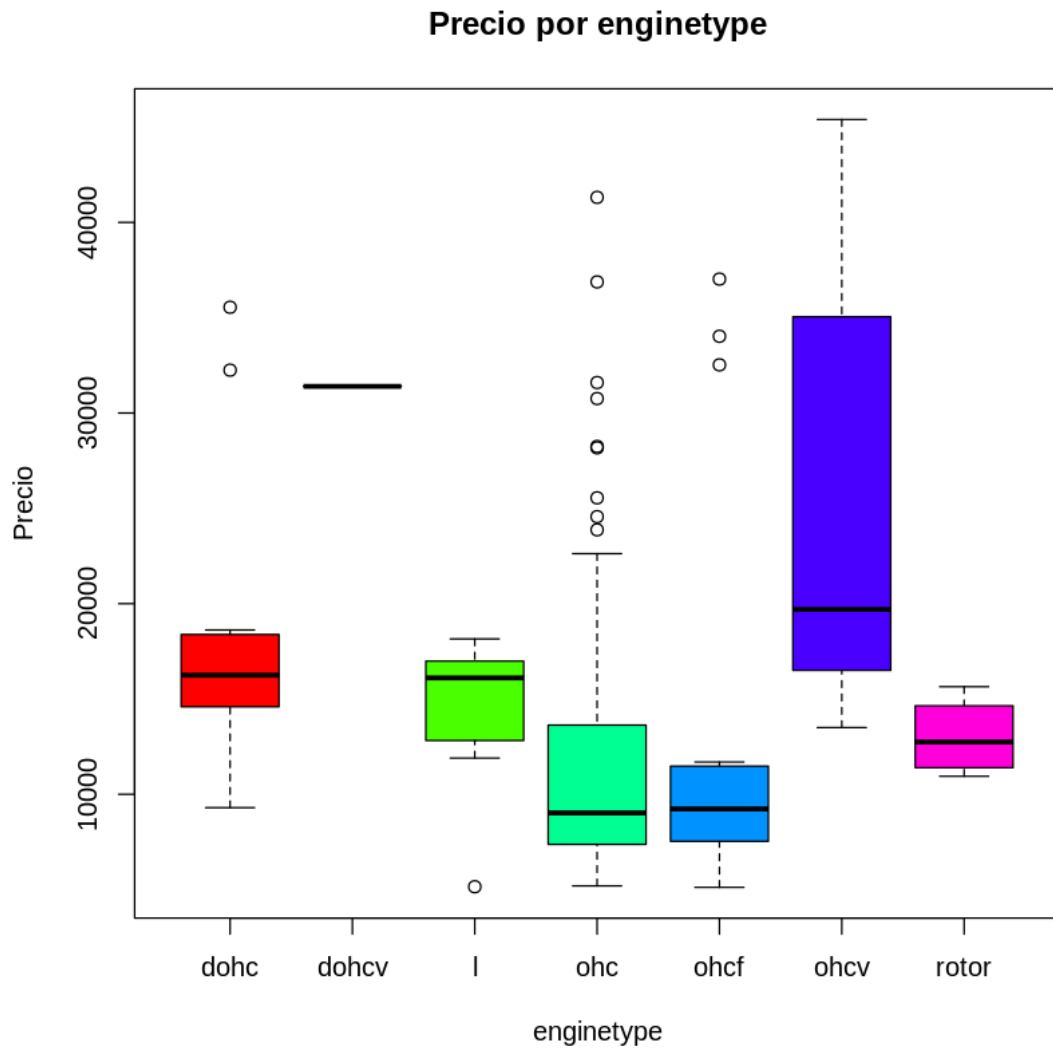
### Engine Type

```
[192]: variable = "enginetype"
data = table(mydata[[variable]])

pie(data, main=paste("Distribución de", variable), col=rainbow(length(data)))
boxplot(mydata$price ~ mydata[[variable]], main=paste("Precio por", variable),
        ylab="Precio", xlab=variable,
        col=rainbow(length(unique(mydata[[variable]]))))
```

## Distribución de enginetype



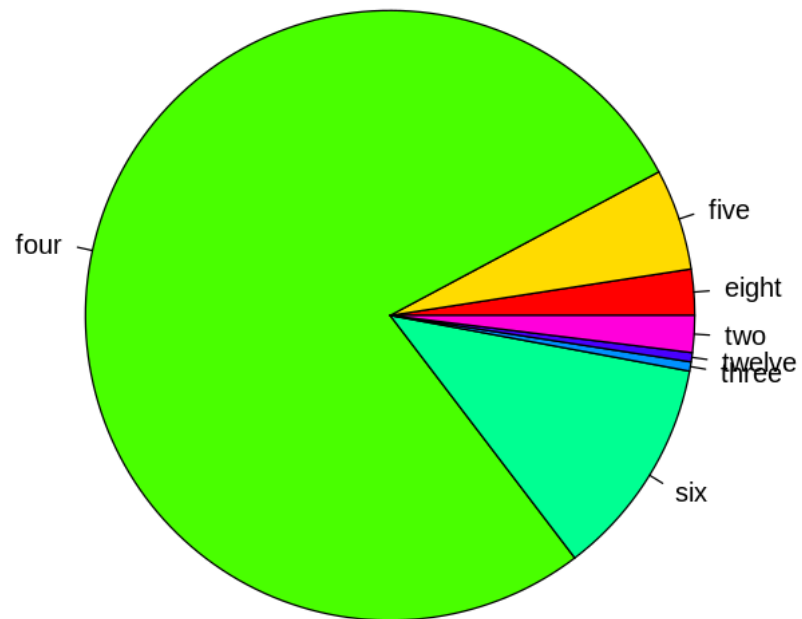


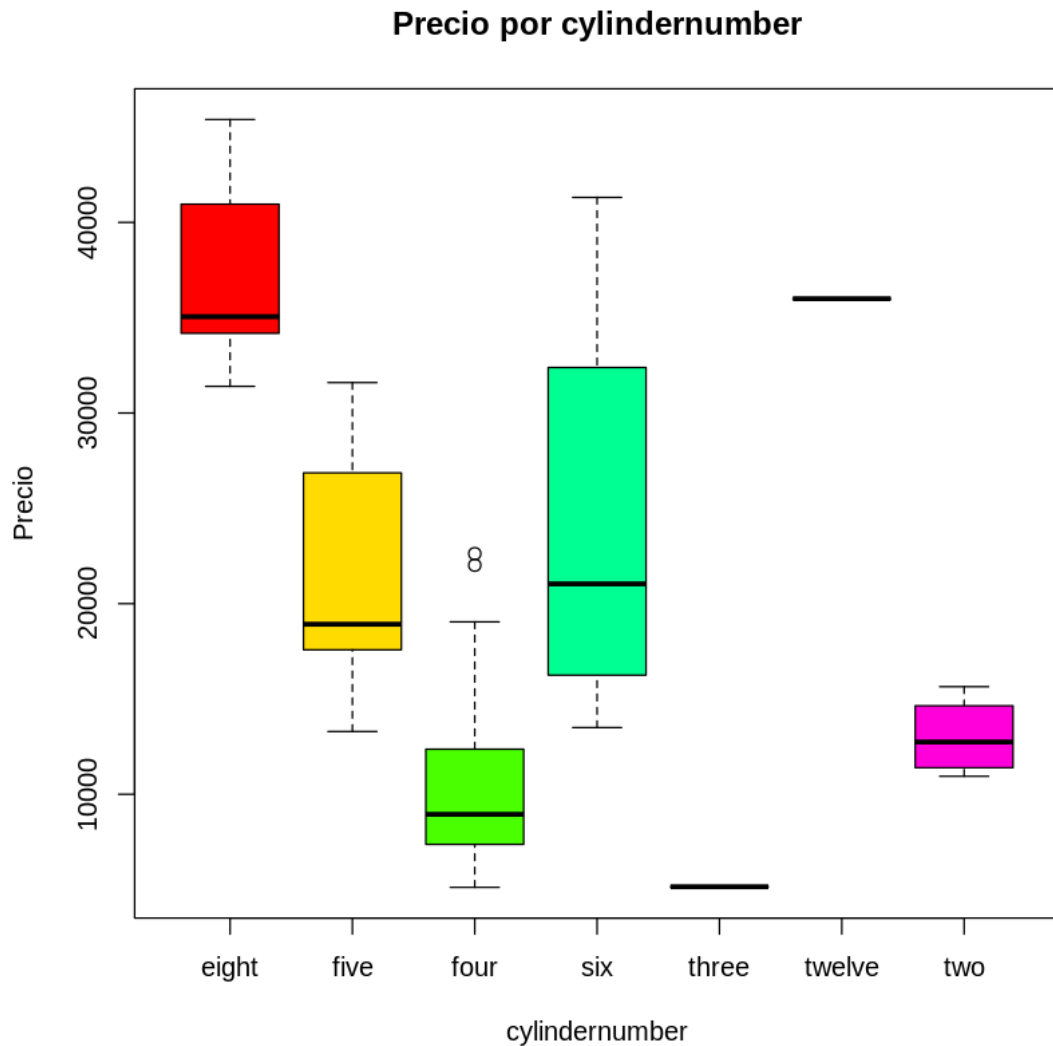
###Cylinder number

```
[193]: variable = "cylindernumber"
data = table(mydata[[variable]])

pie(data, main=paste("Distribución de", variable), col=rainbow(length(data)))
boxplot(mydata$price ~ mydata[[variable]], main=paste("Precio por", variable),
        ylab="Precio", xlab=variable,
        col=rainbow(length(unique(mydata[[variable]]))))
```

### Distribución de cylindernumber





Se puede observar que hay una mayor variabilidad de los datos, y hay más opciones, con una buena y legal cantidad de datos para entrenar cada posible respuesta.

#### 0.4 Datos seleccionados

Dados los cálculos y análisis anteriores, he seleccionado las siguientes variables para el modelo:

1. **Engine Size:** Seleccioné esta variable porque es una de las que tiene el mayor coeficiente de correlación con la variable de precio, y nos puede ayudar a predecir buenos valores. Aunque no está distribuida, esto se puede transformar a futuro.
2. **Car width:** Es también una de las variables con el mayor coeficiente de correlación y puede indicarnos buenas predicciones, quizás mientras más ancho esté el carro y más espacio, puede consumir más recursos y ser más probable a costrar más.

3. **Drive wheel:** Se seleccionó drive wheel porque se notó que además de al parecer, sí hay una diferencia grande y notable cuando se selecciona una rueda de rwd, a comparación de las otras.
2. También, dado que solo son 3 posibles resultados, se facilita también en la implementación, además de que es lógico que el tipo de llanta influya en el costo del auto.
4. **Curb weight:** Esta es una de las variables que quizás consideremos más importantes debido a que el coeficiente de correlación parece ser muy alto, y afortunadamente no cuenta con un sesgo tan alto como otras variables, por lo que puede llegar a ser muy bueno.
5. **Cylinder number:** Es una de las variables más representativas para este modelo, pues afortunadamente con esta variable cuentas con posibles salidas muy lejanas, y esto permite también que se entrene mejor al modelo. La cantidad de cilindros forma un papel fundamental y se manejará adecuadamente.
6. **City mpg:** Este es un valor muy interesante pero que también puede tener un impacto significativo. Tenía una confianza de -0.68 y además será bueno incluirlo porque al parecer trabajar con una función inversa y nos puede llevar a resultados interesantes. No tiene un valor tan grande de sesgo por lo que podríamos trabajar con él fácilmente.

