# Entregable 2, Blanquita

Francisco Castorena, A00827756

2023-08-25

## Exploración y limpieza de la base de datos

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

```
# Observamos dimensiones de la base de datos
dim(df)
```

```
## [1] 205  21
```

La base de datos tiene 205 observaciones y 21 columnas.

```
# Observamos el tipo de dato que se tiene para cada una de las columnas en nuestro dataset
sapply(df, class)
```

```
##        symboling          CarName         fueltype          carbody
##        "integer"      "character"      "character"      "character"
##       drivewheel   enginelocation        wheelbase        carlength
##      "character"      "character"        "numeric"        "numeric"
##         carwidth        carheight       curbweight       enginetype
##        "numeric"        "numeric"        "integer"      "character"
##   cylindernumber       enginesize           stroke compressionratio
##      "character"        "integer"        "numeric"        "numeric"
##       horsepower          peakrpm          citympg       highwaympg
##        "integer"        "integer"        "integer"        "integer"
##            price
##        "numeric"
```

Observamos que muchas de las variables son categóricas con valores numéricos, al analizar la base de datos se puede observar que la variable 'cylindernumber' tiene valores en cadenas de texto, pasaremos a valores numéricos. (No es necesario por el momento hacer este cambio, sin embargo prefiero ver los números a sus nombres en inglés).

```
# Convirtiendo variable de texto categóricas a numéricas categóricas
cylinder_levels <- c("one","two","three","four", "five","six","seven","eight","twelve")
cylinder_numbers <- c(1,2,3,4,5,6,7,8,12)

# Convert the column to a factor with the defined levels
df$cylindernumber <- factor(df$cylindernumber, levels = cylinder_levels)
```

```r
# Map the factor levels to their corresponding numbers
df$cylindernumber <- cylinder_numbers[as.integer(df$cylindernumber)]

# Transformamos las columnas symboling y cylindernumber a variables categóricas, para que no sean interp
df$symboling <- factor(df$symboling)
df$cylindernumber <- factor(df$cylindernumber)
```

Buscamos valores nulos dentro del dataset.

```r
colSums(is.na(df))
```

```
##        symboling          CarName         fueltype          carbody
##                0                0                0                0
##       drivewheel   enginelocation        wheelbase        carlength
##                0                0                0                0
##         carwidth        carheight       curbweight       enginetype
##                0                0                0                0
##   cylindernumber       enginesize           stroke compressionratio
##                0                0                0                0
##       horsepower          peakrpm          citympg       highwaympg
##                0                0                0                0
##            price
##                0
```

No se tienen valores nulos en el dataset.

**Analizamos distribución de los datos para cada variable numérica**

```r
# Filter out non-numeric columns
numeric_cols <- sapply(df, is.numeric)
numeric_df <- df[, numeric_cols]

#adjust plot margins
#par(mar = c(0.5,0.5,0.5,0.5))
library(Hmisc)
```

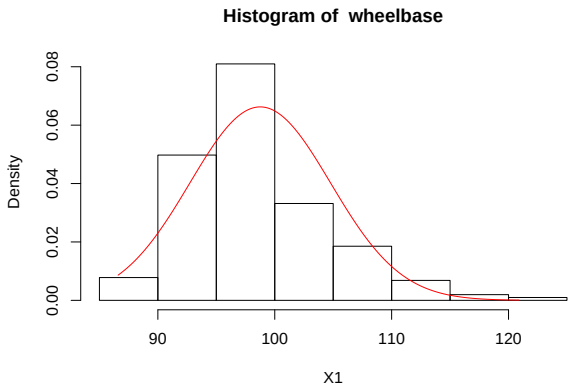
```
## Warning: package 'Hmisc' was built under R version 4.2.3
```

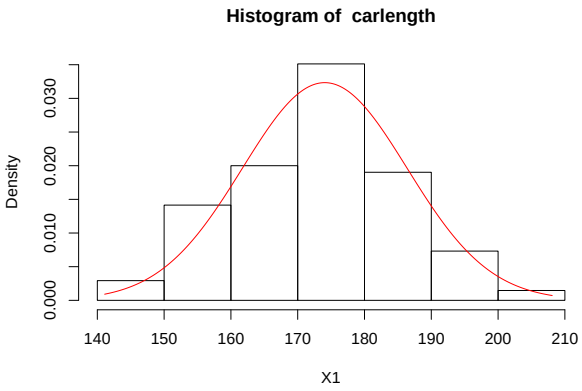
```
##
## Attaching package: 'Hmisc'
```

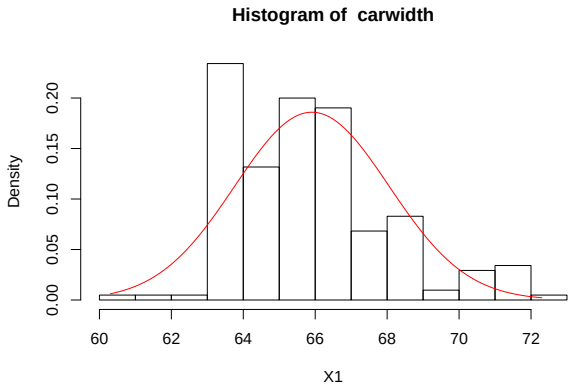
```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

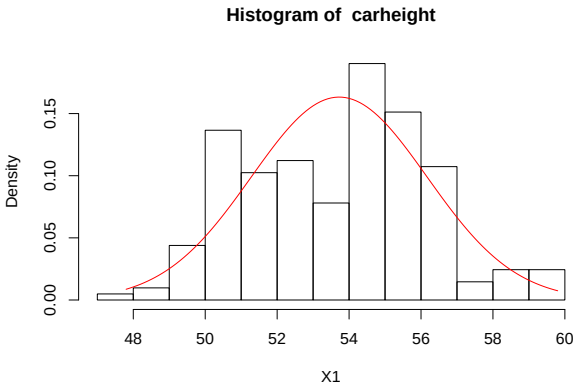
```r
for (col_name in names(numeric_df)) {
  X1 = numeric_df[[col_name]]
  title_ = paste('Histogram of ',col_name)
  hist(X1,prob=TRUE,col=0,main=title_)
```

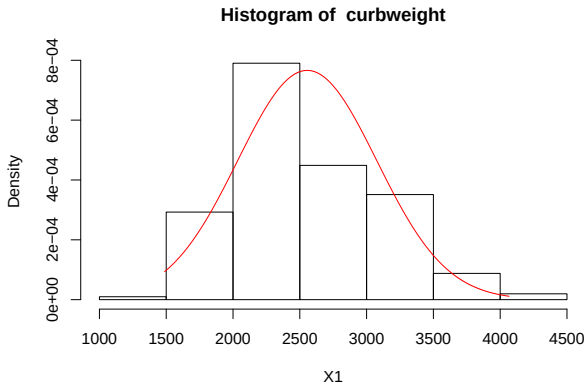
```
  x=seq(min(X1),max(X1),0.1)
  y=dnorm(x,mean(X1),sd(X1))
  lines(x,y,col="red")
}
```

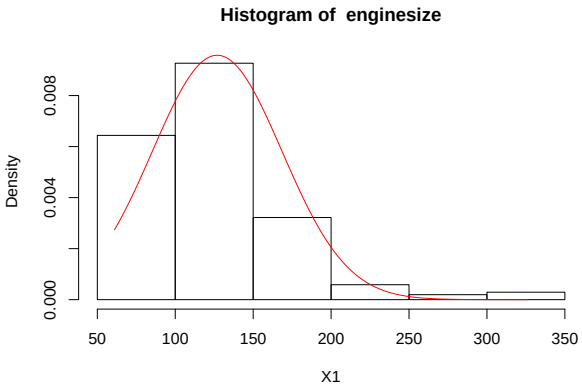


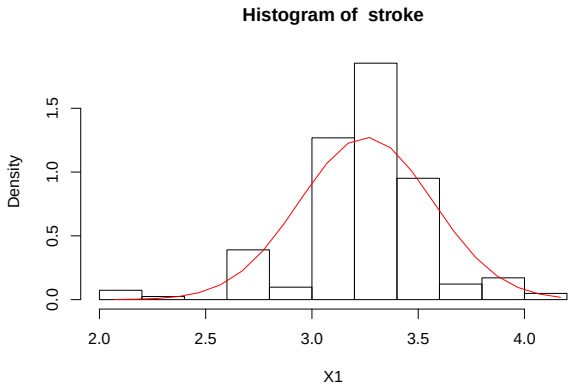



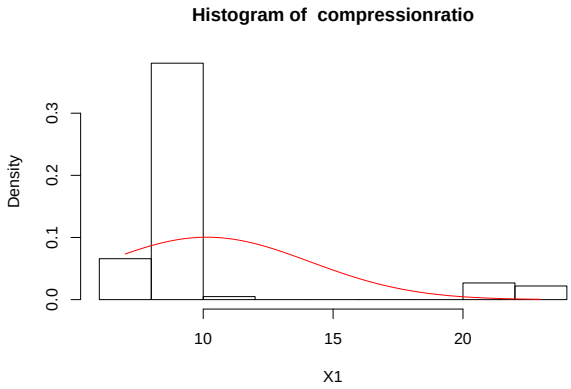



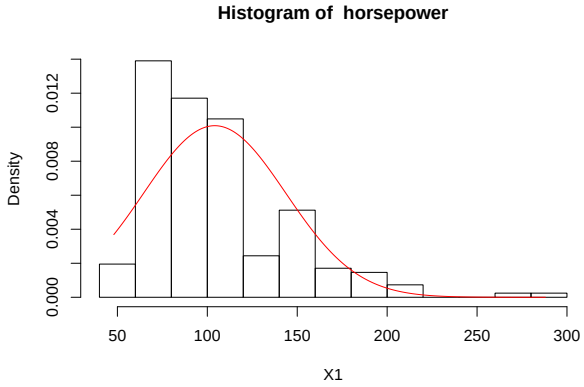



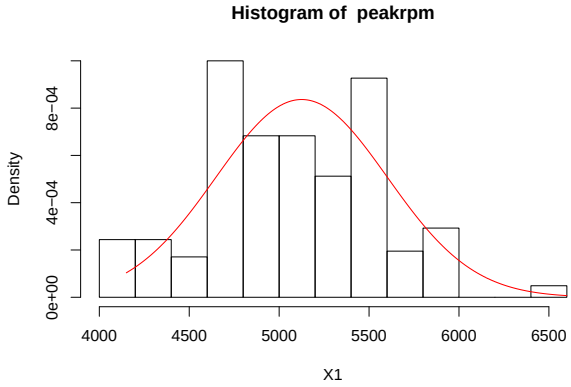



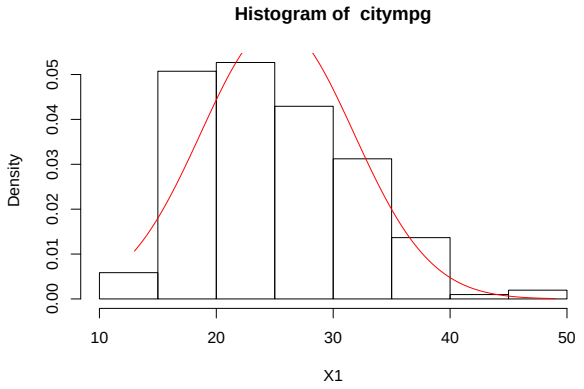



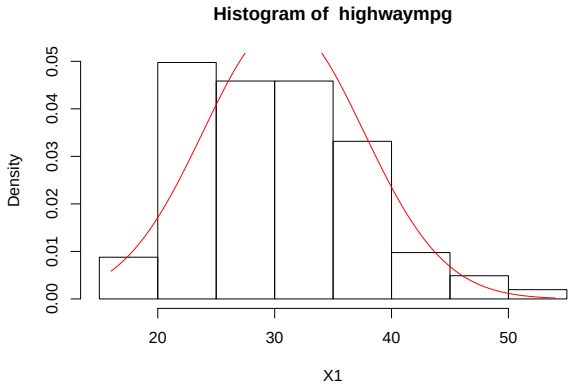

Histogram of stroke
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
X1


Histogram of compressionratio
Density
0.0
0.1
0.2
0.3
10
15
20
X1


Histogram of horsepower
Density
0.000
0.004
0.008
0.012
50
100
150
200
250
300
X1


Histogram of peakrpm
Density
0e+00
4e−04
8e−04
4000
4500
5000
5500
6000
6500
X1


Histogram of citympg
Density
0.00
0.01
0.02
0.03
0.04
0.05
10
20
30
40
50
X1


Histogram of highwaympg
Density
0.00
0.01
0.02
0.03
0.04
0.05
20
30
40
50
X1


Podemos observar como existen distintas variables que parecen seguir una distribución normal, como lo podrían ser 'carlength', 'carwidth', 'curbweight' entre otras. También existen variables con sesgo como por ejemplo 'price', 'horsepower' o 'compressionratio'.

**Medidas estadísticas apropiadas para las variables cuantitativas (media, desviación estándar, cuantiles, etc)**

