Dataset Clasificación: Vehicle

Ricardo Ignacio Shepstone Aramburu

Índice

Análisis exploratorio de datos

Definición del problema

El dataset consta un conjunto de parámetros que describen la silueta bidimensional de un objeto tridimensional. Los datos se obtuvieron de las siluetas producidas por el HIPS (Sistema de Procesamiento de Imágen Jerárquico) enun experimento donde se usaron cuatro vehículos diferentes: un autobús, furgoneta Chevrolet, un Saab 9000 y un Opel Manta.

El objetivo por lo tanto de este problema de clasificación será determinar a partir de los parámetros de las siluetas de estos vehículos determinar de qué vehículo se trata.

En cuanto a la dependencia de las variables, debido a que se tratan de diferentes medidas realizadas sobre la silueta de los vehículos, asumiremos que todas son dependientes, aunque se estudiará si esto es cierto o no.

En base a esta información se procede a plantear ciertas cuestiones e **hipótesis**:

- Hay diferencias en los parámetros de silueta dependiendo del vehículo.
- Al tener dos coches entre las clases la dificultad del problema estará en clasificar estas dos clases.¿Cómo se ve reflejada esa similitud?
- La clase autobús es la más sencilla de clasificar puiesto que la silueta difiere mucho de las del resto.

Preparación de los datos

Descripción de los datos

Con el dataset de vehicle se construye un data frame con 846 observaciones y 19 variables. No se dispone de mucha información del dataset por lo que no se podrá proporcionar una descripción detallada sobre las

variables. Pero en la siguiente lista se ha recopilado la poca información sobre las variables que hay disponible en el repositorio de UCI (https://archive.ics.uci.edu/ml/datasets/Statlog+%28Vehicle+Silhouettes%29)

Nombre	Tipo de dato	Información
Compactness	Cuantitativo discreto	(Perimetro medio)^2/area
Circularity	Cuantitativo discreto	(Radio medio)^2/area
Distance circularity	Cuantitativo discreto	area/(distancia media desde frontera)^2
Radius ratio	Cuantitativo discreto	(radio máx.-radio mín.)/radio medio
Praxis aspect ratio	Cuantitativo discreto	eje menor/eje mayor
Max length aspect ratio	Cuantitativo discreto	longitud perp. de máx. longitud/máx longitud
Scatter ratio	Cuantitativo discreto	inercia del eje menor/inercia del eje mayor
Elongatedness	Cuantitativo discreto	area/(ancho encogido)^2
Praxis rectangular	Cuantitativo discreto	area/(longitud eje*anchura eje)
Length rectangular	Cuantitativo discreto	area/(longitud máx.*longitud perp.)
Major variance	Cuantitativo discreto	momento de 2º orden en eje mayor/area
Minor variance	Cuantitativo discreto	momento de 2º orden en eje menor/area
Gyration radius	Cuantitativo discreto	(mavar+mivar)/area
Major skewness	Cuantitativo discreto	(momento de 3º orden en eje mayor)/sigma_min^3
Minor skewness	Cuantitativo discreto	(momento de 3º orden en eje menor)/sigma_maj^3
Minor kurtosis	Cuantitativo discreto	(momento de 4º orden en eje mayor)/sigma_min^4
Major kurtosis	Cuantitativo discreto	(momento de 4º orden en eje menor)/sigma_maj^4
Hollows ratio	Cuantitativo discreto	(area de huecos)/(area del polígono unido)
Class	Categórico nominal	Clase del vehículo

La variable de salida es la clase de vehículo a predecir y las de entrada serán los parámetros obtenidos de las silueras.

Importación de paquetes y del dataset.

Para empezar hay que incluir los paquetes que utilizaremos y cargar el dataset con el que se va a trabajar, además de cambiar los nombres de las variables.

```
require(tidyverse)
require(readr)
require(moments)
require(car)
require(corrplot)
require(fastDummies)

# Cargamos dataset

vehicle.raw <- read.csv("Input/vehicle/vehicle.dat", comment.char="@", header=FALSE)
vehicle <- vehicle.raw

# renombramos las variables para trabajar con ellas
colnames(vehicle) <- c('Compactness','Circularity','Distance_circularity','Radius_ratio',
                       'Praxis_aspect_ratio', 'Max_length_aspect_ratio', 'Scatter_ratio',
                       'Elongatedness', 'Praxis_rectangular', 'Length_rectangular',
                       'Major_variance', 'Minor_variance', 'Gyration_radius',
```

```
                        'Major_skewness', 'Minor_skewness', 'Minor_kurtosis',
                        'Major_kurtosis', 'Hollows_ratio', 'Class')
```