En este caso, la función de lm trabaja perfectamente con variables numéricas y categóricas, por lo que no es necesario usar la técnica de one hot encoding para convertir a variables dummies.

En este caso, se deberán también transformar todas las variables cuantitativas que identificamos que no tenían distribución normal, esto se podrá realizar por medio de alguno de los métodos Yeo-Johnson o Boxcox igualmente.

Las variables a normalizar serán todas las seleccionadas, pues ninguna contaba con simetría inicial, se buscará que tengan una distribución balanceada en todo el set.

De igual forma, será necesario escalar los datos, pues los datos seleccionados se encuentran en rangos distintos y esto podría afectar el modelo que realicemos.

Esto se implementará en la versión final del entregable y en la entrega para retro de portafolio de implementación.

## 1 PARTE 2: Construcción de un modelo estadístico base

Como se mencionó en la entrega anterior, se seleccionaron las siguientes variables que consideré importantes para analizar las características que determinan el precio de los automóviles. Estas variables seleccionadas fueron: \* Engine Size \* Car width \* Horsepower \* Drive wheel \* Curb weight \* Cylinder number \* City mpg

Como podemos ver, en esta nueva lista se agregó Horsepower, esto debido a que al estar analizando nuevamente las variables, pude identificar que también tenía un coeficiente de correlación muy alto (0.80), por lo que también podría ser una variable muy importante para el modelo.

Ahora que ya tenemos las posibles variables involucradas, se creará un nuevo dataframe con solamente estas variables y la variable de precio, que es la que intentamos predecir.

```
[194]: M <- mydata[, c("drivewheel", "carwidth", "enginesize", "curbweight", "
↪horsepower", "cylindernumber", "citympg", "highwaympg", "fueltype", "price")]
M
```

	drivewheel <chr>	carwidth <dbl>	enginesize <int>	curbweight <int>	horsepower <int>	cylindernumber <chr>	citympg <int>
	rwd	64.1	130	2548	111	four	21
	rwd	64.1	130	2548	111	four	21
	rwd	65.5	152	2823	154	six	19
	fwd	66.2	109	2337	102	four	24
	4wd	66.4	136	2824	115	five	18
	fwd	66.3	136	2507	110	five	19
	fwd	71.4	136	2844	110	five	19
	fwd	71.4	136	2954	110	five	19
	fwd	71.4	131	3086	140	five	17
	4wd	67.9	131	3053	160	five	16
	rwd	64.8	108	2395	101	four	23
	rwd	64.8	108	2395	101	four	23
	rwd	64.8	164	2710	121	six	21
	rwd	64.8	164	2765	121	six	21
	rwd	66.9	164	3055	121	six	20
	rwd	66.9	209	3230	182	six	16
	rwd	67.9	209	3380	182	six	16
	rwd	70.9	209	3505	182	six	15
	fwd	60.3	61	1488	48	three	47
	fwd	63.6	90	1874	70	four	38
	fwd	63.6	90	1909	70	four	38
	fwd	63.8	90	1876	68	four	37
	fwd	63.8	90	1876	68	four	31
	fwd	63.8	98	2128	102	four	24
	fwd	63.8	90	1967	68	four	31
	fwd	63.8	90	1989	68	four	31
	fwd	63.8	90	1989	68	four	31
	fwd	63.8	98	2191	102	four	24
	fwd	64.6	122	2535	88	four	24
A data.frame: 205 × 10	fwd	66.3	156	2811	145	four	19
	fwd	66.5	122	2414	92	four	27
	fwd	66.5	122	2414	92	four	27
	fwd	66.5	122	2458	92	four	27
	rwd	67.7	171	2976	161	six	20
	rwd	67.7	171	3016	161	six	19
	rwd	66.5	171	3131	156	six	20
	rwd	66.5	161	3151	156	six	19
	fwd	65.5	97	2261	52	four	37
	fwd	65.5	109	2209	85	four	27
	fwd	65.5	97	2264	52	four	37
	fwd	65.5	109	2212	85	four	27
	fwd	65.5	109	2275	85	four	27
	fwd	65.5	97	2319	68	four	37
	fwd	65.5	109	2300	100	four	26
	fwd	64.2	109	2254	90	four	24
	fwd	64.0	109	2221	90	four	24
	fwd	66.9	136	2661	110	five	19
	fwd	66.9	97	2579	68	four	33
	fwd	66.9	109	2563	88	four	25
	rwd	67.2	141	2912	114	four	23



Nos quedamos solamente con las variables seleccionadas a analizar y con nuestra variable objetivo (precio). Ahora, como se analizó previamente, todas las variables no tienen una distribución completamente normal, por lo que quizás será útil utilizar una técnica de normalización de los datos y transformarlos.

## 1.1 Normalización de datos

Como sabemos, para crear un modelo de regresión lineal, es ideal que normalicemos nuestros datos, es decir, transformarlos para que sigan una distribución normal.

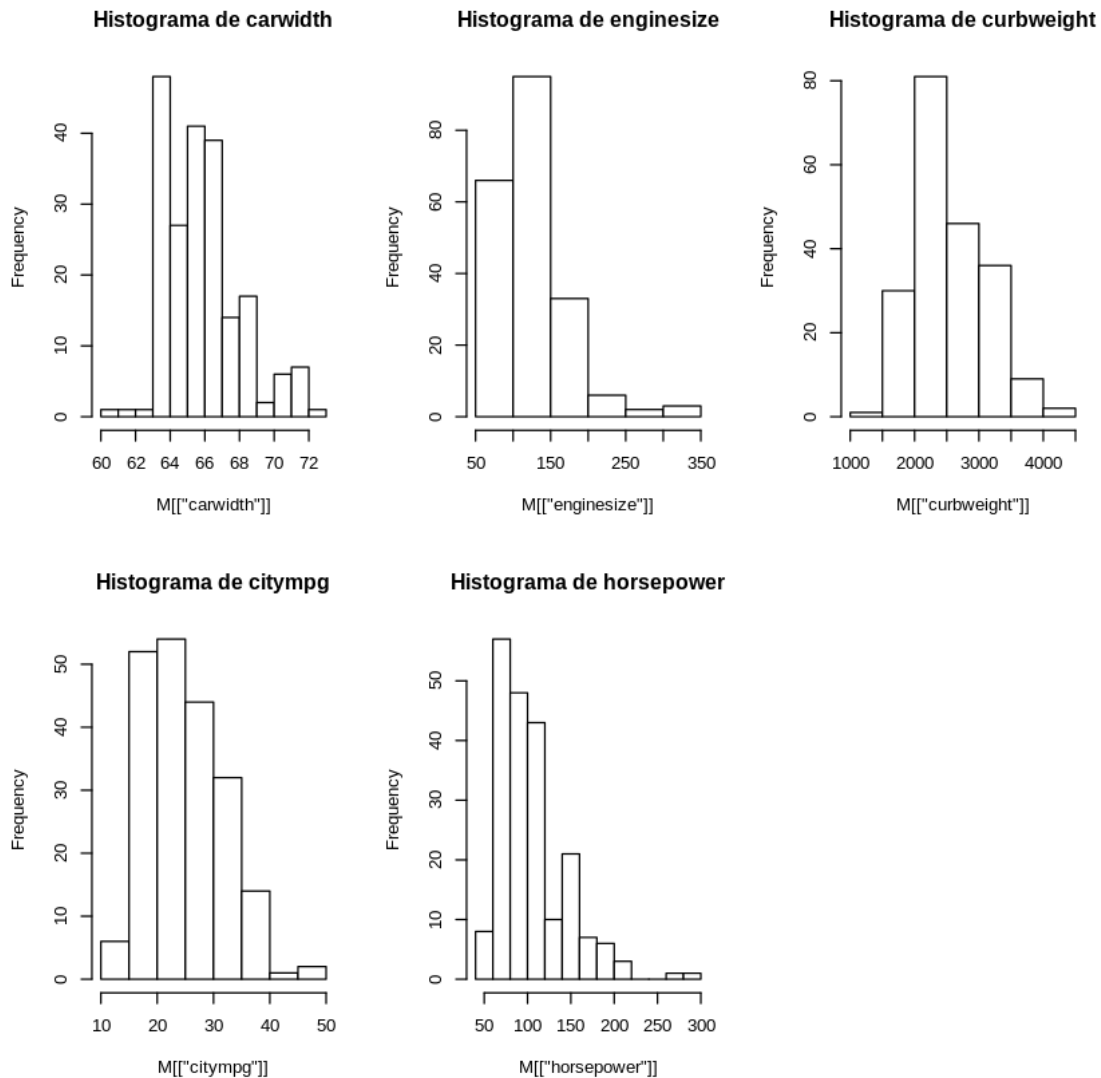
Para comenzar con la normalización de los datos, vamos a instalar la librería de MASS.

```
[195]: install.packages("MASS")  
library(MASS)
```

Installing package into ‘/usr/local/lib/R/site-library’  
(as ‘lib’ is unspecified)

Nuevamente graficaremos las distribuciones por medio de un histograma para analizar qué tal se ve la distribución de los datos de las variables seleccionadas. (cuantitativas)

```
[196]: par(mfrow = c(2, 3))  
hist(M[["carwidth"]], col=0, main=paste("Histograma de carwidth"))  
hist(M[["enginesize"]], col=0, main=paste("Histograma de enginesize"))  
hist(M[["curbweight"]], col=0, main=paste("Histograma de curbweight"))  
hist(M[["citympg"]], col=0, main=paste("Histograma de citympg"))  
hist(M[["horsepower"]], col=0, main=paste("Histograma de horsepower"))
```



Antes de normalizar los datos, vamos a intentar crear un modelo para ver qué es lo que nos arroja, y ver cuáles son las variables que están teniendo una mayor significancia en el modelo.

```
[197]: A = lm(M$price~M$drivewheel+M$carwidth+M$enginesize+M$curbweight+M$citympg+M$cylindernumber+M$horsepower)
summary(A)
```

Call:

```
lm(formula = M$price ~ M$drivewheel + M$carwidth + M$enginesize +
    M$curbweight + M$citympg + M$cylindernumber + M$horsepower)
```

Residuals:

Min	1Q	Median	3Q	Max
-7466.1	-1173.8	52.1	1211.2	14091.2

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-23979.136	15314.338	-1.566	0.11905
M\$drivewheel fwd	-437.896	1207.710	-0.363	0.71732
M\$drivewheel rwd	1844.928	1247.453	1.479	0.14080
M\$carwidth	337.824	255.885	1.320	0.18834
M\$enginesize	83.261	17.786	4.681	5.39e-06 ***
M\$curbweight	1.248	1.493	0.835	0.40453
M\$citympg	63.320	71.395	0.887	0.37625
M\$cylindernumber five	-1475.759	2220.164	-0.665	0.50704
M\$cylindernumber four	-6000.201	2293.367	-2.616	0.00960 **
M\$cylindernumber six	-4124.883	1926.720	-2.141	0.03355 *
M\$cylindernumber three	-2801.141	4233.564	-0.662	0.50899
M\$cylindernumber twelve	-10001.708	3651.013	-2.739	0.00674 **
M\$cylindernumber two	-1907.075	3356.628	-0.568	0.57060
M\$horsepower	43.477	13.062	3.329	0.00105 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3128 on 191 degrees of freedom

Multiple R-squared: 0.8564, Adjusted R-squared: 0.8467

F-statistic: 87.65 on 13 and 191 DF, p-value: < 2.2e-16

A simple vista, y sin entrar en detalles, podemos identificar que las variables determinantes y significativas en el modelo. Estas variables son las siguientes:

- Enginesize
- Cylindernumberfour
- Cylindernumbertwelve
- Horsepower

Estas variables cuentan con los valores t más altos, y además tienen un valor de p pequeño en relación con el valor t. Sin embargo, aun no vamos a ahondar mucho en este análisis, vamos a normalizar las variables primero e intentar crear otra vez el modelo con las variables normalizadas.