```
quantitative_vars <- c("wheelbase", "carlength", "carwidth", "carheight", "curbweight", "enginesize", "
for (col_name in quantitative_vars) {
  X1 = df[[col_name]]
  print(col_name)
  print(summary(X1))
  cat("\n")
}
```

```
## [1] "wheelbase"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   86.60   94.50   97.00   98.76  102.40  120.90
##
## [1] "carlength"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   141.1   166.3   173.2   174.0   183.1   208.1
##
## [1] "carwidth"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.30   64.10   65.50   65.91   66.90   72.30
##
## [1] "carheight"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   47.80   52.00   54.10   53.72   55.50   59.80
##
## [1] "curbweight"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1488    2145    2414    2556    2935    4066
##
## [1] "enginesize"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##    61.0    97.0   120.0   126.9   141.0   326.0
##
## [1] "stroke"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.070   3.110   3.290   3.255   3.410   4.170
##
## [1] "compressionratio"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00    8.60    9.00   10.14    9.40   23.00
##
## [1] "horsepower"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    48.0    70.0    95.0   104.1   116.0   288.0
##
## [1] "peakrpm"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4150    4800    5200    5125    5500    6600
##
## [1] "citympg"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.00   19.00   24.00   25.22   30.00   49.00
##
## [1] "highwaympg"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.00   25.00   30.00   30.75   34.00   54.00
##
## [1] "price"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5118    7788   10295   13277   16503   45400
```

Podemos ver a primera instancia que los datos mínimos y máximos en muchas variables no distan demasiado, sobre todo en aquellas variables que describen las dimensiones de los carros, lo cual es algo esperado debido a la naturaleza de los datos, se entiende que muchas de las variables descriptivas se comporten de manera "normal", ya que todo carro por distinto que sea el precio, el tamaño, motor etc. debe de cumplir con ciertas normas y lineamientos para poder transitar de forma correcta.

**Medidas estadísticas apropiadas para las variables cualitativas (cuantiles, frecuencias).**

No se va a tener en cuenta a la variable "CarName" para el análisis de frecuencias y proporciones, debido a que son demasiados valores categóricos como para poder tener un análisis claro por tipo de valor.

```
qualitative_vars <- c("symboling","fueltype","carbody","drivewheel","enginelocation","enginetype")

for (col_name in qualitative_vars) {
  # Cálculo de medidas estadísticas
  X1 = df[[col_name]]
  frecuencias <- table(X1)
  proporciones <- prop.table(frecuencias)
  cuantiles <- quantile(as.numeric(factor(X1)), probs = c(0.25, 0.5, 0.75))

  # Imprimir resultados
  cat("\n")
  cat("Frecuencias de ", col_name,":\n")
```

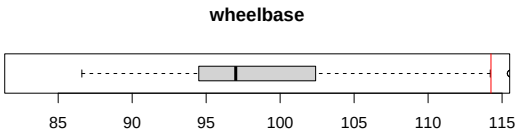
```
  print(frecuencias)
  cat("\n")
  cat("Proporciones de ", col_name,":\n")
  print(proporciones)
  cat("\n")
  cat("Cuantiles de ", col_name,":\n")
  print(cuantiles)
  cat("\n")

}
```

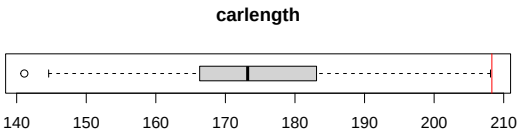
```
##
## Frecuencias de  symboling :
## X1
## -2 -1  0  1  2  3
##  3 22 67 54 32 27
##
## Proporciones de  symboling :
## X1
##         -2         -1          0          1          2          3
## 0.01463415 0.10731707 0.32682927 0.26341463 0.15609756 0.13170732
##
## Cuantiles de  symboling :
## 25% 50% 75%
##   3   4   5
##
##
## Frecuencias de  fueltype :
## X1
## diesel    gas
##     20    185
##
## Proporciones de  fueltype :
## X1
##     diesel        gas
## 0.09756098 0.90243902
##
## Cuantiles de  fueltype :
## 25% 50% 75%
##   2   2   2
##
##
## Frecuencias de  carbody :
## X1
## convertible     hardtop   hatchback       sedan       wagon
##           6           8          70          96          25
##
## Proporciones de  carbody :
## X1
## convertible     hardtop   hatchback       sedan       wagon
##  0.02926829  0.03902439  0.34146341  0.46829268  0.12195122
##
## Cuantiles de  carbody :
```

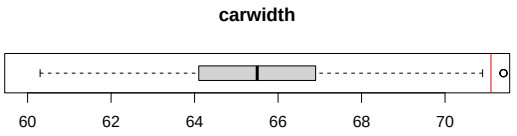
```
## 25% 50% 75%
##   3   4   4
##
##
## Frecuencias de  drivewheel :
## X1
## 4wd fwd rwd
##   9 120  76
##
## Proporciones de  drivewheel :
## X1
##        4wd        fwd        rwd
## 0.04390244 0.58536585 0.37073171
##
## Cuantiles de  drivewheel :
## 25% 50% 75%
##   2   2   3
##
##
## Frecuencias de  enginelocation :
## X1
## front  rear
##   202     3
##
## Proporciones de  enginelocation :
## X1
##      front       rear
## 0.98536585 0.01463415
##
## Cuantiles de  enginelocation :
## 25% 50% 75%
##   1   1   1
##
##
## Frecuencias de  enginetype :
## X1
##  dohc dohcv     l   ohc  ohcf  ohcv rotor
##    12     1    12   148    15    13     4
##
## Proporciones de  enginetype :
## X1
##        dohc       dohcv           l         ohc        ohcf        ohcv
## 0.058536585 0.004878049 0.058536585 0.721951220 0.073170732 0.063414634
##       rotor
## 0.019512195
##
## Cuantiles de  enginetype :
## 25% 50% 75%
##   4   4   4
```

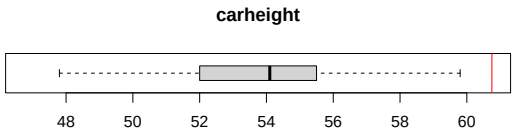
### Boxplots para variables cuantitativas

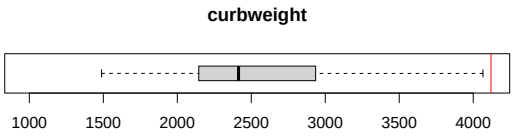
```
for (col_name in quantitative_vars){
  X = df[[col_name]]
  q1=quantile(X,0.25)  #Cuantil 1 de la variable X
  q3=quantile(X,0.75)
  ri= q3-q1   #Rango intercuartílico de X
  par(mfrow=c(2,1)) #Matriz de gráficos de 2x1
  y1 = q1-1.5*ri
  y2 = q3+1.5*ri
  boxplot(X,horizontal=TRUE,ylim=c(y1,y2),main=col_name)
  abline(v=q3+1.5*ri,col="red")  #linea vertical en el límite de los datos atípicos o extremos
}
```

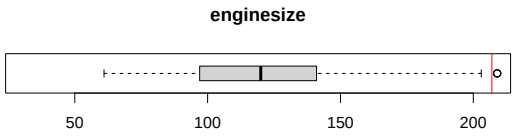


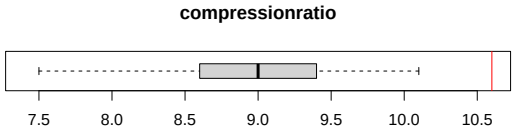



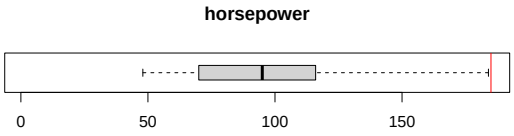



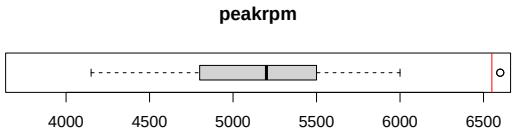

curbweight
1000
1500
2000
2500
3000
3500
4000

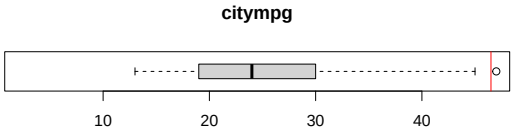

enginesize
50
100
150
200

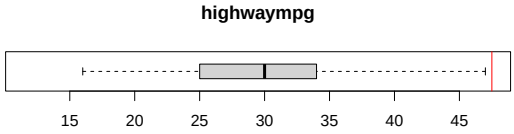

stroke
2.8
3.0
3.2
3.4
3.6
3.8


compressionratio
7.5
8.0
8.5
9.0
9.5
10.0
10.5


horsepower
0
50
100
150


peakrpm
4000
4500
5000
5500
6000
6500





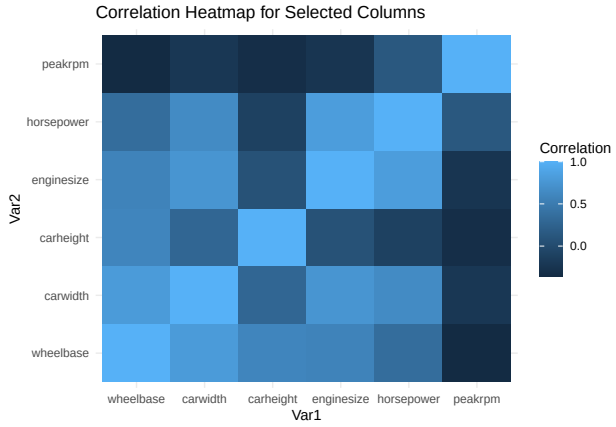
Con los boxplots generados para las variables cuantitativas podemos observar como el grupo del cuantil 2 y el grupo del cuantil 3 son más pequeños que los de los demás cuantiles, lo cual indica que hay una menor variación de los datos en estos cuantiles, y va de acorde a la distribución normal, los datos suelen estar mayormente distribuidos cerca de la mediana en este caso, por otra parte podemos ver que hay muy pocos outliers, esto como se explicó anteriormente puede ser debido a la naturaleza de los datos que se están analizando, no puedes cambiar mucho ciertas características de estos para que sigan siendo útiles.

### Analisis de correlación y diagramas de dispersión

Debido a la cantidad de variables dentro del dataframe, el mostrar una matriz de correlación o de diagramas de dispersión podía ser engorroso y no verse bien, por lo que seleccionamos solo algunas de las variables para desplegar en la gráfica, de igual forma, la matriz númerica de correlaciones completa se encuentra en la variable corr_mat

```
library(ggplot2)

selected_columns <- c(1, 3, 4, 6, 9, 10)
subset_df <- numeric_df[, selected_columns]

cor_matrix <- cor(subset_df)

cor_data <- as.data.frame(as.table(cor_matrix))
colnames(cor_data) <- c("Var1", "Var2", "Correlation")

# Create a correlation heatmap using ggplot2
```

```r
ggplot(data = cor_data, aes(x = Var1, y = Var2, fill = Correlation)) +
  geom_tile() +
  theme_minimal() +
  labs(title = "Correlation Heatmap for Selected Columns")
```




A continuación se muestran las diez correlaciones mayores que se tienen para nuestro dataframe numérico (excluyendo la diagnoal principal), debido a que por las dimensiones de la matriz identificar estos valores a vista puede ser dificil.

Esta parte de código se realizó con ayuda de ChatGPT.

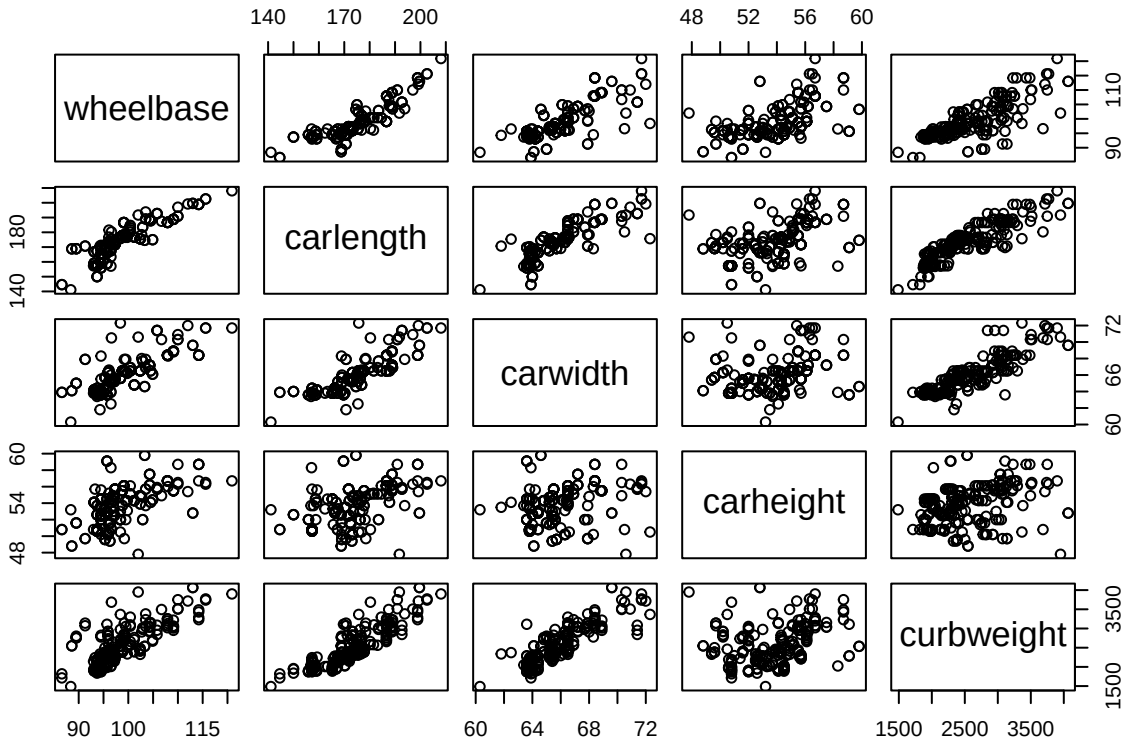
```r
# Replace these with your actual correlation matrix and variable names
cor_matrix <- cor(numeric_df)
variable_names <- colnames(numeric_df)

# Set the diagonal values to NA to exclude them
diag(cor_matrix) <- NA

# Flatten the upper triangle of the correlation matrix
cor_flat <- cor_matrix[upper.tri(cor_matrix, diag = TRUE)]

# Order the correlations in descending order
sorted_cor <- sort(cor_flat, decreasing = TRUE)

# Select the top 10 highest correlated pairs
top_pairs <- head(sorted_cor, 10)

# Get the indices of the top pairs
top_indices <- order(cor_flat, decreasing = TRUE)[1:10]

# Get the variable names for the top pairs
variable_pairs <- expand.grid(Var1 = variable_names, Var2 = variable_names)
top_variable_pairs <- variable_pairs[top_indices, ]

# Print the top 10 highest correlated pairs along with their correlations
result <- cbind(top_variable_pairs, Correlation = top_pairs)
print(result)
```

```
##          Var1       Var2 Correlation
```

```
## 77 highwaympg enginesize   0.9713370
## 12 highwaympg  wheelbase   0.8777285
## 2   carlength  wheelbase   0.8745875
## 84 enginesize     stroke   0.8741448
## 13      price  wheelbase   0.8670325
## 20     stroke  carlength   0.8505941
## 5  curbweight  wheelbase   0.8411183
## 83 curbweight     stroke   0.8353049
## 42   carwidth  carheight   0.8097687
## 87 horsepower     stroke   0.8081388
```

De igual forma, a continuación se incluyen algunos diagramas de dispersión (no se incluyen todos por dimensiones de la matriz)

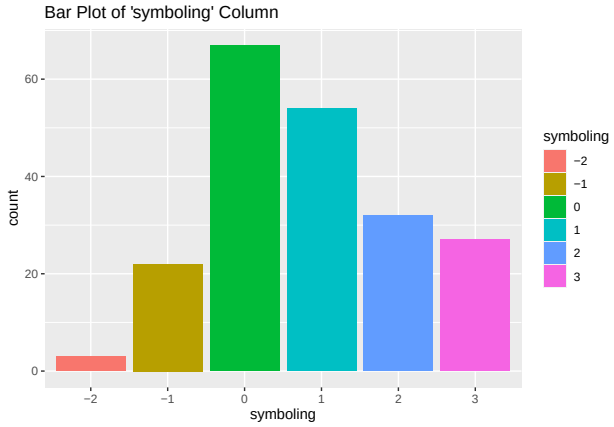
```r
# Select the first 5 variables from numeric_df
selected_vars <- numeric_df[, 1:5]

# Create the dispersion matrix plot
pairs(selected_vars)
```



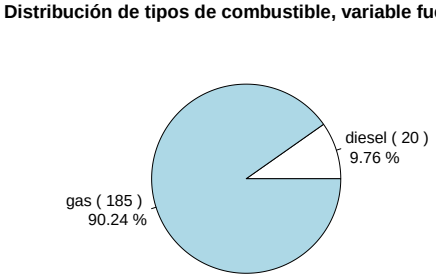

```r
# Creamos dataset de variables categoricas
qualitative_vars <- c("symboling","fueltype","carbody","drivewheel","enginelocation","enginetype")
categorical_df <- df[, qualitative_vars]
```

**Analizamos por medio de gráficas, distribución de datos categóricos** Diagramas de barras

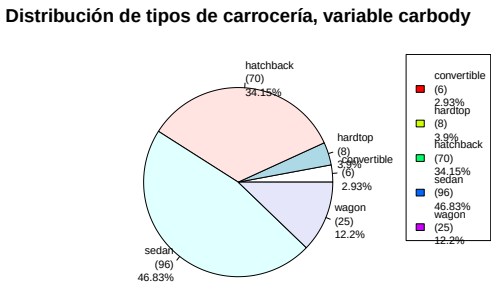
```
# Convert 'symboling' to a factor for categorical interpretation
categorical_df$symboling <- factor(categorical_df$symboling)

# Create a bar plot for 'symboling'
bar_plot <- ggplot(categorical_df, aes(x = symboling, fill = symboling)) +
  geom_bar() +
  labs(title = "Bar Plot of 'symboling' Column")

print(bar_plot)
```



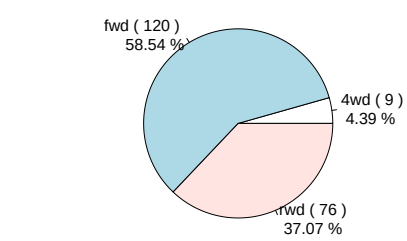

```
conteo <- table(categorical_df$fueltype)

# Calcular el porcentaje
porcentaje <- prop.table(conteo) * 100

etiquetas <- paste(names(conteo), "(", conteo, ")", "\n", round(porcentaje, 2), "%")

# Crear el pie chart con etiquetas y título
pie(conteo, labels = etiquetas, main = "Distribución de tipos de combustible, variable fueltype")
```



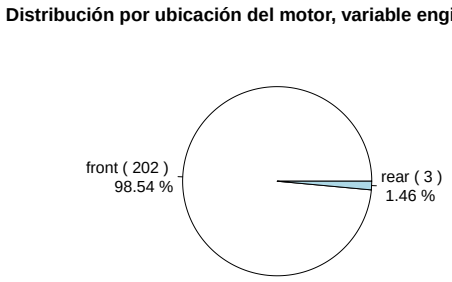
```r
# Contar la frecuencia de cada categoría
conteo <- table(categorical_df$carbody)

# Calcular el porcentaje
porcentaje <- prop.table(conteo) * 100

# Crear etiquetas con número de ocurrencias y porcentaje
etiquetas <- paste(names(conteo), "\n", "(", conteo, ")\n", round(porcentaje, 2), "%", sep = "")

# Crear el pie chart con etiquetas y título
pie(conteo, labels = etiquetas, main = "Distribución de tipos de carrocería, variable carbody", cex = 0

# Crear leyenda fuera del gráfico
legend("topright", legend = etiquetas, fill = rainbow(length(conteo)), cex = 0.7)
```




```r
conteo <- table(categorical_df$drivewheel)

# Calcular el porcentaje
porcentaje <- prop.table(conteo) * 100

etiquetas <- paste(names(conteo), "(", conteo, ")", "\n", round(porcentaje, 2), "%")

# Crear el pie chart con etiquetas y título
pie(conteo, labels = etiquetas, main = "Distribución por tipos de rueda, variable drivewheel")
```

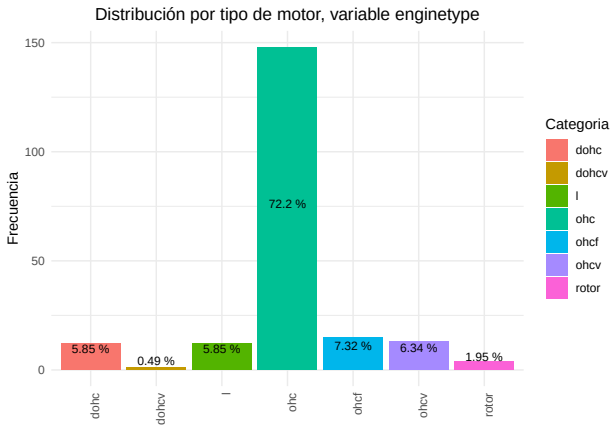


```r
conteo <- table(categorical_df$enginelocation)

# Calcular el porcentaje
porcentaje <- prop.table(conteo) * 100

etiquetas <- paste(names(conteo), "(", conteo, ")", "\n", round(porcentaje, 2), "%")

# Crear el pie chart con etiquetas y título
pie(conteo, labels = etiquetas, main = "Distribución por ubicación del motor, variable enginelocation")
```




```r
# Cargar la biblioteca ggplot2 si no está cargada
if (!require(ggplot2)) {
  install.packages("ggplot2")
  library(ggplot2)
}

data <- data.frame(Categoria = categorical_df$enginetype)

conteo <- table(data$Categoria)

porcentaje <- prop.table(conteo) * 100

data_etiquetas <- data.frame(Categoria = names(conteo), Porcentaje = porcentaje)

ggplot(data, aes(x = Categoria, fill = Categoria)) +
  geom_bar() +
  geom_text(data = data_etiquetas, aes(label = paste(round(porcentaje, 2), "%"), y = porcentaje), vjust
  labs(title = "Distribución por tipo de motor, variable enginetype", x = NULL, y = "Frecuencia") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

```
## Don't know how to automatically pick scale for object of type <table>.
## Defaulting to continuous.
```




### Selección de nuestras 6 variables para el análisis de precio

Para los sigiuentes estudios primeramente se escogerán 6 variables no categóricas, esto debido a que symboling no tiene una correlación alta con price, por otra parte considero que las otras variables categóricas no son especialmente valiosas para determinar el precio, (puede que algunas de estas si tengan un buen desempeño para ayudar a definir el precio, sin embargo considero que las variables continuas por si solas, son lo suficientemente significativas para definir un buen rango de precios de un automovil)

Para las variables continuas escogeremos aquellas que tengan una alta correlación con la variable 'price' y una correlación entre ellas cercanas a 0, esto para evitar colinealidad, a continuación se muestra el método de selección de variables.