Creamos una función que nos permita normalizar los datos de manera modular, para que no tengamos que incluir el proceso tantas veces. La función recibe el dataframe y el nombre de la variable a utilizar y retorna los datos ya normalizados.

```
[198]: normalize_data <- function(df, variable) {
  # Importamos la librería necesaria para boxcox
  library(MASS)

  # Realizamos la transformación Box-Cox
  bc <- boxcox((df[[variable]] + 1) ~ 1)
```

```

# Encontramos el valor de lambda que maximiza la log-verosimilitud
lambda <- bc$x[which.max(bc$y)]

# Normalizamos los datos usando el valor de lambda encontrado
normal_data <- ((df[[variable]] + 1)^lambda - 1) / lambda

return(normal_data)}

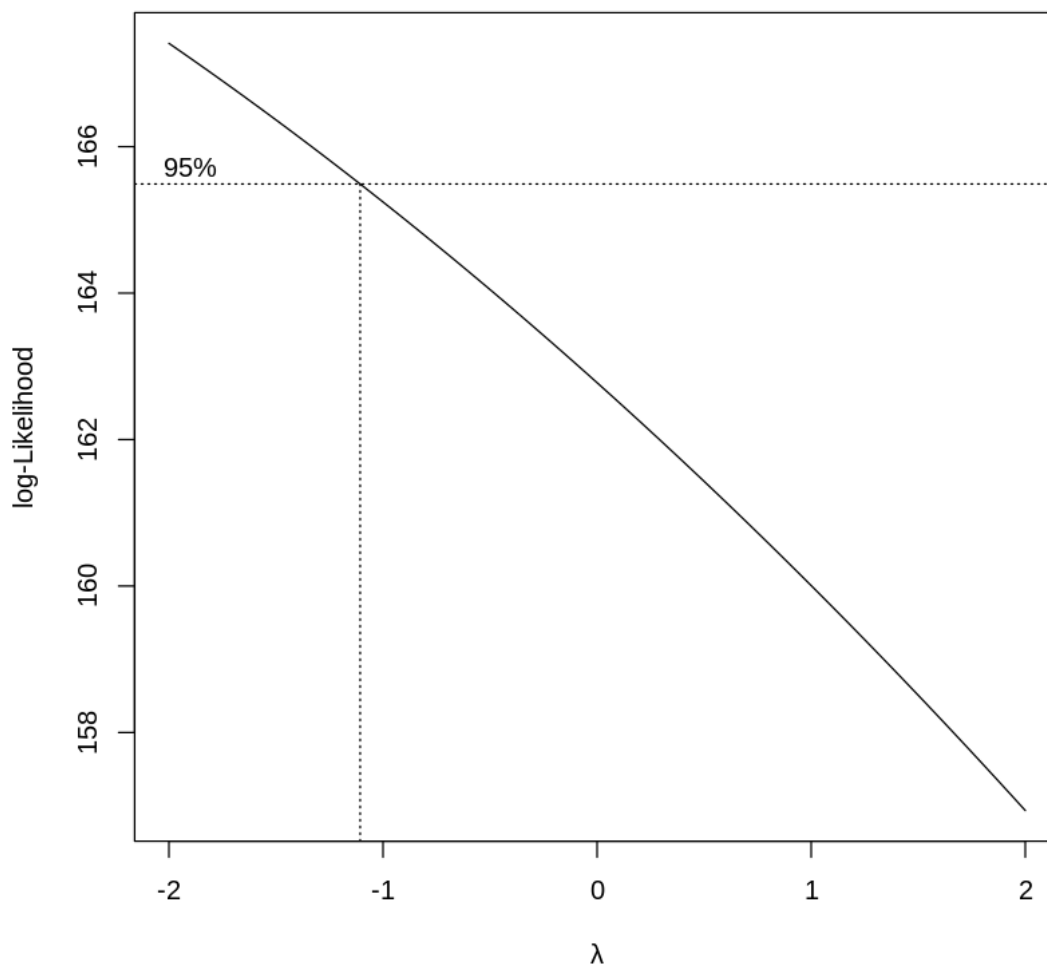
```

Comenzamos a aplicar la función de normalización en las variables cuantitativas seleccionadas, y utilizamos gráficas para observar el cambio.

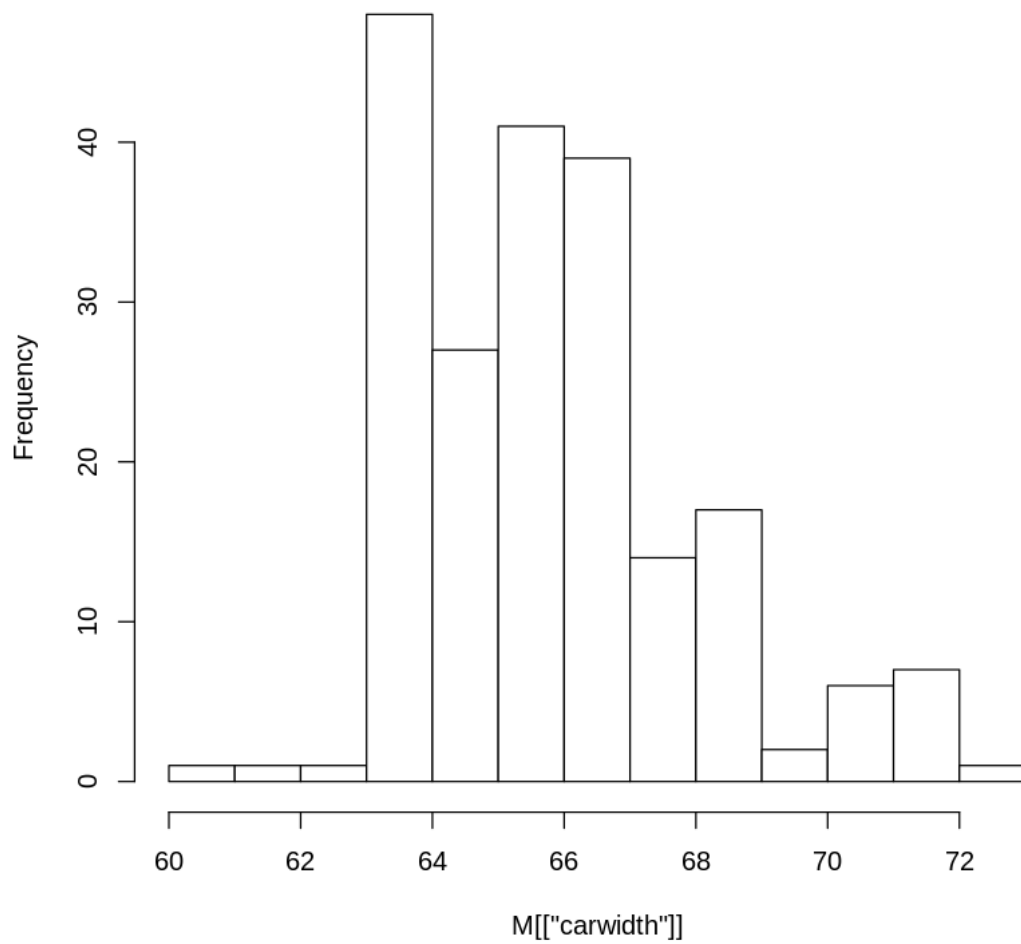
```

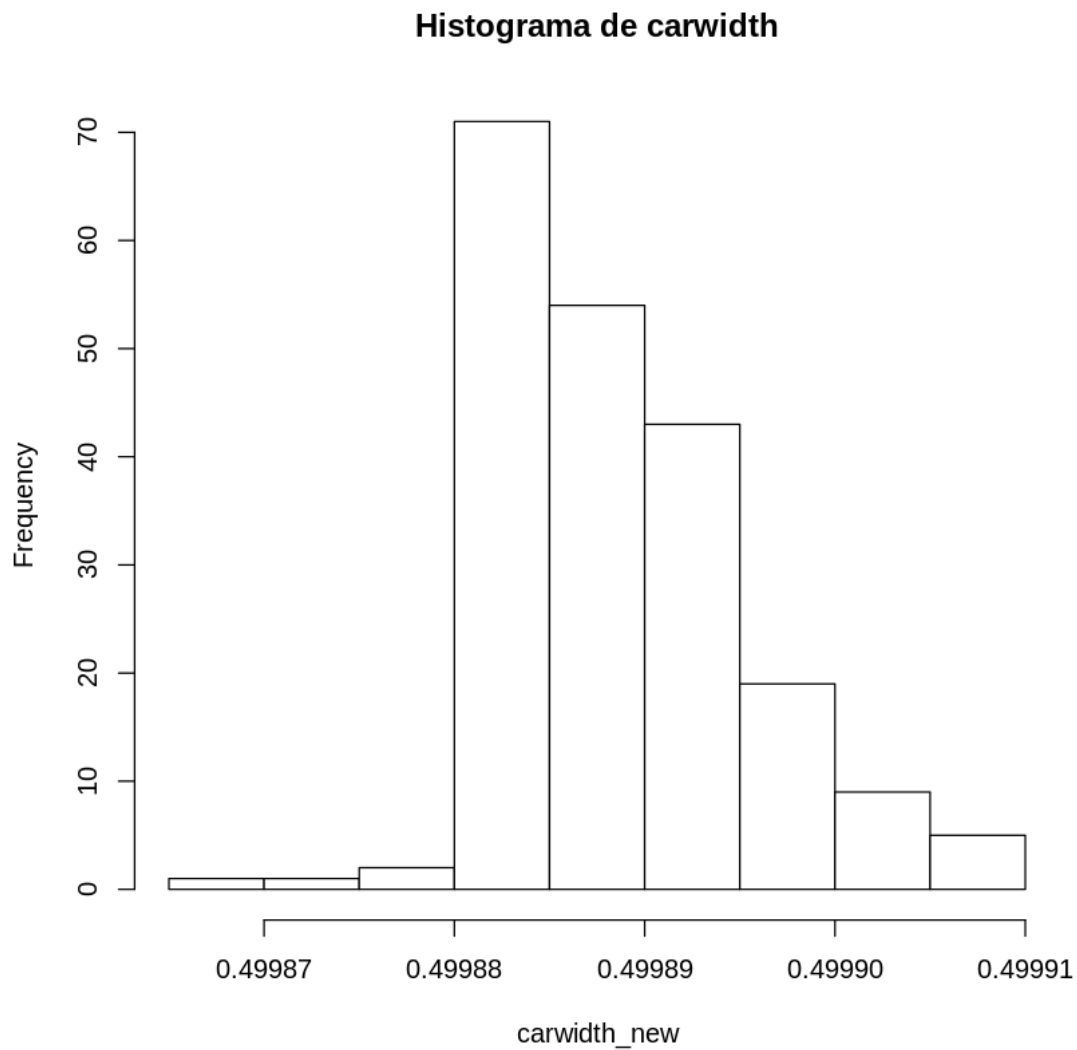
[199]: carwidth_new <- normalize_data(M, "carwidth")
hist(M[["carwidth"]],col=0,main=paste("Histograma de carwidth"))
hist(carwidth_new,col=0,main=paste("Histograma de carwidth"))