Nos apoyaremos del análisis de componentes principales, así como de algunas gráficas que se obtienen a partir de este análisis para decidir las variables en las cuales se realizarán más estudios.

Referencias para el análisis hecho con PCA: https://www.datacamp.com/tutorial/pca-analysis-r

### Análisis de componentes principales para la selección de las variables numéricas

```
data_normalized <- scale(numeric_df[,!names(numeric_df) %in% c("symboling")])
```

```
library('corrr')
```

```
## Warning: package 'corrr' was built under R version 4.2.3
```

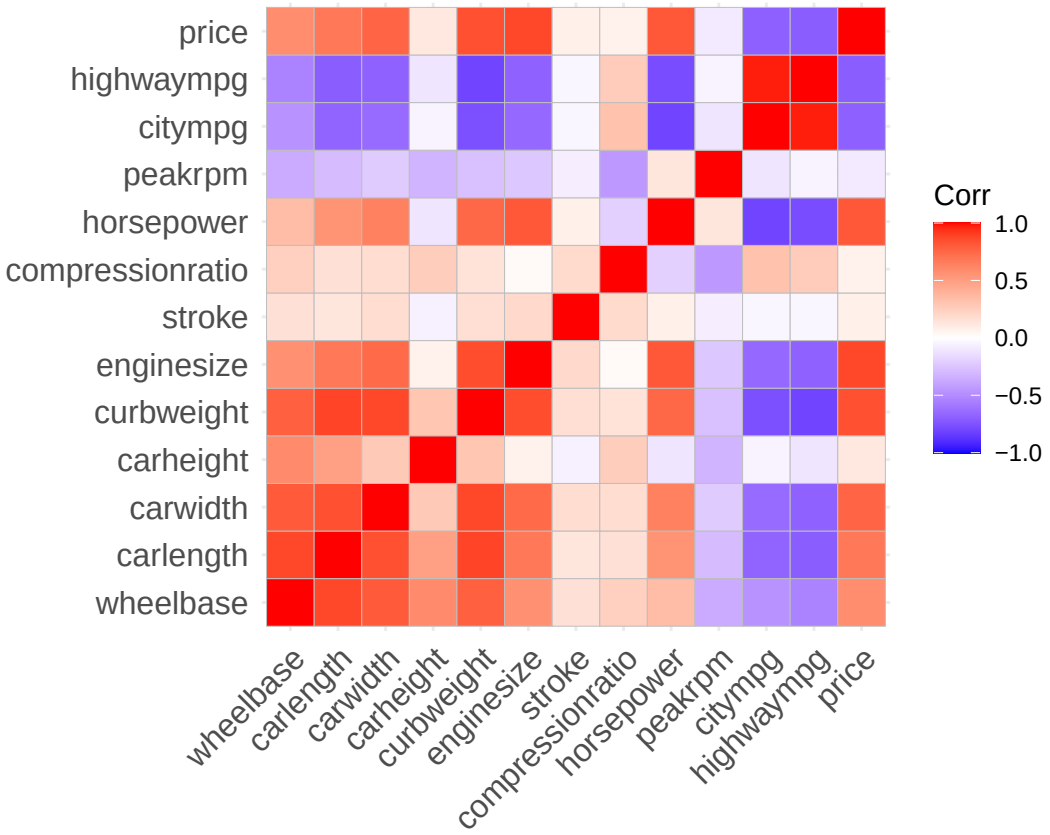
```
library("ggcorrplot")
```

```
## Warning: package 'ggcorrplot' was built under R version 4.2.3
```

```
library("FactoMineR")
```

```
## Warning: package 'FactoMineR' was built under R version 4.2.3
```

```
corr_matrix <- cor(data_normalized)
ggcorrplot(corr_matrix)
```




```
data.pca <- princomp(corr_matrix)
summary(data.pca)
```

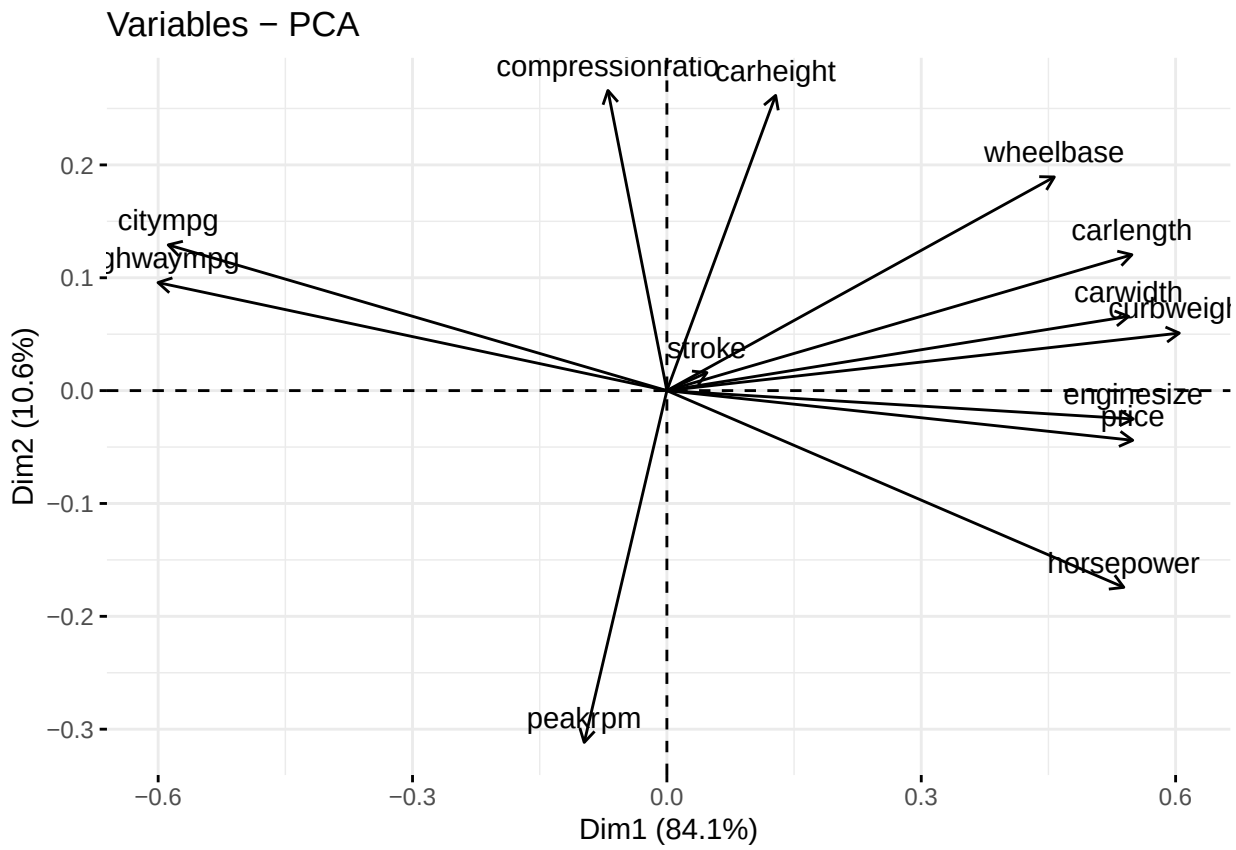
```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4      Comp.5
## Standard deviation     1.6749218 0.5938538 0.30829629 0.22591628 0.132923698
## Proportion of Variance 0.8413712 0.1057688 0.02850593 0.01530713 0.005299116
## Cumulative Proportion  0.8413712 0.9471401 0.97564601 0.99095314 0.996252256
##                            Comp.6       Comp.7       Comp.8      Comp.9
## Standard deviation     0.085209491 0.0504558527 0.0371451032 0.023887256
## Proportion of Variance 0.002177582 0.0007635223 0.0004138107 0.000171132
## Cumulative Proportion  0.998429838 0.9991933601 0.9996071708 0.999778303
##                           Comp.10      Comp.11      Comp.12 Comp.13
## Standard deviation     0.0206335951 1.685662e-02 5.413729e-03       0
## Proportion of Variance 0.0001276875 8.521963e-05 8.790057e-06       0
## Cumulative Proportion  0.9999059903 9.999912e-01 1.000000e+00       1
```

```
library(factoextra)
```

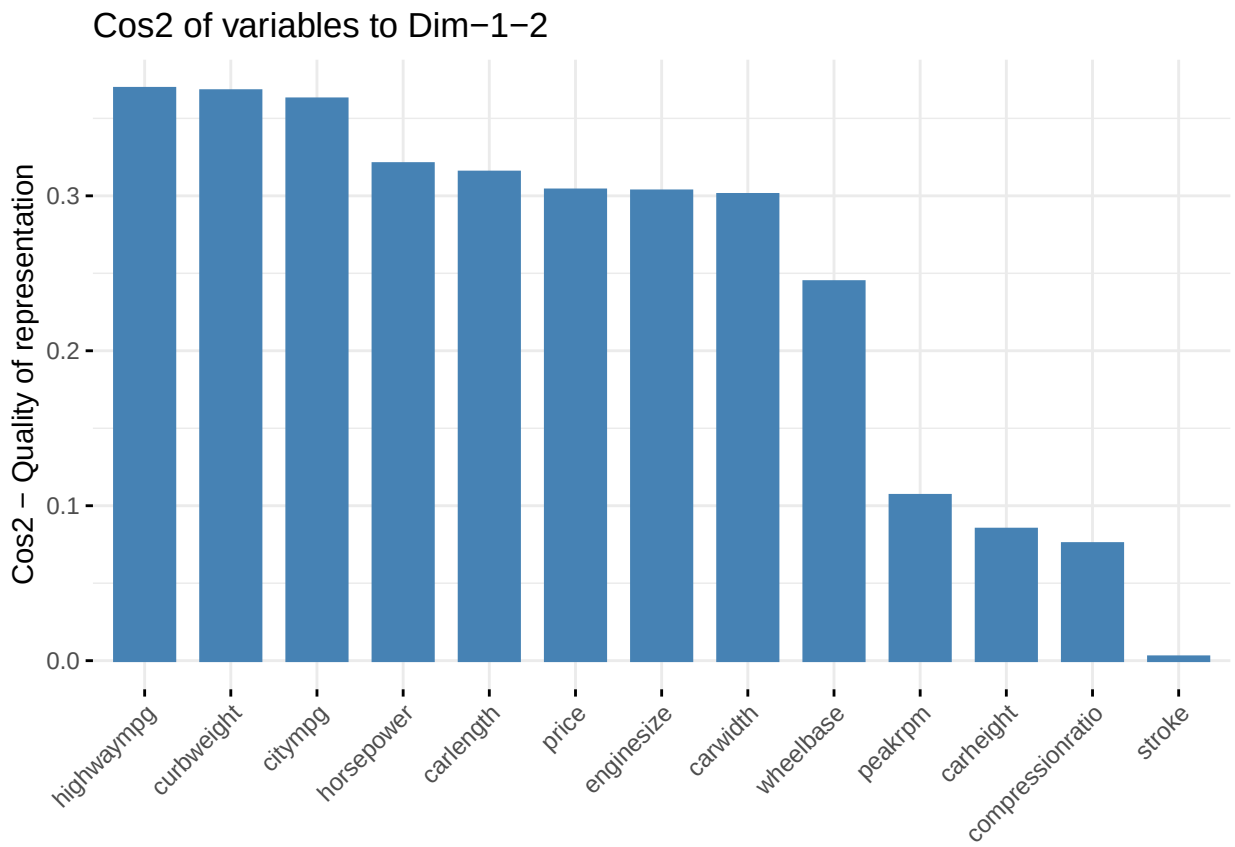
```
## Warning: package 'factoextra' was built under R version 4.2.3
```

```
#fviz_eig(data.pca, addlabels = TRUE)
```

```r
# Graph of the variables
fviz_pca_var(data.pca, col.var = "black")
```



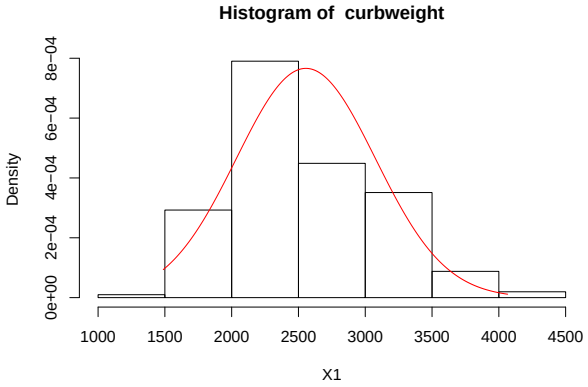

```r
fviz_cos2(data.pca, choice = "var", axes = 1:2)
```



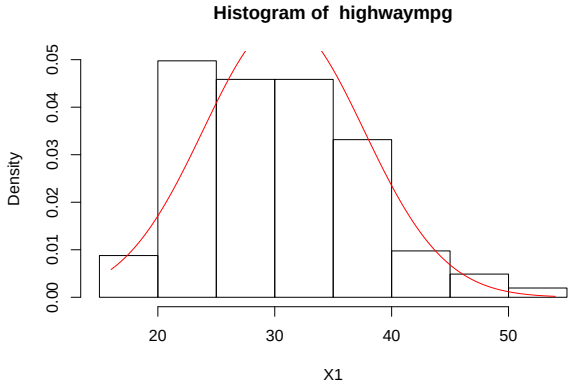

Se escogerán a las variables curbweight, highwaympg, horsepower,wheelbase, carheight, enginesize y la variable categórica symboling.

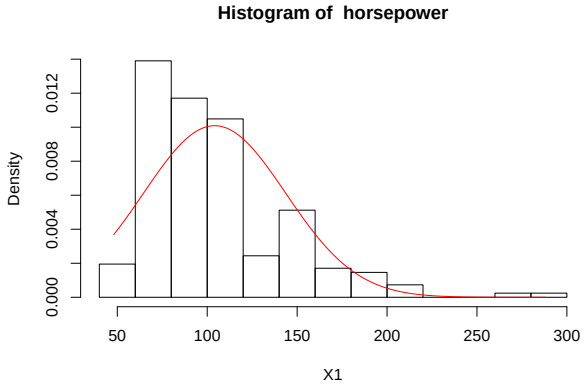
```
# Creamos dataset de variables categoricas
dummy <- c("symboling","curbweight","highwaympg","horsepower","wheelbase","carheight","enginesize","pri
main_df <- df[, dummy]
```

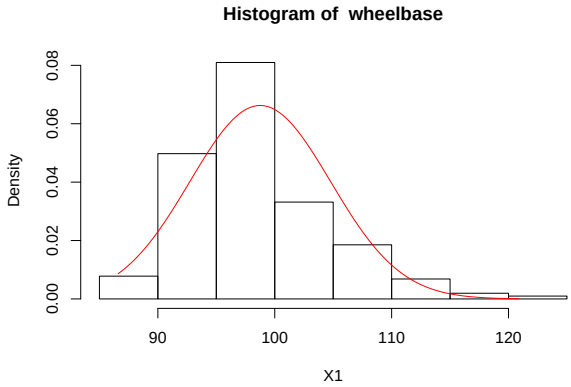
```
# Filter out non-numeric columns
numeric_cols <- sapply(main_df, is.numeric)
num_main_df <- main_df[, numeric_cols]
for (col_name in names(num_main_df)) {
  X1 = num_main_df[[col_name]]
  title_ = paste('Histogram of ',col_name)
  hist(X1,prob=TRUE,col=0,main=title_)
  x=seq(min(X1),max(X1),0.1)
  y=dnorm(x,mean(X1),sd(X1))
  lines(x,y,col="red")
}
```

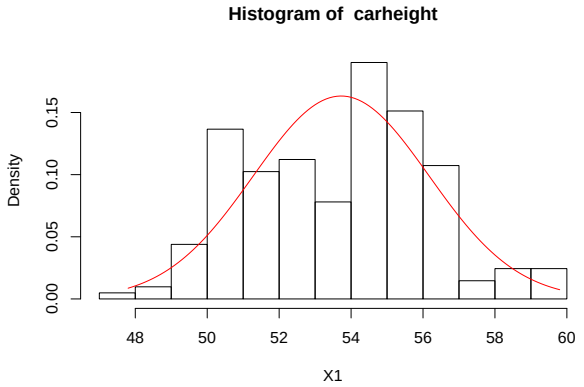


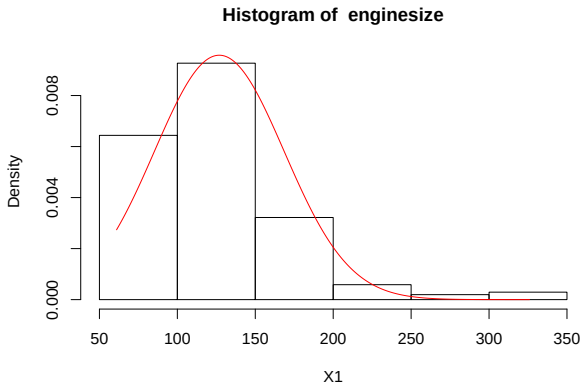

## Creación de dataset de variables seleccionadas

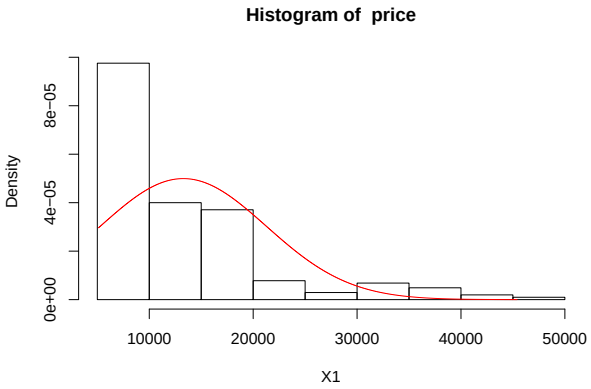





**Prueba de Anderson-Darling sobre normalidad de los datos originales** Si el p-value no está por debajo de nuestro nivel de significancia 0.05, no tenemos suficiente evidencia para rechazar la hipótesis nula.

hipotesis nula: los datos no siguen una distribución normal

hipotesis alternativa: los datos siguen una distribución normal

```
library(nortest)
for (col_name in names(num_main_df)){
  cat("\nPara la variable ", col_name)
  print(ad.test(num_main_df[[col_name]]))
  cat("\n")

}
```

```
##
## Para la variable  curbweight
##  Anderson-Darling normality test
##
## data:  num_main_df[[col_name]]
## A = 2.8001, p-value = 4.542e-07
##
##
##
## Para la variable  highwaympg
##  Anderson-Darling normality test
##
## data:  num_main_df[[col_name]]
## A = 1.4759, p-value = 0.0008068
##
##
##
## Para la variable  horsepower
##  Anderson-Darling normality test
##
## data:  num_main_df[[col_name]]
## A = 6.3485, p-value = 1.233e-15
##
##
```

```
##
## Para la variable  wheelbase
##  Anderson-Darling normality test
##
## data:  num_main_df[[col_name]]
## A = 6.9217, p-value < 2.2e-16
##
##
##
## Para la variable  carheight
##  Anderson-Darling normality test
##
## data:  num_main_df[[col_name]]
## A = 1.0331, p-value = 0.00999
##
##
##
## Para la variable  enginesize
##  Anderson-Darling normality test
##
## data:  num_main_df[[col_name]]
## A = 8.7408, p-value < 2.2e-16
##
##
##
## Para la variable  price
##  Anderson-Darling normality test
##
## data:  num_main_df[[col_name]]
## A = 12.341, p-value < 2.2e-16
```

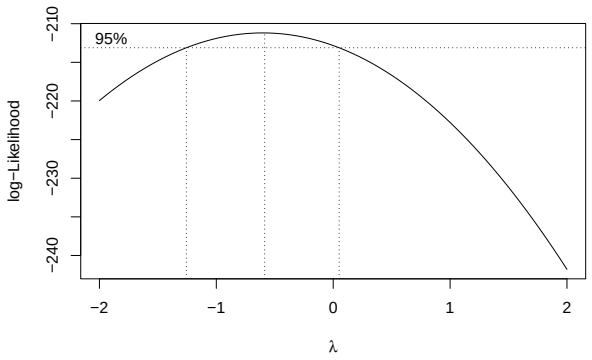
Podemos ver que para todas las variables escogidas, el p-value es menor a 0.05, por lo que podemos rechazar la hipótesis nula que dice que los datos no siguen una distribución normal, por lo que concluimos que todas las variables dentro de main_df siguen una distribución normal.

**Análisis de sesgo y curtosis de las variables** Ahora realizaremos un análisis de sesgo y curtosis para las variables seleccionadas, con y sin transformación de estos, utilizaremos la transformación de box-cox en sobre estas variables para que, en caso de tener sesgos o curtosis grandes en los datos, buscar ajustarlos según sea necesario, en caso de que el sesgo sea poco o que la curtosis sea cercana a una curtosis mesocúrtica, no se cambiaran los datos de la variable respectiva a sus transformados.

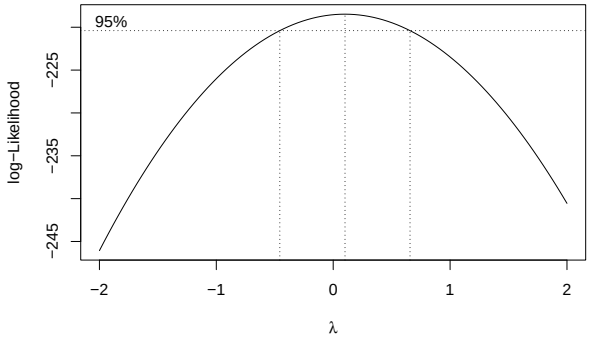
```r
library(moments)
library(MASS)

# Create an empty dataframe with the same number of rows as main_df
transf_df <- data.frame(matrix(NA, nrow = nrow(num_main_df), ncol = ncol(num_main_df)))

# Copy column names from main_df to transf_df
colnames(transf_df) <- colnames(num_main_df)

for (col_name in names(num_main_df)) {
  cat('Resultados del análisis, variable ', col_name, " sin transformación \n")
  cat('Sesgo: ', skewness(num_main_df[[col_name]]), "\n")
```

```
  cat('Curtosis: ', kurtosis(num_main_df[[col_name]]), "\n")
  cat('\n')

  # Compute the Box-Cox transformation and store it in transf_df
  bc_result <- boxcox((num_main_df[[col_name]] + 3) ~ 1)
  opt_lambda <- bc_result$x[which.max(bc_result$y)]
  dummy_exact <- ((num_main_df[[col_name]] + 1) ^ opt_lambda - 1) / opt_lambda

  # Add the transformed values to transf_df
  transf_df[[col_name]] <- dummy_exact

  cat('Resultados del análisis, variable ', col_name, " con transformación \n")
  cat('Sesgo: ', skewness(dummy_exact), "\n")
  cat('Curtosis: ', kurtosis(dummy_exact), "\n")
  cat('\n')
}
```