```

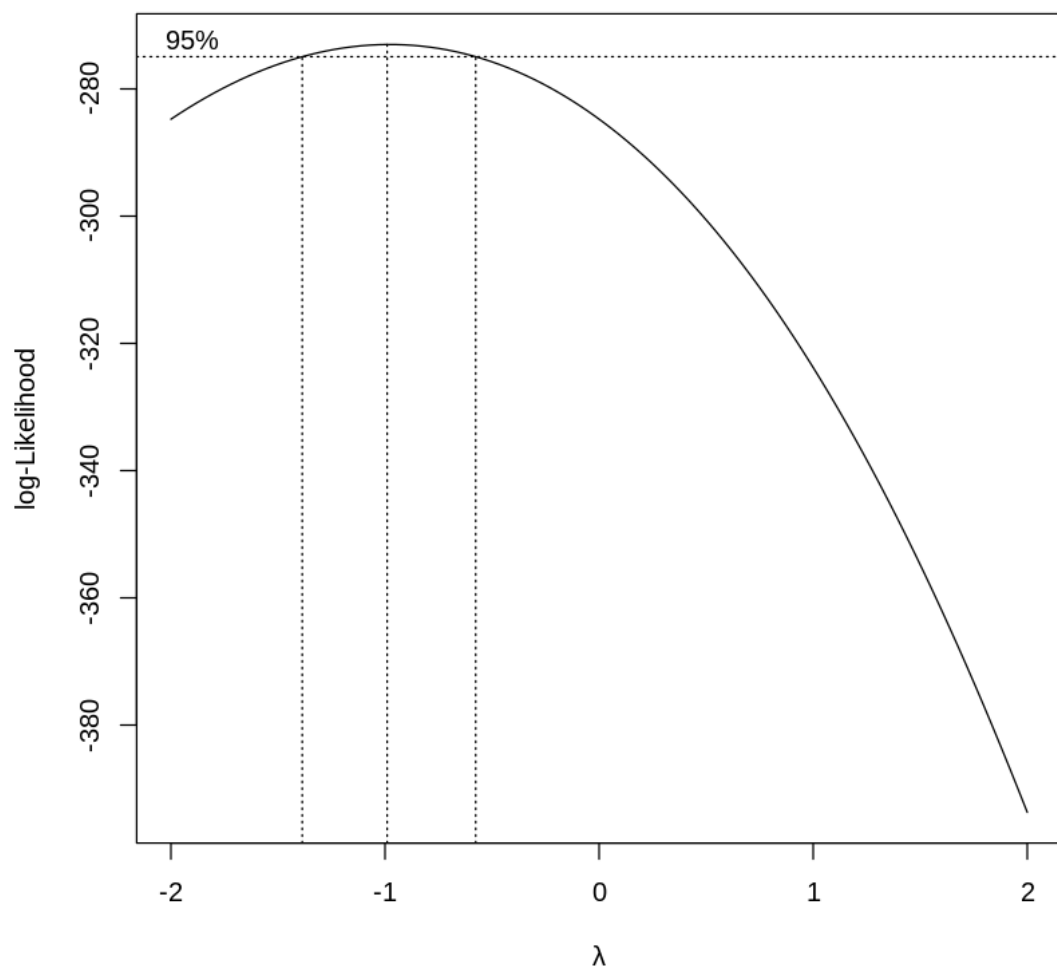


**Histograma de carwidth**

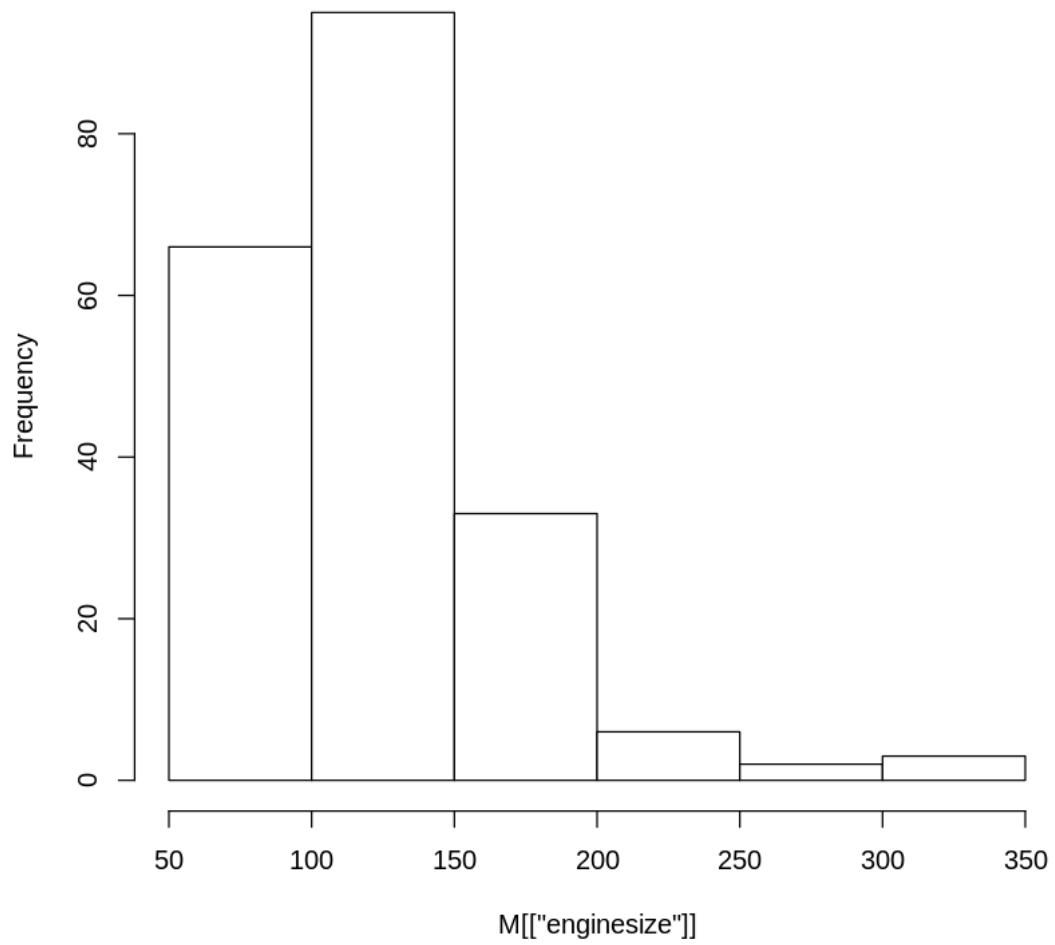




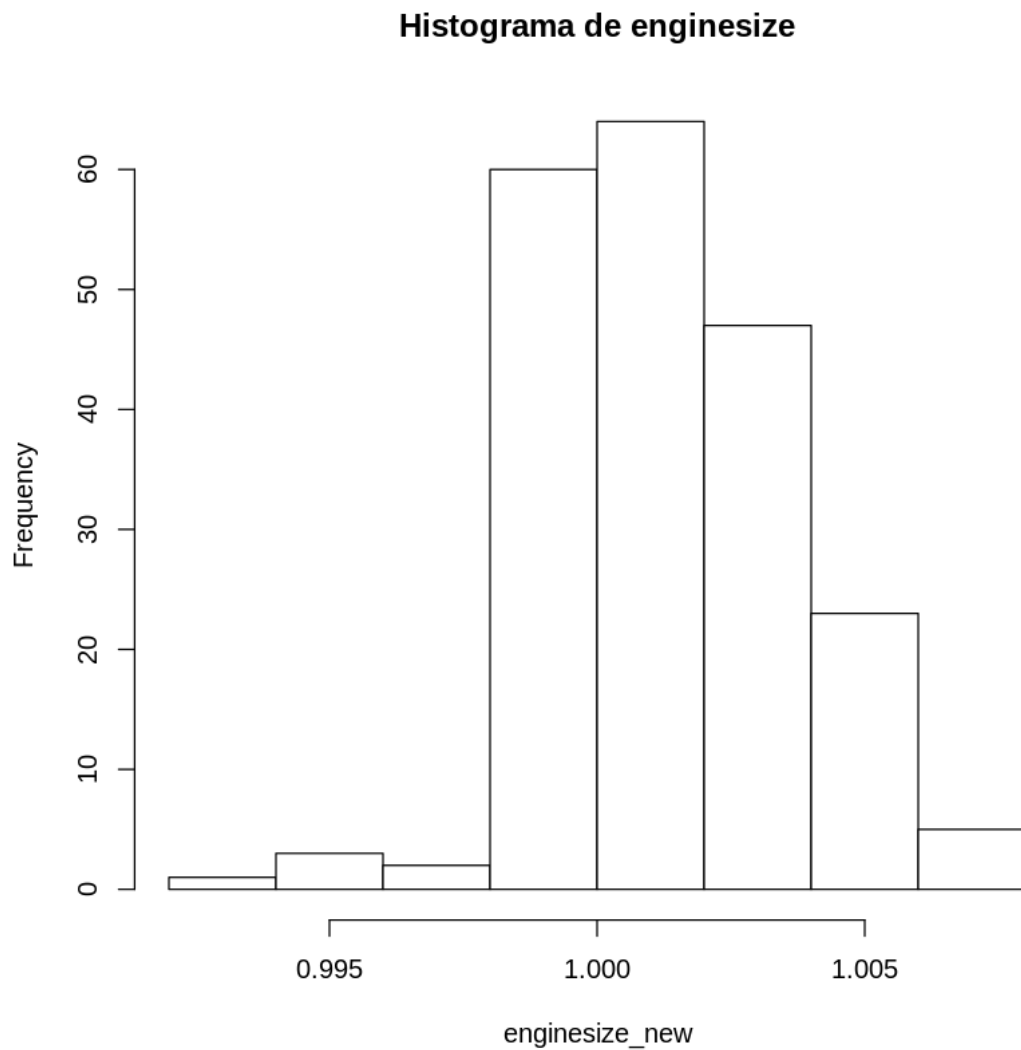
```
[200]: enginesize_new <- normalize_data(M, "enginesize")
hist(M[["enginesize"]],col=0,main=paste("Histograma de enginesize"))
hist(enginesize_new,col=0,main=paste("Histograma de enginesize"))
```



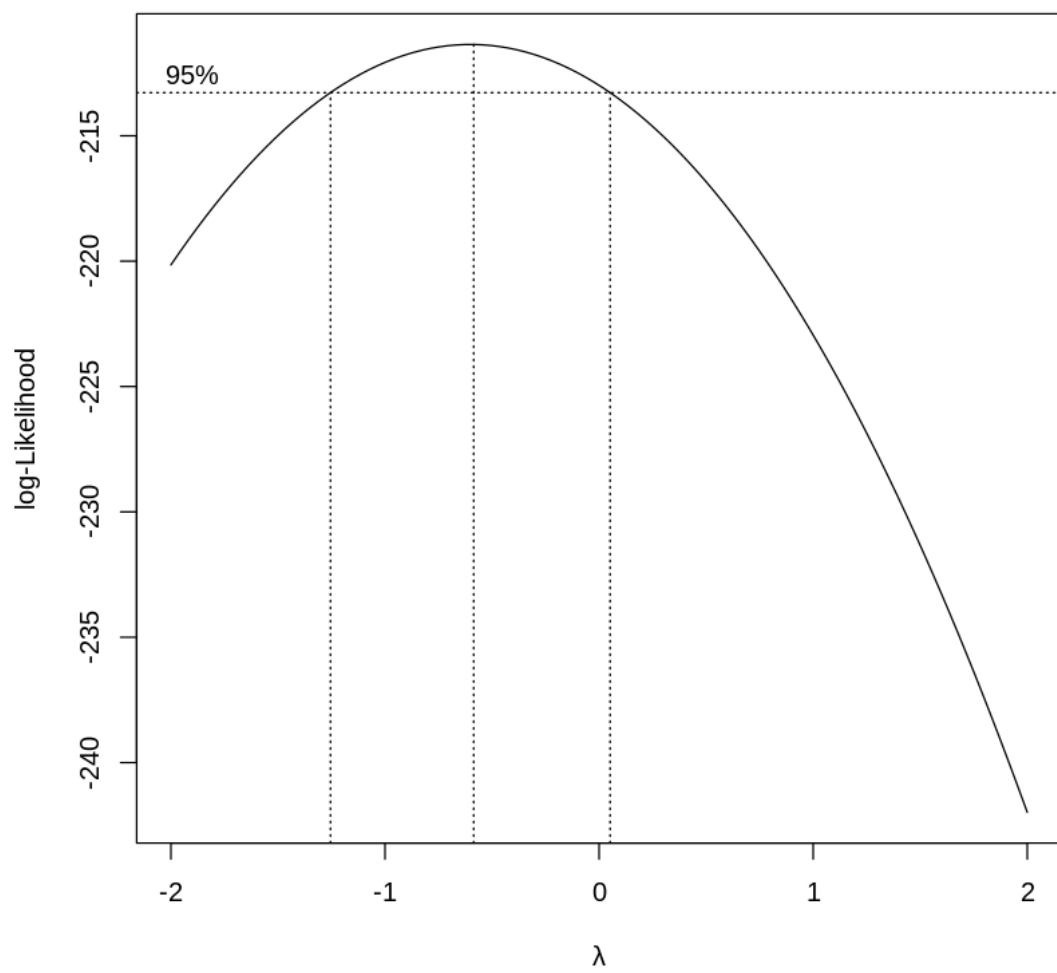
**Histograma de enginesize**



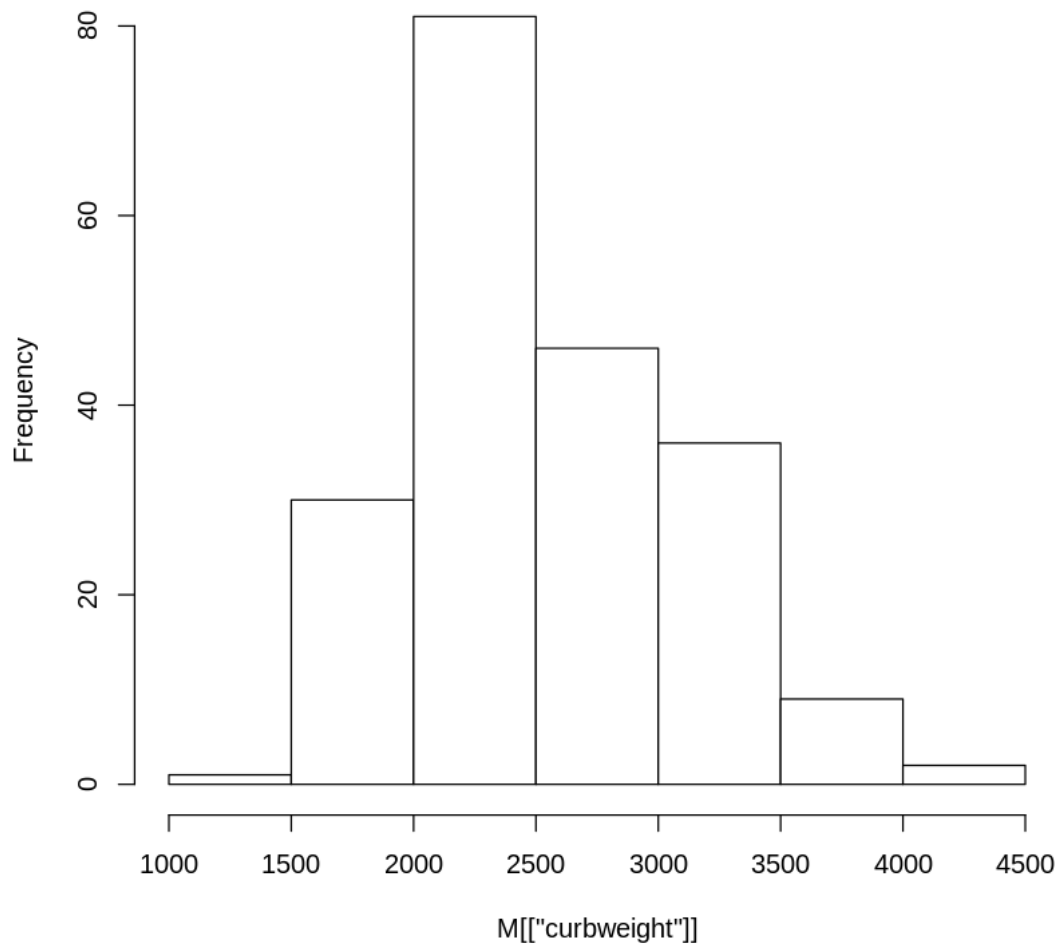


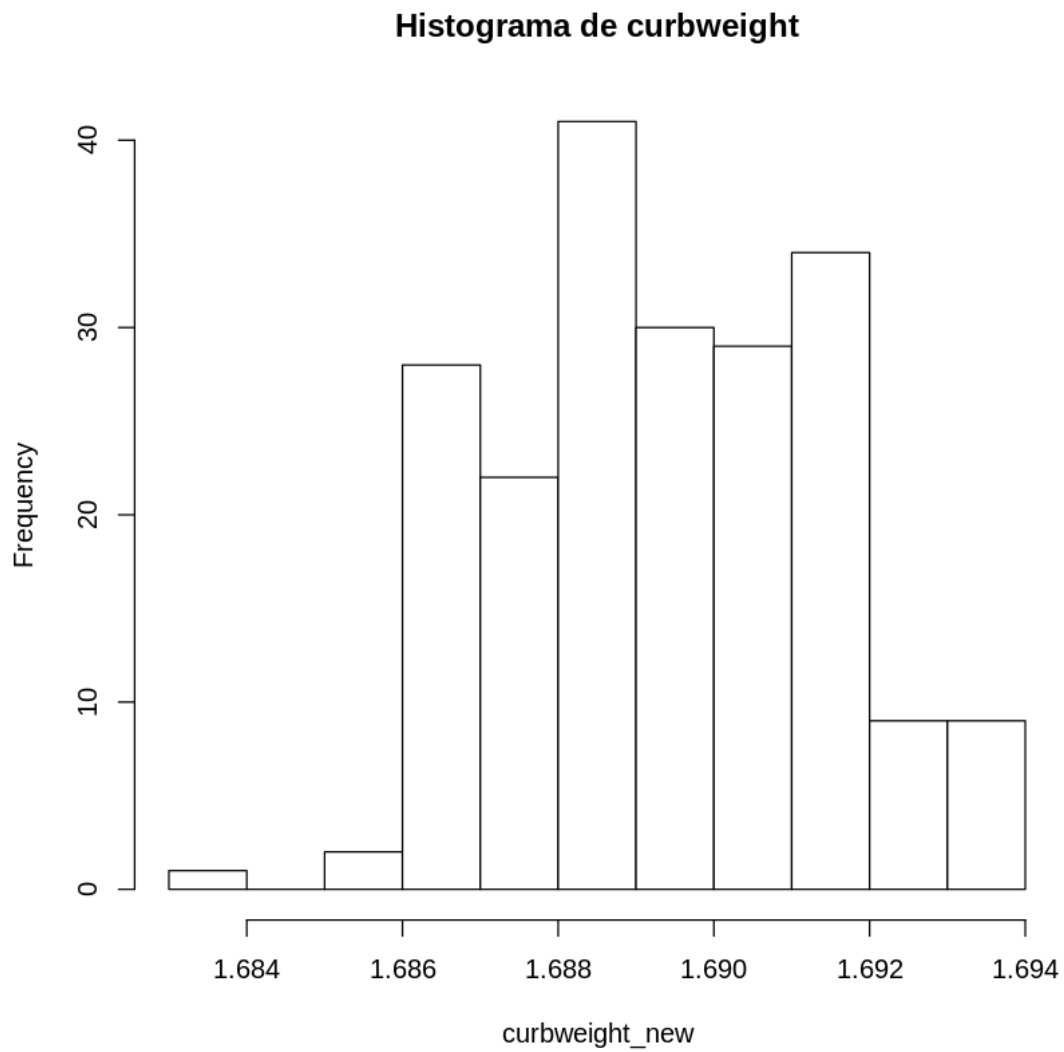


```
[201]: curbweight_new <- normalize_data(M, "curbweight")  
hist(M[["curbweight"]],col=0,main=paste("Histograma de curbweight"))  
hist(curbweight_new,col=0,main=paste("Histograma de curbweight"))
```