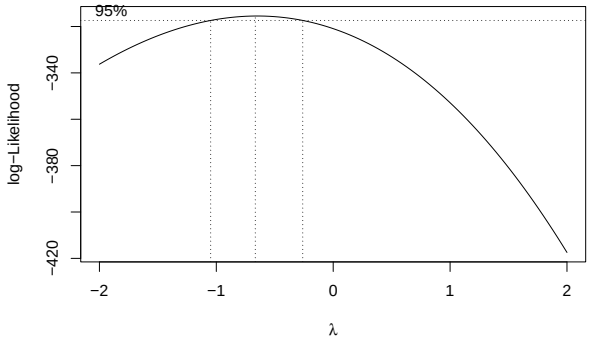
```
## Resultados del análisis, variable  curbweight  sin transformación
## Sesgo:  0.6764022
## Curtosis:  2.929058
```



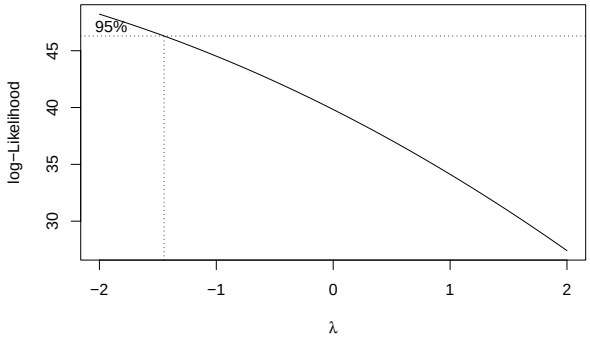

```
## Resultados del análisis, variable  curbweight  con transformación
## Sesgo:  0.03097682
## Curtosis:  2.408091
##
## Resultados del análisis, variable  highwaympg  sin transformación
## Sesgo:  0.5360379
## Curtosis:  3.400284
```



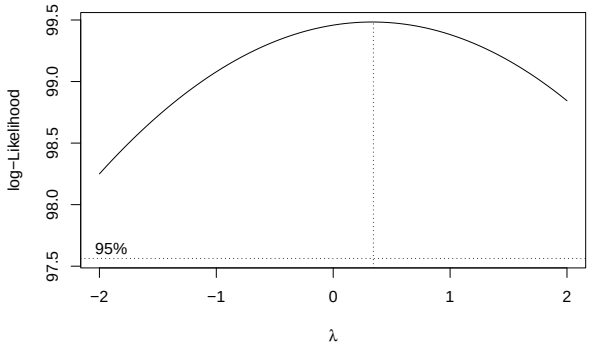

```
## Resultados del análisis, variable  highwaympg  con transformación
## Sesgo:  -0.03479979
## Curtosis:  2.956976
##
## Resultados del análisis, variable  horsepower  sin transformación
## Sesgo:  1.395006
## Curtosis:  5.589862
```



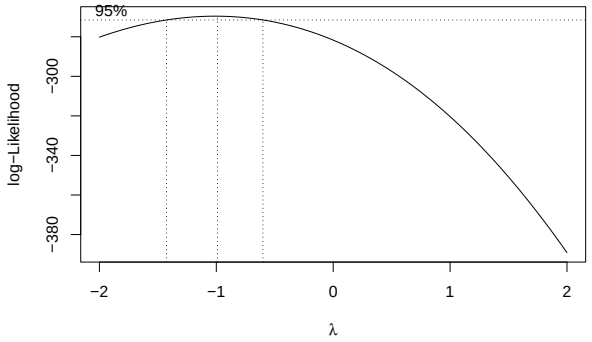

```
## Resultados del análisis, variable  horsepower  con transformación
## Sesgo:  0.01770273
## Curtosis:  2.324334
##
## Resultados del análisis, variable  wheelbase  sin transformación
## Sesgo:  1.042514
## Curtosis:  3.963276
```



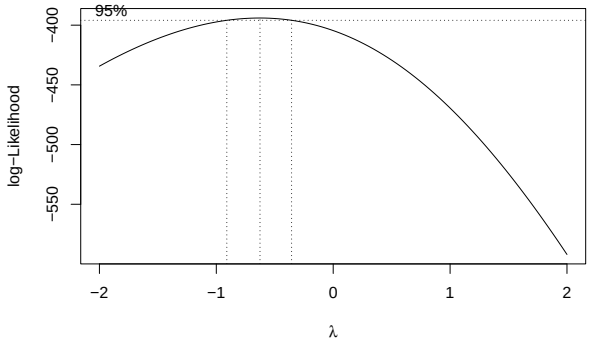

```
## Resultados del análisis, variable  wheelbase  con transformación
## Sesgo:  0.5472698
## Curtosis:  3.256407
##
## Resultados del análisis, variable  carheight  sin transformación
## Sesgo:  0.06265992
## Curtosis:  2.537812
```




```
## Resultados del análisis, variable  carheight  con transformación
## Sesgo:  -0.003715992
## Curtosis:  2.503206
##
## Resultados del análisis, variable  enginesize  sin transformación
## Sesgo:  1.933375
## Curtosis:  8.14803
```




```
## Resultados del análisis, variable  enginesize  con transformación
## Sesgo:  -0.01303968
## Curtosis:  3.277446
##
## Resultados del análisis, variable  price  sin transformación
## Sesgo:  1.764644
## Curtosis:  5.948598
```




```
## Resultados del análisis, variable  price  con transformación
## Sesgo:  0.09511397
## Curtosis:  2.168426
```

Podemos ver una disminución en el sesgo y en la curtosis al realizar la transformación box-cox en las variables, curbweight, highwaympg, horsepower, wheelbase, carheight y enginesize, crearemos un dataset nuevo que tenga estas variables transformadas,la variable symboling debido a que es categórica permanecera sin cambios.

```
transf_df$symboling <- main_df$symboling
```

Ahora procederemos a hacer nuestros estudios estadísticos para determinar la significancia de las distintas variables predictoras sobre la variable 'price'

**ANOVA para estimar el cambio de 'price' de acuerdo a la variable categórica 'symboling'**

Hipótesis nula: No hay diferencia entre las medias de los grupos de las variables categóricas. Hipótesis alterna: Las medias por grupo son diferentes entre ellas.

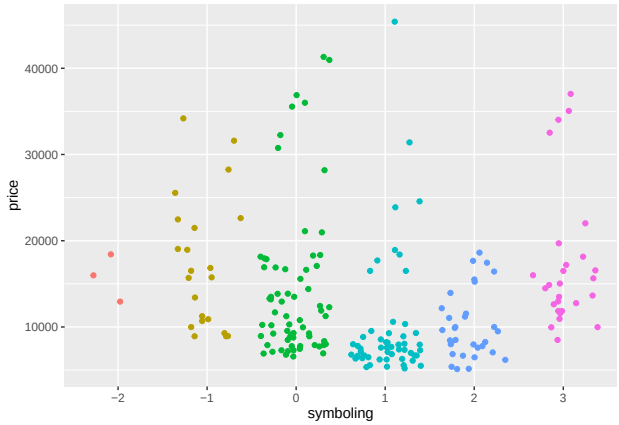
```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## v purrr   0.3.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x dplyr::select()    masks MASS::select()
## x dplyr::src()       masks Hmisc::src()
## x dplyr::summarize() masks Hmisc::summarize()
```

```
dat <- main_df %>%
  select(symboling, price)

summary(dat)
```

```
##  symboling     price
##  -2: 3     Min.   : 5118
##  -1:22     1st Qu.: 7788
##  0 :67     Median :10295
##  1 :54     Mean   :13277
##  2 :32     3rd Qu.:16503
##  3 :27     Max.   :45400
```

```
library(ggplot2)

ggplot(dat) +
  aes(x = symboling, y = price, color = symboling) +
  geom_jitter() +
  theme(legend.position = "none")
```



Aquí podemos observar como no hay una clara diferencia de varianza en los precios dependiendo del tipo de valor categórico en la variable symboling, se ve como para todo tipo de valor categórico, los precios tienden a estar concentrados en un mismo precio (alrededor de 10000).

```
res_aov <- aov(price ~ symboling,
  data = dat
)

summary(res_aov)
```

```
##              Df    Sum Sq   Mean Sq F value   Pr(>F)
## symboling     5 1.768e+09 353527182   6.252 2.07e-05 ***
## Residuals   199 1.125e+10  56542731
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dado que el p-value es menor a 0.05, rechazamos la hipótesis nula, la cual nos dice que todas las medias de los grupos de symboling son iguales. Con lo cual podemos concluir que al menos la media de un grupo es significativamente diferente a otras en términos del precio.

```
# Assuming your dataframe is named 'transf_df'
# Calculate the means by symboling group
means_by_symboling <- aggregate(main_df$price, by=list(Symboling=main_df$symboling), FUN=mean)

# Rename the columns for clarity
colnames(means_by_symboling) <- c("Symboling", "Mean_Price")

# Display the result
print(means_by_symboling)
```

```
##   Symboling Mean_Price
## 1        -2   15781.67
## 2        -1   17330.68
## 3         0   14366.97
## 4         1   10037.91
## 5         2   10109.28
## 6         3   17221.30
```

Para identificar que grupos específicos son aquellos que tienen una diferencia significativa realizaremos la prueba de Tukey.

```
TukeyHSD(res_aov, conf.level = .95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = price ~ symboling, data = dat)
##
## $symboling
##              diff        lwr        upr     p adj
## -1--2  1549.01515 -11768.745 14866.7751 0.9994375
## 0--2  -1414.70149 -14184.503 11355.0998 0.9995565
```

```
## 1--2  -5743.75926 -18579.267  7091.7489 0.7914833
## 2--2  -5672.38542 -18738.051  7393.2806 0.8118181
## 3--2   1439.62963 -11729.324 14608.5832 0.9995844
## 0--1  -2963.71664  -8280.873  2353.4394 0.5968994
## 1--1  -7292.77441 -12765.853 -1819.6953 0.0023044
## 2--1  -7221.40057 -13214.393 -1228.4086 0.0083362
## 3--1   -109.38552  -6324.340  6105.5691 1.0000000
## 1-0   -4329.05777  -8286.290  -371.8254 0.0229971
## 2-0   -4257.68393  -8907.528   392.1598 0.0937247
## 3-0    2854.33112  -2078.290  7786.9526 0.5564065
## 2-1      71.37384  -4755.995  4898.7429 1.0000000
## 3-1    7183.38889   2083.075 12283.7027 0.0010122
## 3-2    7112.01505   1457.410 12766.6202 0.0049706
```

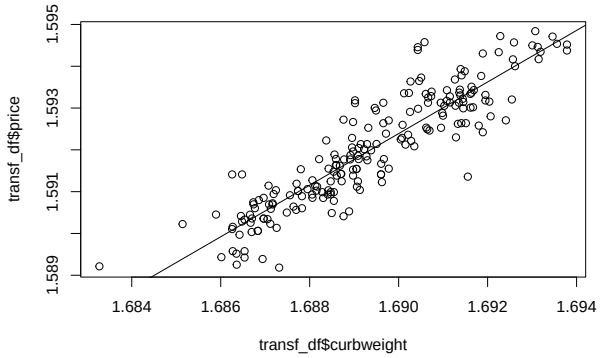
Concluimos con un 95% de confianza que hay una diferencia significativa en la media del precio de automoviles entre los grupos (-1,1), (-1,2), (0,1), (1,3) y (2,3) de la variable symboling.

**Análisis de regresión sobre variables numéricas transformadas (transf_df)**

```
linear_model <- lm(price ~ curbweight, data=transf_df)
summary(linear_model)
```

```
##
## Call:
## lm(formula = price ~ curbweight, data = transf_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.990e-03 -3.552e-04 -9.450e-06  3.146e-04  1.827e-03
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.54944    0.03454   15.91   <2e-16 ***
## curbweight   0.61713    0.02044   30.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0005847 on 203 degrees of freedom
## Multiple R-squared:  0.8178, Adjusted R-squared:  0.8169
## F-statistic: 911.2 on 1 and 203 DF,  p-value: < 2.2e-16
```

```
#plot(linear_model)
# Plot abline plot
plot(transf_df$curbweight, transf_df$price)
abline(linear_model)
```




En el análisis de residuos podemos observar que la mediana es muy cercana a 0 lo cual nos dice que hay muy poco sesgo, podemos observar también que los residuos están simétricamente distribuidos lo cual también indica que no hay sesgo. Vemos que 81% de la varianza de 'price' puede ser explicada en esta regresión, el cual es un porcentaje alto.

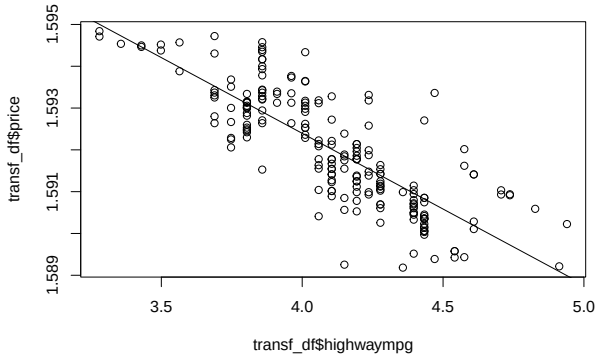
El estadístico F es de 911.2 lo cual es bueno, entre mayor sea este valor nos indica que el modelo es estadísticamente significante, también podemos observar el p-value que apoya la teoría de que este modelo es estadísticamente significante.

Con este modelo concluimos que la variable curbweight es estadísticamente significante para determinar el precio de automoviles de nuestra base de datos.