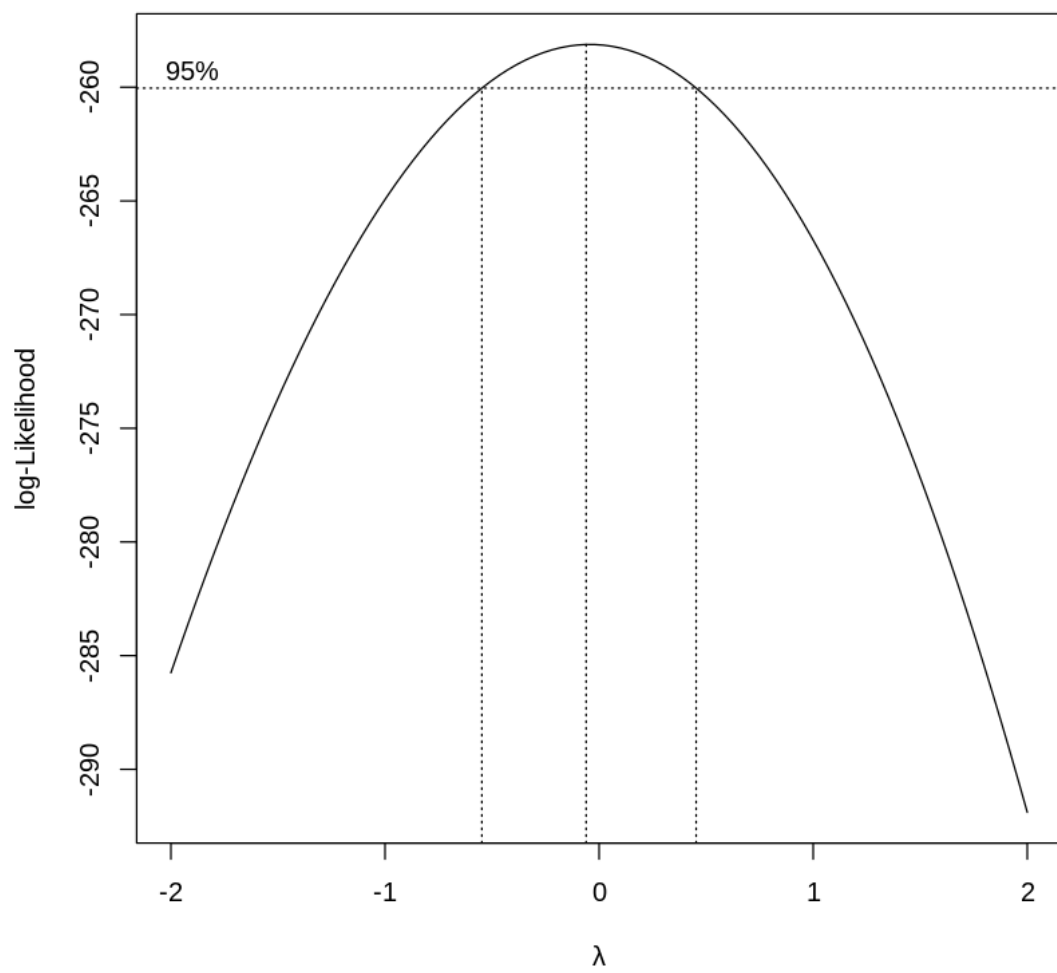


**Histograma de curbweight**

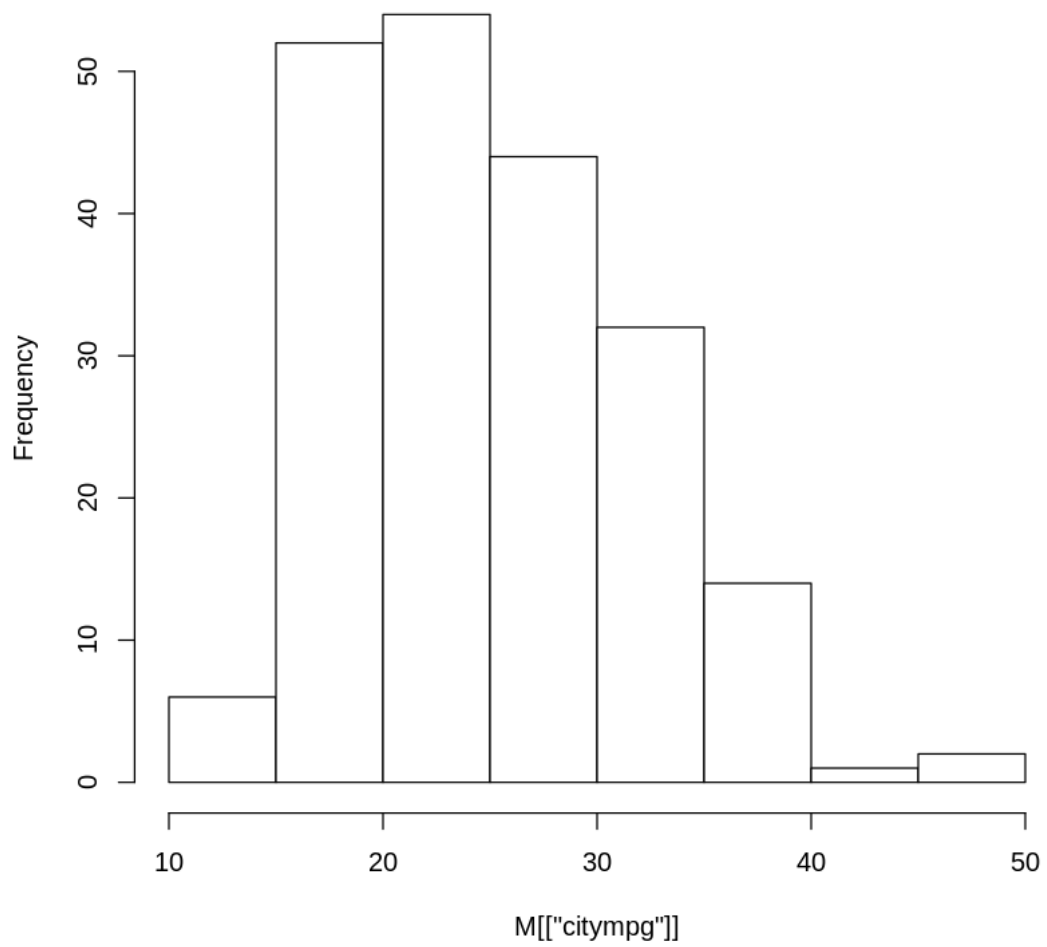


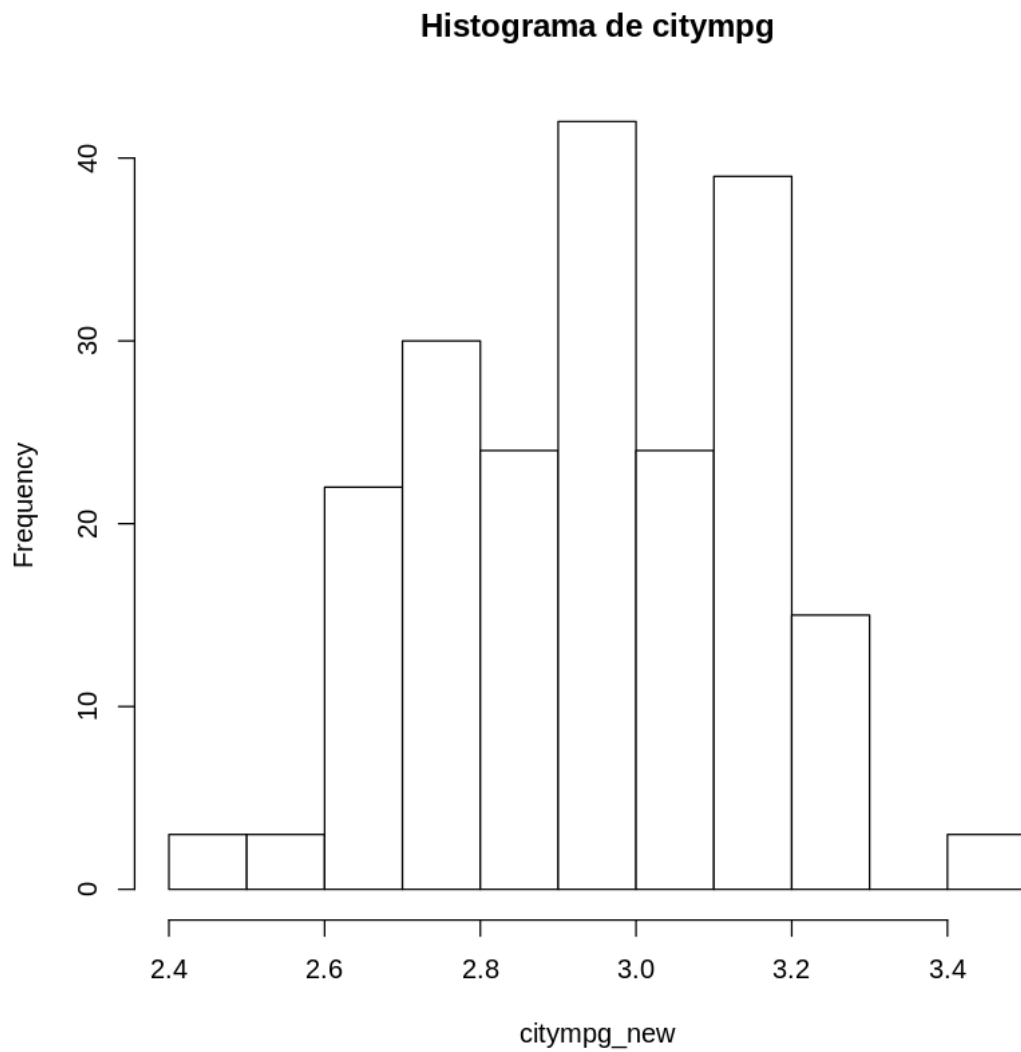


```
[202]: citympg_new <- normalize_data(M, "citympg")
hist(M[["citympg"]],col=0,main=paste("Histograma de citympg"))
hist(citympg_new,col=0,main=paste("Histograma de citympg"))
```

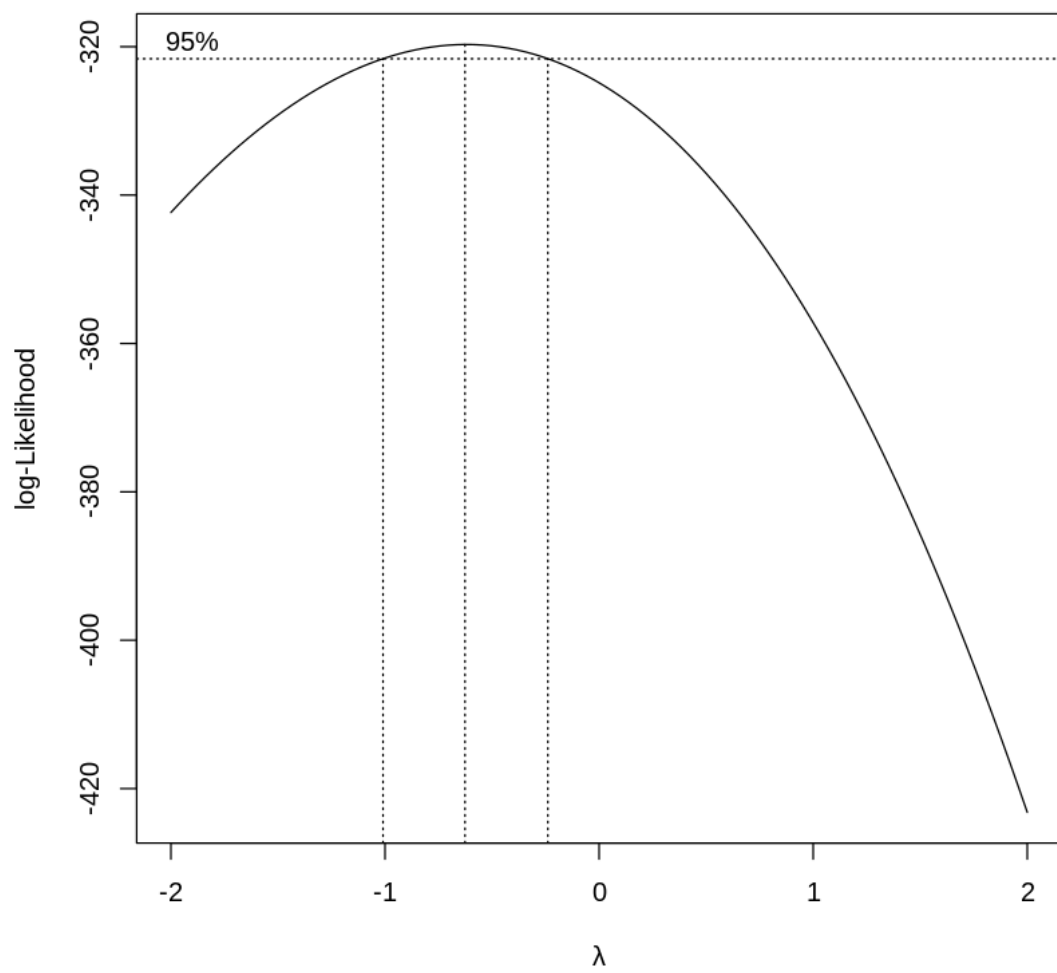


**Histograma de citympg**



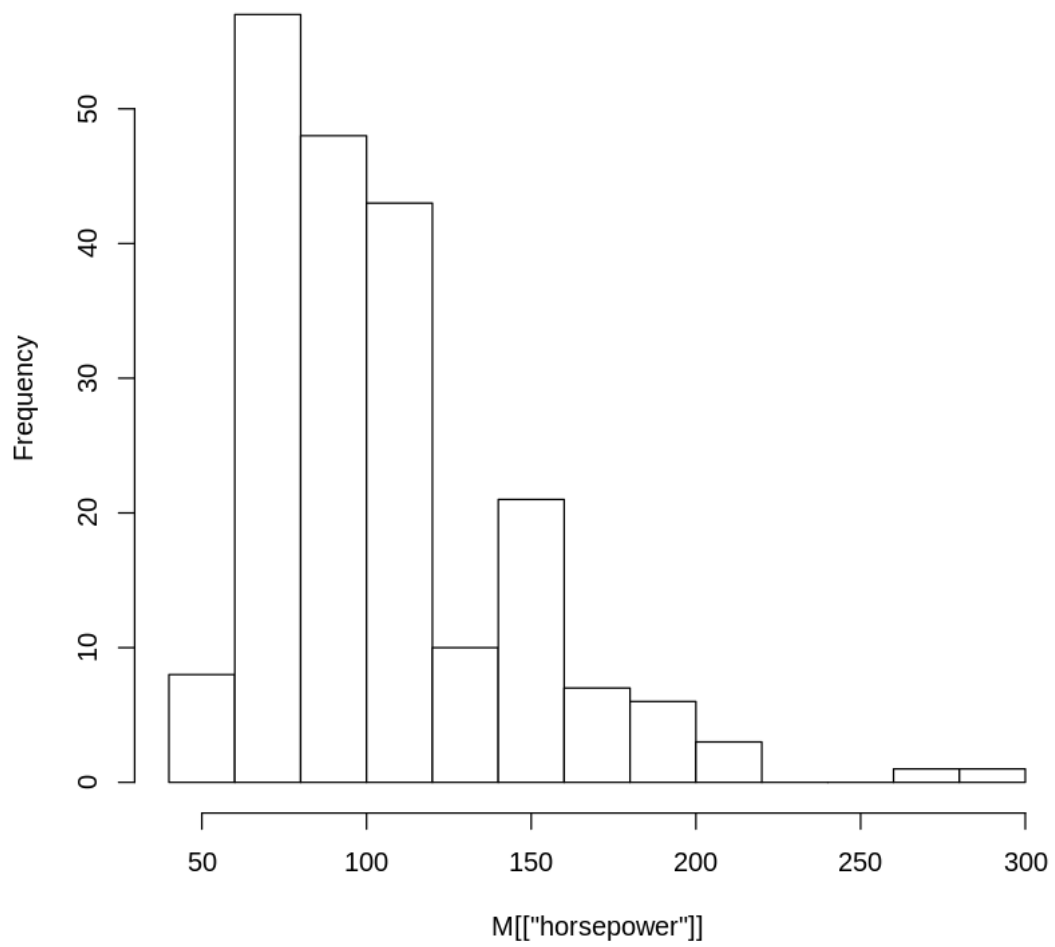


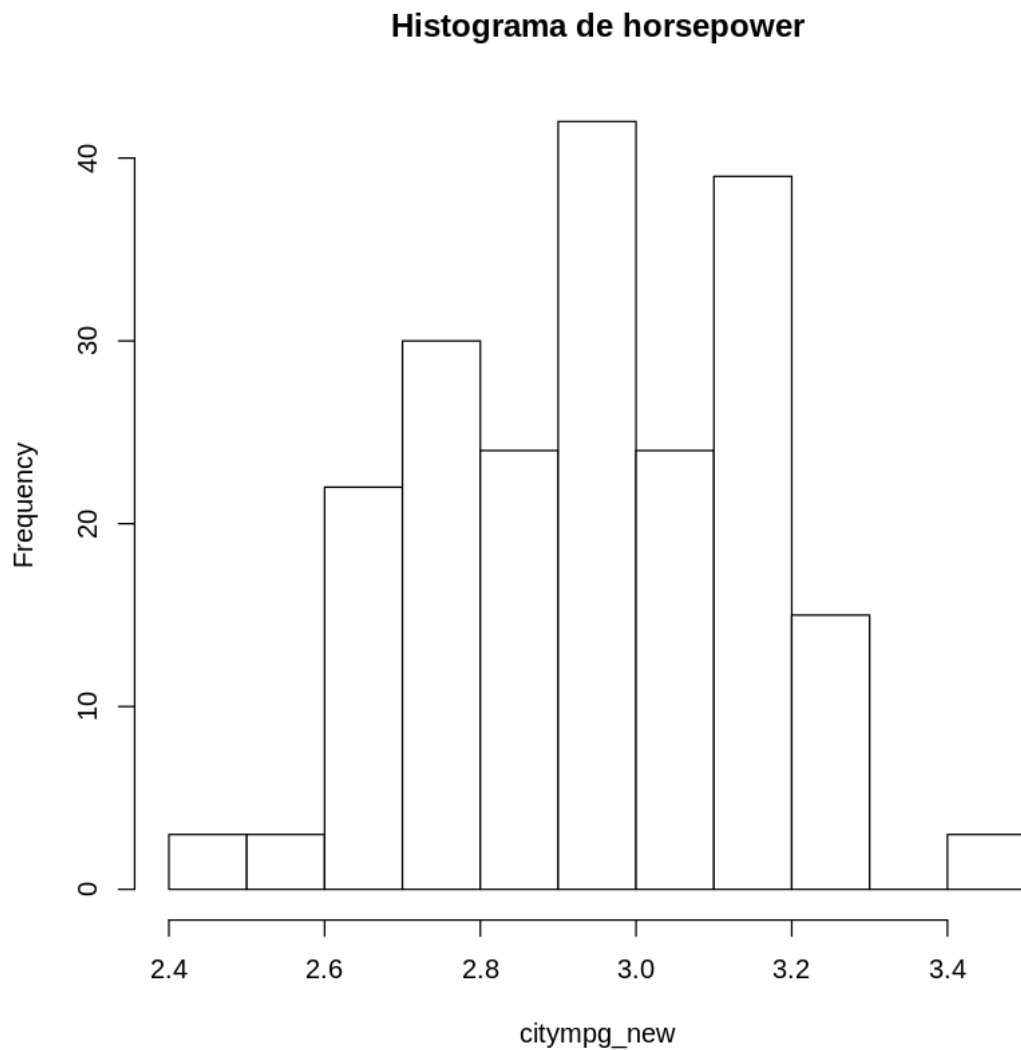
```
[203]: horsepower_new <- normalize_data(M, "horsepower")
hist(M[["horsepower"]],col=0,main=paste("Histograma de horsepower"))
hist(citympg_new,col=0,main=paste("Histograma de horsepower"))
```





**Histograma de horsepower**





Ahora, con nuestros datos ya normalizados, creamos nuevamente el modelo para ver si hubo algún cambio con las variables que parecían ser las más significativas.

```
[204]: B = lm(M$price~M$drivewheel+carwidth_new+enginesize_new+curbweight_new+citympg_new+M$cylindernu
summary(B)
```

Call:

```
lm(formula = M$price ~ M$drivewheel + carwidth_new + enginesize_new +
    curbweight_new + citympg_new + M$cylindernumber + horsepower_new)
```

Residuals:

Min	1Q	Median	3Q	Max
-7918.4	-1407.6	17.6	1193.3	15649.4

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-8.025e+07	4.166e+07	-1.926	0.05557 .
M\$drivewheel	4.815e+02	1.352e+03	0.356	0.72225
M\$drivewheelrwd	3.510e+03	1.400e+03	2.507	0.01303 *
carwidth_new	1.584e+08	8.424e+07	1.881	0.06150 .
enginesize_new	5.101e+05	3.805e+05	1.341	0.18162
curbweight_new	3.292e+05	4.421e+05	0.745	0.45733
citympg_new	-4.961e+03	3.595e+03	-1.380	0.16922
M\$cylindernumberfive	-9.627e+03	1.988e+03	-4.842	2.64e-06 ***
M\$cylindernumberfour	-1.498e+04	1.993e+03	-7.516	2.13e-12 ***
M\$cylindernumbersix	-9.243e+03	1.908e+03	-4.843	2.63e-06 ***
M\$cylindernumberthree	-7.822e+03	4.628e+03	-1.690	0.09267 .
M\$cylindernumbertwelve	-2.499e+03	3.804e+03	-0.657	0.51201
M\$cylindernumbertwo	-1.440e+04	3.688e+03	-3.905	0.00013 ***
horsepower_new	1.615e+04	3.819e+04	0.423	0.67278

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 3448 on 191 degrees of freedom

Multiple R-squared: 0.8256, Adjusted R-squared: 0.8137

F-statistic: 69.56 on 13 and 191 DF, p-value: < 2.2e-16

Podemos ver que la normalización afectó el modelo, y ahora las variables que fueron modificadas ya no aparecen como significativas, como el enginesize y el horsepower. Esto me lleva a pensar que quizás debamos de trabajar con los datos sin normalizar, e ir analizando como se comporta el modelo. Por lo pronto, nos quedaremos con el modelo sin normalización de datos y o analizaremos profundamente.