```
linear_model <- lm(price ~ highwaympg, data=transf_df)
summary(linear_model)
```

```
##
## Call:
## lm(formula = price ~ highwaympg, data = transf_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.599e-03 -5.130e-04 -8.736e-05  4.425e-04  2.668e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.6068867  0.0007515 2138.16   <2e-16 ***
## highwaympg  -0.0036228  0.0001824  -19.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0007985 on 203 degrees of freedom
## Multiple R-squared:  0.6603, Adjusted R-squared:  0.6586
## F-statistic: 394.5 on 1 and 203 DF,  p-value: < 2.2e-16
```

```
# plot(linear_model)
# Plot abline plot
plot(transf_df$highwaympg, transf_df$price)
abline(linear_model)
```

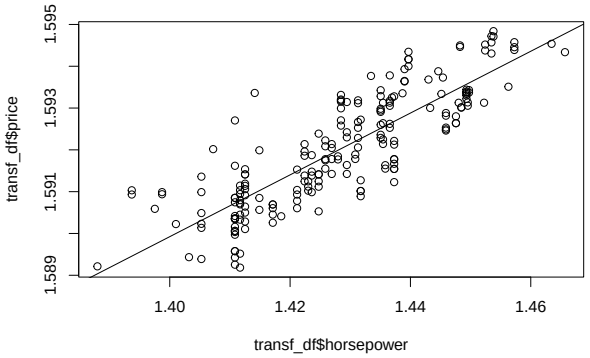



También podemos ver buenos resultados para el modelo lineal hecho con la variable highwaympg, la media de los residuos es cercana a 0 y estos se distribuyen de forma simétrica, el valor de F estadístico es alto y el p-value del modelo es bajo, lo cual indica que el modelo es estadísticamente significante, podemos interpretar en el contexto de los datos y con la gráfica de la regresión lineal, que a eficiencia de millas por galón, los precios suelen ser más bajos, esto tiene sentido considerando que aquellos autos con motores más grandes suelen tener más piezas, equipo, son más pesados, etc. Mientras que los motores pequeños que suelen ser los quue consumen menos gasolina, son más baratos.

```
linear_model <- lm(price ~ horsepower, data=transf_df)
summary(linear_model)
```

```
##
## Call:
## lm(formula = price ~ horsepower, data = transf_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0016043 -0.0004988 -0.0001194  0.0005531  0.0023938
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.486428   0.004737  313.79   <2e-16 ***
## horsepower  0.073925   0.003317   22.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0007379 on 203 degrees of freedom
## Multiple R-squared:  0.7099, Adjusted R-squared:  0.7085
## F-statistic: 496.8 on 1 and 203 DF,  p-value: < 2.2e-16
```

```
# plot(linear_model)
# Plot abline plot
plot(transf_df$horsepower, transf_df$price)
abline(linear_model)
```

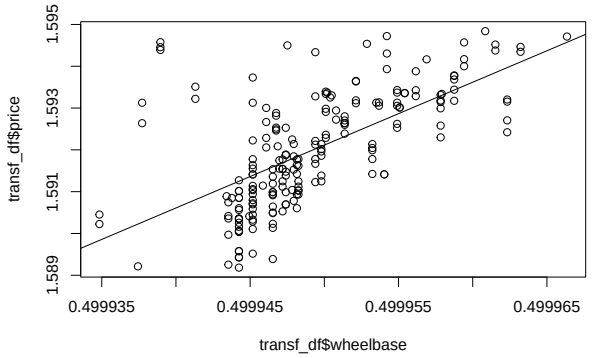



El modelo cumple supuestos que lo hacene stadísticamente significativo, buenos valores para el análisis de residuos, un F estadístico alto y un p-value bajo, el 70% de la varianza de la variable price puede ser explicada con la regresión, vemos como a mayor cantidad de caballos de fuerza, mayor el precio, esto tiene sentido debido a que los motores que tienen más caballos de fuerza son más grandes, tienen más componentes, el auto tiene que tener componentes que se adapten al motor, etc.

```
linear_model <- lm(price ~ wheelbase, data=transf_df)
summary(linear_model)
```

```
##
## Call:
## lm(formula = price ~ wheelbase, data = transf_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0022029 -0.0007359 -0.0001118  0.0004820  0.0041236
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -73.607      6.575  -11.20   <2e-16 ***
## wheelbase    150.414     13.151   11.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001068 on 203 degrees of freedom
## Multiple R-squared:  0.3919, Adjusted R-squared:  0.3889
## F-statistic: 130.8 on 1 and 203 DF,  p-value: < 2.2e-16
```

```
# plot(linear_model)
# Plot abline plot
plot(transf_df$wheelbase, transf_df$price)
abline(linear_model)
```

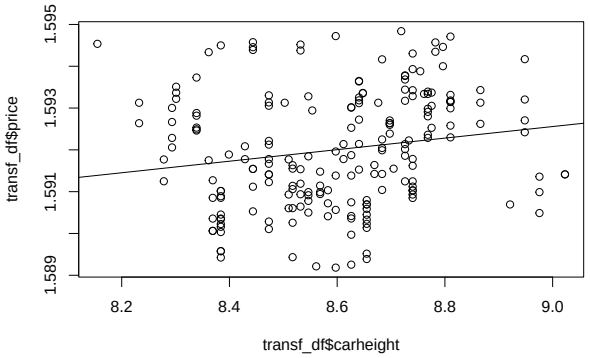



En esta variable vemos como aún y que se tengan buenas estimaciones de normalidad en el análisis de residuos y que el valor F estadístico sea relativamente alto, el modelo estima únicamente el 39% de la varianza de la variable price, aun y que este valor sea bajo, nos puede ayudar en una regresión lineal multiple al ser combinada con otras variables con las que no se tengan colinealidad.

```
linear_model <- lm(price ~ carheight, data=transf_df)
summary(linear_model)
```

```
##
## Call:
## lm(formula = price ~ carheight, data = transf_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0028183 -0.0011512 -0.0000649  0.0010575  0.0031466
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.5801222  0.0045964 343.773   <2e-16 ***
## carheight   0.0013817  0.0005346   2.585   0.0104 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001348 on 203 degrees of freedom
## Multiple R-squared:  0.03186,    Adjusted R-squared:  0.02709
## F-statistic: 6.681 on 1 and 203 DF,  p-value: 0.01045
```

```
# plot(linear_model)
# Plot abline plot
plot(transf_df$carheight, transf_df$price)
abline(linear_model)
```

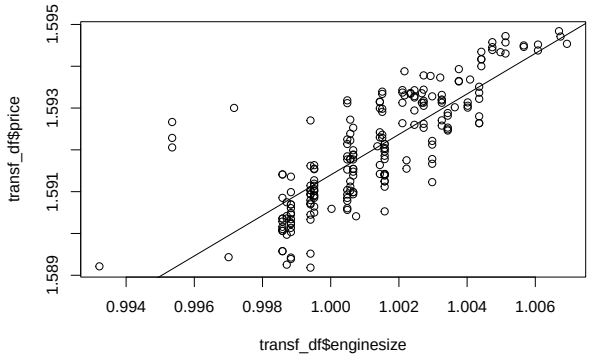



En esta variable podemos observar como el F estadpistico y el p-value son altos comparandose con las previos modelos hechos con las variables analizadas, el valor ajustado de R cuadrado es muy pequeño, solo del 2%, aún y que el modelo explique muy poca de la varianza de price no se puede descartar que sea estadísticamente significativo debido a que el p-value es menor a 0.05

```
linear_model <- lm(price ~ enginesize, data=transf_df)
summary(linear_model)
```

```
##
## Call:
## lm(formula = price ~ enginesize, data = transf_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0019301 -0.0005727 -0.0001043  0.0004603  0.0035193
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.10683    0.02506   44.17   <2e-16 ***
## enginesize   0.48457    0.02503   19.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.000812 on 203 degrees of freedom
## Multiple R-squared:  0.6487, Adjusted R-squared:  0.647
## F-statistic: 374.8 on 1 and 203 DF,  p-value: < 2.2e-16
```

```
# plot(linear_model)
# Plot abline plot
plot(transf_df$enginesize, transf_df$price)
abline(linear_model)
```



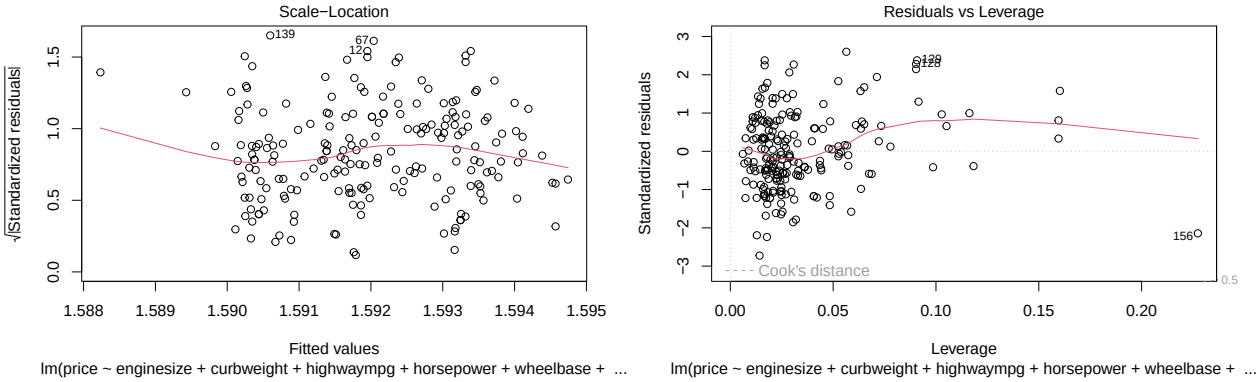

La interpretación y conclusiones para la variable enginesize, son muy parecidas a las definidas para la variable horsepower, ambos presentan un r-squared similar y en la gráfica de componentes principales se podía observar como existía una colinealidad entre ambas variables, aunque por separadas ambas sean de significancia para determinar el precio de un automovil, el juntarlas en un modelo multi lineal no asegura que el modelo vaya a explicar mucha nueva varianza de la que se tenía por separado, se podría considerar descartar una de estas variables de un modelo multi-lineal para evitar overfitting.

### Análisis de modelo multi-lineal con variables numéricas transformadas (transf_df)

```
linear_model <- lm(price ~ enginesize + curbweight + highwaympg + horsepower + wheelbase + carheight, da
summary(linear_model)
```