[205]:

```
A = 
lm(M$price~M$drivewheel+M$carwidth+M$enginesize+M$curbweight+M$citympg+M$cylindernumber+M$horsepower)
summary(A)
```

Call:

```
lm(formula = M$price ~ M$drivewheel + M$carwidth + M$enginesize +
    M$curbweight + M$citympg + M$cylindernumber + M$horsepower)
```

Residuals:

Min	1Q	Median	3Q	Max
-7466.1	-1173.8	52.1	1211.2	14091.2

Coefficients:

Estimate	Std. Error	t value	Pr(> t )
----------	------------	---------	----------

(Intercept)	-23979.136	15314.338	-1.566	0.11905
M\$drivewheelrwd	-437.896	1207.710	-0.363	0.71732
M\$drivewheelrwd	1844.928	1247.453	1.479	0.14080
M\$carwidth	337.824	255.885	1.320	0.18834
M\$enginesize	83.261	17.786	4.681	5.39e-06 ***
M\$curbweight	1.248	1.493	0.835	0.40453
M\$citympg	63.320	71.395	0.887	0.37625
M\$cylindernumberfive	-1475.759	2220.164	-0.665	0.50704
M\$cylindernumberfour	-6000.201	2293.367	-2.616	0.00960 **
M\$cylindernumbersix	-4124.883	1926.720	-2.141	0.03355 *
M\$cylindernumberthree	-2801.141	4233.564	-0.662	0.50899
M\$cylindernumbertwelve	-10001.708	3651.013	-2.739	0.00674 **
M\$cylindernumbertwo	-1907.075	3356.628	-0.568	0.57060
M\$horsepower	43.477	13.062	3.329	0.00105 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3128 on 191 degrees of freedom

Multiple R-squared: 0.8564, Adjusted R-squared: 0.8467



F-statistic: 87.65 on 13 and 191 DF, p-value: < 2.2e-16

En este caso, podemos observar diferentes cosas en nuestro modelo:

1. **Significancia Global:** El valor de la prueba F fue de 87.65, y el valor p global obtenido fue de 2.2e-16, el cual es un valor extremadamente bajo, y nos permite afirmar que rechazamos la hipótesis nula de que todas las variables en el modelo son insignificantes.
2. **Significancia Individual:** En la significancia individual, ya podemos analizar las variables que le dan sentido al modelo, en este caso, las variables mencionadas previamente son las que tienen una mayor probabilidad de describir y definir el modelo, pues tienen valores del estadístico de prueba t relativamente altos, especialmente 4.681 (engine size), y lo que se busca es que estén alejados a 0. Por otro lado, también es esencial fijarse en los valores p de estas variables individuales, pues estamos buscando que los valores p sean muy muy pequeños, lo cual nos indica que es poco probable que los efectos observados sean debido al azar, lo que a su vez sugiere que la variable es un predictor significativo de la variable dependiente.
3. **R-cuadrada ajustada:** En este caso, obtuvimos un valor de 0.84 como el coeficiente de determinación, y nos indica que el modelo se está ajustando correctamente a los datos, debido a que está cerca del valor ideal o máximo, el cual es 1. Esto nos indica que vamos por buen camino.

Para simplificar el modelo, voy a intentar eliminar poco a poco algunas de las variables que no nos son útiles en este modelo, debido a sus valores p y el análisis de los estadísticos de t.

Primeramente, probaré eliminando la variable de “drivewheel”, la cual es una de las que tiene el número más alto de valor - p, y tiene también un número muy cercano a 0 de t.

[206]: A =   
 lm(M\$price~M\$carwidth+M\$enginesize+M\$curbweight+M\$citympg+M\$cylindernumber+M\$horsepower)

```
summary(A)
```

Call:

```
lm(formula = M$price ~ +M$carwidth + M$enginesize + M$curbweight +  
    M$citympg + M$cylindernumber + M$horsepower)
```

Residuals:

Min	1Q	Median	3Q	Max
-9011.2	-1372.3	42.1	1139.7	13928.2

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-22690.136	15130.667	-1.500	0.135349
M\$carwidth	228.899	247.698	0.924	0.356587
M\$enginesize	83.812	17.680	4.740	4.13e-06 ***
M\$curbweight	3.092	1.340	2.308	0.022042 *
M\$citympg	84.871	72.223	1.175	0.241392
M\$cylindernumberfive	-1702.596	2261.905	-0.753	0.452531
M\$cylindernumberfour	-5582.764	2294.516	-2.433	0.015882 *
M\$cylindernumbersix	-3736.831	1925.543	-1.941	0.053756 .
M\$cylindernumberthree	-2023.319	4283.629	-0.472	0.637219
M\$cylindernumbertwelve	-10987.563	3720.745	-2.953	0.003537 **
M\$cylindernumbertwo	335.521	3291.149	0.102	0.918905
M\$horsepower	49.149	13.237	3.713	0.000268 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3200 on 193 degrees of freedom

Multiple R-squared: 0.8482, Adjusted R-squared: 0.8396

F-statistic: 98.05 on 11 and 193 DF, p-value: < 2.2e-16

La variable de carwidth, tiene un valor de p de 0.35, el cual es uno de los más altos que hay en el modelo y además su valor de t (0.924) se encuentra muy cercano a 0, por lo que al parecer esta variable tampoco es muy útil para incluirla en nuestro modelo. Se eliminará también.

```
[207]: A =  
      ↪ lm(M$price~+M$enginesize+M$curbweight+M$citympg+M$cylindernumber+M$horsepower)  
      summary(A)
```

Call:

```
lm(formula = M$price ~ +M$enginesize + M$curbweight + M$citympg +  
    M$cylindernumber + M$horsepower)
```

Residuals:

Min	1Q	Median	3Q	Max
-----	----	--------	----	-----

```
-9094.1 -1256.5    21.4  1122.9 13913.0
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	-9429.583	4796.558	-1.966	0.050736	.
M\$engine size	83.372	17.667	4.719	4.53e-06	***
M\$curbweight	3.876	1.037	3.738	0.000244	***
M\$citympg	92.751	71.690	1.294	0.197282	
M\$cylindernumberfive	-1755.648	2260.325	-0.777	0.438268	
M\$cylindernumberfour	-6049.961	2237.282	-2.704	0.007456	**
M\$cylindernumbersix	-4353.154	1805.670	-2.411	0.016849	*
M\$cylindernumberthree	-3051.542	4135.041	-0.738	0.461424	
M\$cylindernumbertwelve	-11474.068	3681.924	-3.116	0.002109	**
M\$cylindernumbertwo	-14.571	3268.041	-0.004	0.996447	
M\$horsepower	50.418	13.161	3.831	0.000172	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3199 on 194 degrees of freedom

Multiple R-squared: 0.8475, Adjusted R-squared: 0.8397

F-statistic: 107.9 on 10 and 194 DF, p-value: < 2.2e-16

Por el momento, continuaremos el análisis y la validación con esta última versión del modelo.

Para verificar la normalidad de los residuos, instalaremos la librería de `nortest` para hacer la prueba de anderson darling.

```
[208]: install.packages('nortest')
       library(nortest)
```

Installing package into ‘/usr/local/lib/R/site-library’  
(as ‘lib’ is unspecified)

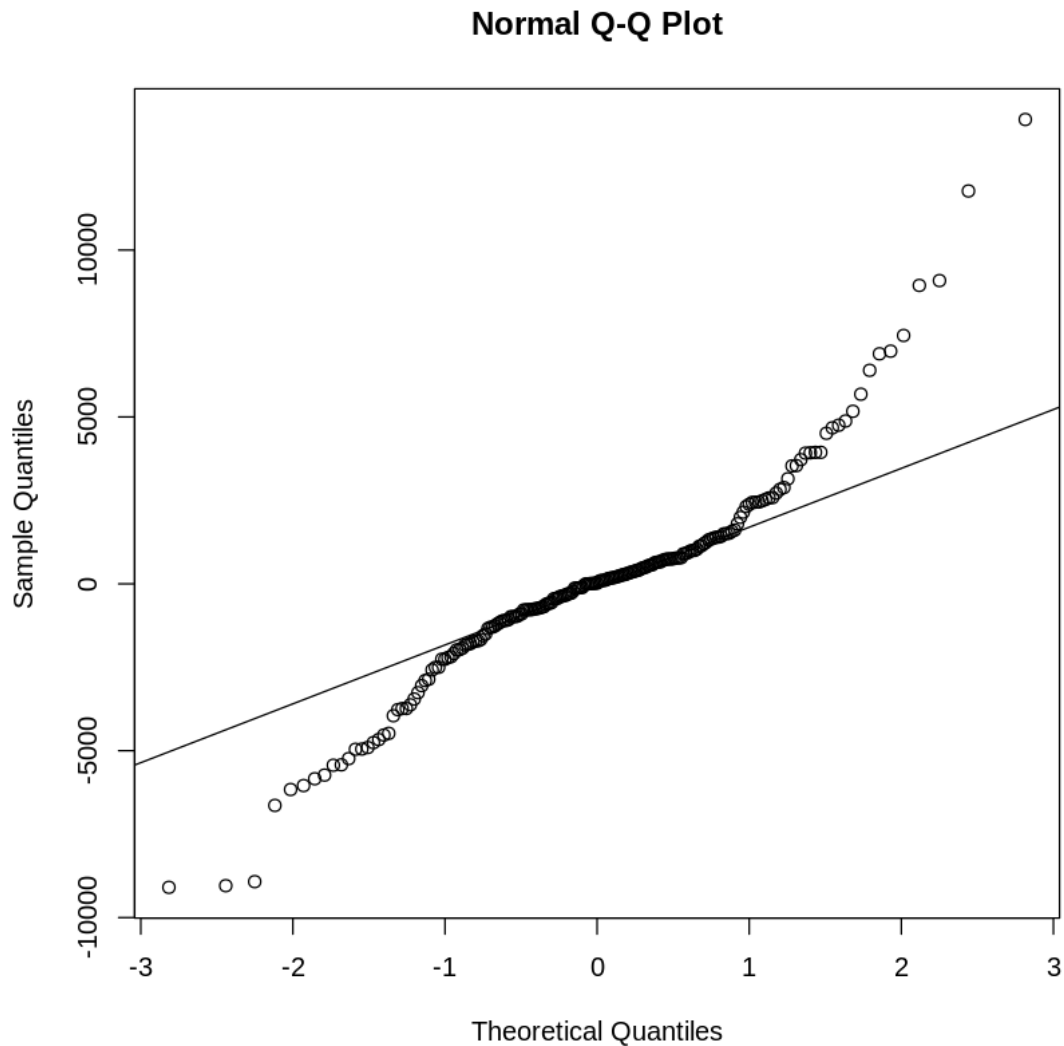
```
[209]: ad.test(A$residuals)
```

Anderson-Darling normality test

data: A\$residuals

A = 4.5186, p-value = 3.041e-11

```
[210]: qqnorm(A$residuals)
       qqline(A$residuals)
```



En este caso, para el modelo, se puede observar que los residuos no cumplen con normalidad en su distribución, esto principalmente porque el valor  $p$  que obtuvimos en la prueba de anderson darling es muy pequeño, cuando en este caso, lo que buscamos es que este valor sea grande, para poder comprobar la hipótesis de que nuestros datos siguen una distribución normal.

De igual forma, en la qqplot que se gráfico, se puede observar que los datos no siguen la recta esperada, en donde se trata de una distribución normal, y esto también se ve reflejado en el valor  $p$ .

Para la próxima entrega, se busca poder realizar transformaciones o modificaciones a la selección de variables para poder llegar a que nuestros residuos se comporten de manera normal.

Además, después se buscan también realizar pruebas para verificar que la media sea 0, y revisar que exista independencia y homeostacidad en los datos.