```
##
## Call:
## lm(formula = price ~ enginesize + curbweight + highwaympg + horsepower +
##     wheelbase + carheight, data = transf_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.416e-03 -3.171e-04 -5.127e-05  3.577e-04  1.322e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.284e-01  6.133e+00   0.070    0.944
## enginesize  -2.529e-02  3.546e-02  -0.713    0.477
## curbweight   4.707e-01  5.776e-02   8.150 4.07e-14 ***
## highwaympg   3.997e-04  3.020e-04   1.324    0.187
## horsepower   3.513e-02  6.295e-03   5.580 7.82e-08 ***
## wheelbase    6.826e-01  1.235e+01   0.055    0.956
## carheight    6.691e-05  2.904e-04   0.230    0.818
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0005232 on 198 degrees of freedom
## Multiple R-squared:  0.8577, Adjusted R-squared:  0.8534
## F-statistic:   199 on 6 and 198 DF,  p-value: < 2.2e-16
```

```r
plot(linear_model)
```


```r
#abline(linear_model)
```

```r
step(linear_model,direction = "both",trace=1)
```

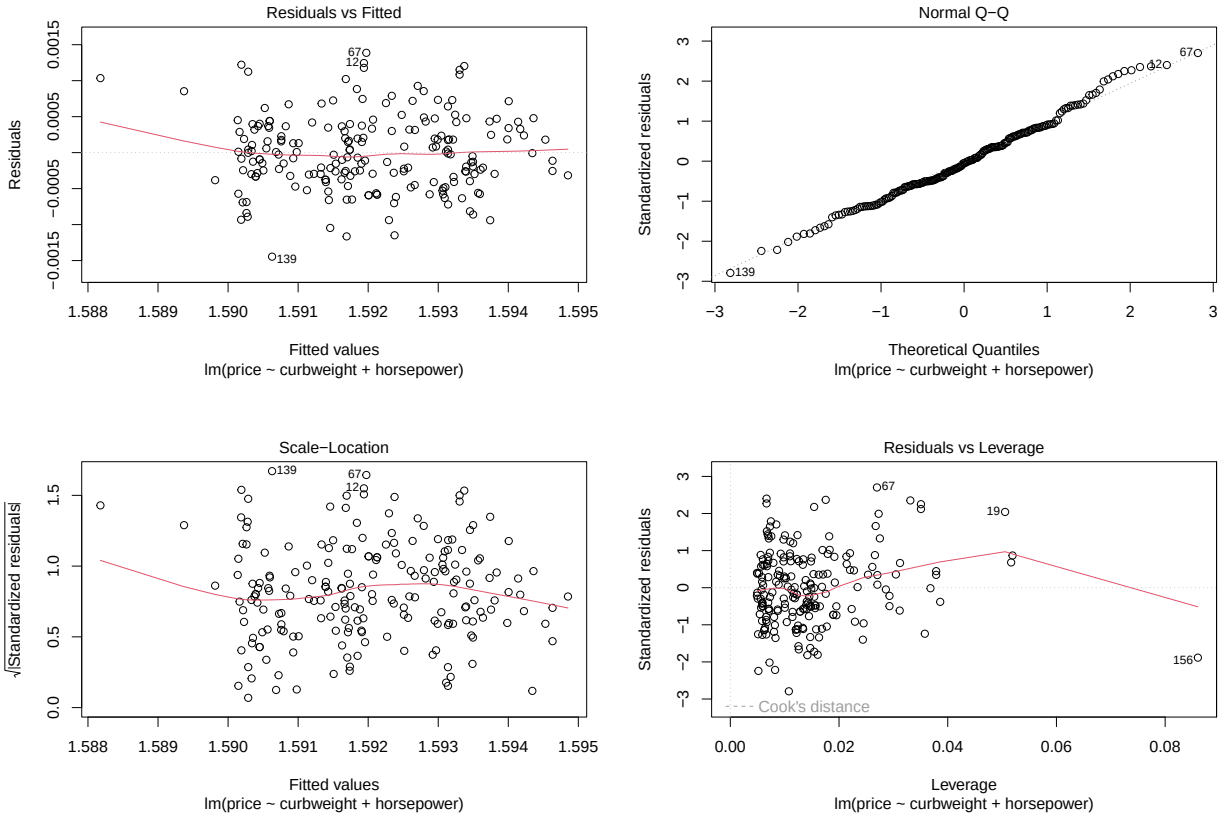
```
## Start:  AIC=-3090.91
## price ~ enginesize + curbweight + highwaympg + horsepower + wheelbase +
##     carheight
##
##               Df  Sum of Sq        RSS     AIC
## - wheelbase    1 8.0000e-10 5.4199e-05 -3092.9
## - carheight    1 1.4500e-08 5.4212e-05 -3092.8
## - enginesize   1 1.3920e-07 5.4337e-05 -3092.4
## - highwaympg   1 4.7960e-07 5.4677e-05 -3091.1
## <none>                      5.4198e-05 -3090.9
## - horsepower   1 8.5239e-06 6.2722e-05 -3063.0
## - curbweight   1 1.8180e-05 7.2378e-05 -3033.6
##
## Step:  AIC=-3092.9
## price ~ enginesize + curbweight + highwaympg + horsepower + carheight
##
##               Df  Sum of Sq        RSS     AIC
## - carheight    1 2.3800e-08 5.4222e-05 -3094.8
```

```
## - enginesize  1 1.3930e-07 5.4338e-05 -3094.4
## - highwaympg  1 4.7930e-07 5.4678e-05 -3093.1
## <none>                     5.4199e-05 -3092.9
## + wheelbase   1 8.0000e-10 5.4198e-05 -3090.9
## - horsepower  1 8.7291e-06 6.2928e-05 -3064.3
## - curbweight  1 2.3043e-05 7.7242e-05 -3022.3
##
## Step:  AIC=-3094.81
## price ~ enginesize + curbweight + highwaympg + horsepower
##
##              Df  Sum of Sq        RSS     AIC
## - enginesize  1 1.4040e-07 5.4363e-05 -3096.3
## - highwaympg  1 4.7260e-07 5.4695e-05 -3095.0
## <none>                     5.4222e-05 -3094.8
## + carheight   1 2.3800e-08 5.4199e-05 -3092.9
## + wheelbase   1 1.0100e-08 5.4212e-05 -3092.8
## - horsepower  1 9.9302e-06 6.4153e-05 -3062.3
## - curbweight  1 2.8382e-05 8.2605e-05 -3010.5
##
## Step:  AIC=-3096.28
## price ~ curbweight + highwaympg + horsepower
##
##              Df  Sum of Sq        RSS     AIC
## - highwaympg  1 3.5300e-07 5.4716e-05 -3096.9
## <none>                     5.4363e-05 -3096.3
## + enginesize  1 1.4000e-07 5.4222e-05 -3094.8
## + carheight   1 2.5000e-08 5.4338e-05 -3094.4
## + wheelbase   1 1.1000e-08 5.4352e-05 -3094.3
## - horsepower  1 1.1228e-05 6.5591e-05 -3059.8
## - curbweight  1 4.7276e-05 1.0164e-04 -2970.0
##
## Step:  AIC=-3096.95
## price ~ curbweight + horsepower
##
##              Df  Sum of Sq        RSS     AIC
## <none>                     5.4716e-05 -3096.9
## + highwaympg  1 3.5300e-07 5.4363e-05 -3096.3
## + enginesize  1 2.1000e-08 5.4695e-05 -3095.0
## + carheight   1 1.8000e-08 5.4698e-05 -3095.0
## + wheelbase   1 2.0000e-09 5.4714e-05 -3095.0
## - horsepower  1 1.4696e-05 6.9412e-05 -3050.2
## - curbweight  1 5.5804e-05 1.1052e-04 -2954.8


##
## Call:
## lm(formula = price ~ curbweight + horsepower, data = transf_df)
##
## Coefficients:
## (Intercept)   curbweight   horsepower
##     0.81220      0.43721      0.02885
```

```
alfredo <- lm(formula = price ~ curbweight + horsepower, data = transf_df)
summary(alfredo)
```

```
##
## Call:
## lm(formula = price ~ curbweight + horsepower, data = transf_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0014454 -0.0003236 -0.0000271  0.0003452  0.0013873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.812204   0.047092  17.247  < 2e-16 ***
## curbweight  0.437205   0.030460  14.353  < 2e-16 ***
## horsepower  0.028846   0.003916   7.366  4.4e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0005205 on 202 degrees of freedom
## Multiple R-squared:  0.8564, Adjusted R-squared:  0.855
## F-statistic: 602.2 on 2 and 202 DF,  p-value: < 2.2e-16
```

```
plot(alfredo)
```




```
aldo <- influence.measures(alfredo)
summary(aldo)
```

```
## Potentially influential observations of
```

```
##   lm(formula = price ~ curbweight + horsepower, data = transf_df) :
##
##     dfb.1_ dfb.crbw dfb.hrsp dffit   cov.r   cook.d hat
## 11   0.09  -0.09     0.09     0.19    0.95_*  0.01   0.01
## 12   0.09  -0.10     0.10     0.20    0.94_*  0.01   0.01
## 19   0.27  -0.24    -0.03     0.48_*  1.00    0.07   0.05_*
## 45   0.22  -0.21     0.07     0.32    0.95_*  0.03   0.02
## 49   0.03  -0.02     0.01    -0.04    1.04_*  0.00   0.03
## 67  -0.36   0.38    -0.41     0.46_*  0.93_*  0.07   0.03
## 69  -0.06   0.05    -0.03     0.06    1.04_*  0.00   0.03
## 86  -0.02   0.02     0.01    -0.16    0.95_*  0.01   0.01
## 114  0.01  -0.01     0.01    -0.01    1.04_*  0.00   0.03
## 115  0.07  -0.07     0.06    -0.08    1.05_*  0.00   0.04
## 127  0.23  -0.25     0.37     0.41_*  0.98    0.05   0.04
## 128  0.24  -0.26     0.39     0.43_*  0.97    0.06   0.04
## 129  0.23  -0.25     0.39     0.44_*  0.97    0.06   0.03
## 130  0.00   0.00     0.00     0.00    1.05_*  0.00   0.04
## 139 -0.08   0.06     0.08    -0.30    0.91_*  0.03   0.01
## 156  0.51  -0.53     0.54    -0.58_*  1.05_*  0.11   0.09_*
## 159 -0.05   0.06    -0.08     0.09    1.05_*  0.00   0.04
## 160 -0.04   0.04    -0.06     0.07    1.05_*  0.00   0.04
## 168 -0.08   0.09    -0.13    -0.20    0.95_*  0.01   0.01
## 183 -0.10   0.11    -0.15     0.16    1.06_*  0.01   0.05_*
## 185 -0.13   0.14    -0.19     0.20    1.06_*  0.01   0.05_*
```

Podemos observar como la varianza explicada por el modelo es del 85%, se realizaron modelos lineales de una solo variable que tenian valores de r cuadrada ajustada similares a este modelo (como curbweight con cerca de 0.81), la poca varianza extra obtenida a pesar de tener distintas variables con un r-squared alto se puede deber a la colinealidad, los valores aún y que sean distintos de las variables tienden a comportarse de la misma forma cuando cambia la variable price, por lo que esto confirma como el tratar de hacer un modelo mejor solo arrojando más variables a este no es necesariamente cierto, tiene incluso sus desventajas, el modelo puede ser más pesado de procesar por la cantidad de datos y puede darse overfitting que parece no ser nuestro caso debido a que el multiple r-squared y el adjusted r-squared son muy similares.

Este modelo es bueno, ya que explica el 85% de la varianza de la variable price, sin embargo pueden existir ciertos tratamientos de datos, como otro tipo de transformaciones, amputación de datos atípicos, combinación de variables que entran en el modelo, etc. que pueden crear mejores modelos, sin embargo vemos como cada una de las variables escogidas por si solas si son de significancia al momento de definir el valor de precio de un automovil de esta base de datos.