Hacemos las comprobaciones y obtenemos los datos de interés, como el número de filas y columnas.

```
#comprobamos número de columnas, filas y si se han cargado bien los datos, la estructura de estos
nrow(vehicle)
```

```
## [1] 846
```

```
ncol(vehicle)
```

```
## [1] 19
```

```
str(vehicle)
```

```
## 'data.frame':    846 obs. of  19 variables:
##  $ Compactness            : int  95 91 104 93 85 107 97 90 86 93 ...
##  $ Circularity            : int  48 41 50 41 44 57 43 43 34 44 ...
##  $ Distance_circularity   : int  83 84 106 82 70 106 73 66 62 98 ...
##  $ Radius_ratio           : int  178 141 209 159 205 172 173 157 140 197 ...
##  $ Praxis_aspect_ratio    : int  72 57 66 63 103 50 65 65 61 62 ...
##  $ Max_length_aspect_ratio: int  10 9 10 9 52 6 6 9 7 11 ...
##  $ Scatter_ratio          : int  162 149 207 144 149 255 153 137 122 183 ...
##  $ Elongatedness          : int  42 45 32 46 45 26 42 48 54 36 ...
##  $ Praxis_rectangular     : int  20 19 23 19 19 28 19 18 17 22 ...
##  $ Length_rectangular     : int  159 143 158 143 144 169 143 146 127 146 ...
##  $ Major_variance         : int  176 170 223 160 241 280 176 162 141 202 ...
##  $ Minor_variance         : int  379 330 635 309 325 957 361 281 223 505 ...
##  $ Gyration_radius        : int  184 158 220 127 188 264 172 164 112 152 ...
##  $ Major_skewness         : int  70 72 73 63 127 85 66 67 64 64 ...
##  $ Minor_skewness         : int  6 9 14 6 9 5 13 3 2 4 ...
##  $ Minor_kurtosis         : int  16 14 9 10 11 9 1 3 14 14 ...
##  $ Major_kurtosis         : int  187 189 188 199 180 181 200 193 200 195 ...
##  $ Hollows_ratio          : int  197 199 196 207 183 183 204 202 208 204 ...
##  $ Class                  : chr  " van " " van " " saab" " van " ...
```

```
summary(vehicle)
```

```
##   Compactness      Circularity    Distance_circularity  Radius_ratio
##  Min.   : 73.00   Min.   :33.00   Min.   : 40.00       Min.   :104.0
##  1st Qu.: 87.00   1st Qu.:40.00   1st Qu.: 70.00       1st Qu.:141.0
##  Median : 93.00   Median :44.00   Median : 80.00       Median :167.0
##  Mean   : 93.68   Mean   :44.86   Mean   : 82.09       Mean   :168.9
##  3rd Qu.:100.00   3rd Qu.:49.00   3rd Qu.: 98.00       3rd Qu.:195.0
##  Max.   :119.00   Max.   :59.00   Max.   :112.00       Max.   :333.0
##  Praxis_aspect_ratio Max_length_aspect_ratio Scatter_ratio   Elongatedness
##  Min.   : 47.00      Min.   : 2.000          Min.   :112.0   Min.   :26.00
##  1st Qu.: 57.00      1st Qu.: 7.000          1st Qu.:146.2   1st Qu.:33.00
##  Median : 61.00      Median : 8.000          Median :157.0   Median :43.00
##  Mean   : 61.69      Mean   : 8.567          Mean   :168.8   Mean   :40.93
##  3rd Qu.: 65.00      3rd Qu.:10.000          3rd Qu.:198.0   3rd Qu.:46.00
##  Max.   :138.00      Max.   :55.000          Max.   :265.0   Max.   :61.00
##  Praxis_rectangular Length_rectangular Major_variance  Minor_variance
##  Min.   :17.00      Min.   :118        Min.   :130.0   Min.   : 184.0
##  1st Qu.:19.00      1st Qu.:137        1st Qu.:167.0   1st Qu.: 318.2
##  Median :20.00      Median :146        Median :178.5   Median : 364.0
```

```
##  Mean   :20.58      Mean   :148        Mean   :188.6   Mean   : 439.9
##  3rd Qu.:23.00      3rd Qu.:159        3rd Qu.:217.0   3rd Qu.: 587.0
##  Max.   :29.00      Max.   :188        Max.   :320.0   Max.   :1018.0
##  Gyration_radius Major_skewness   Minor_skewness   Minor_kurtosis
##  Min.   :109.0   Min.   : 59.00   Min.   : 0.000   Min.   : 0.0
##  1st Qu.:149.0   1st Qu.: 67.00   1st Qu.: 2.000   1st Qu.: 5.0
##  Median :173.0   Median : 71.50   Median : 6.000   Median :11.0
##  Mean   :174.7   Mean   : 72.46   Mean   : 6.377   Mean   :12.6
##  3rd Qu.:198.0   3rd Qu.: 75.00   3rd Qu.: 9.000   3rd Qu.:19.0
##  Max.   :268.0   Max.   :135.00   Max.   :22.000   Max.   :41.0
##  Major_kurtosis  Hollows_ratio      Class
##  Min.   :176.0   Min.   :181.0   Length:846
##  1st Qu.:184.0   1st Qu.:190.2   Class :character
##  Median :188.0   Median :197.0   Mode  :character
##  Mean   :188.9   Mean   :195.6
##  3rd Qu.:193.0   3rd Qu.:201.0
##  Max.   :206.0   Max.   :211.0
```

Observamos que la variable "Class" tiene entre sus valores espacios al principio y al final de cada string.

```
vehicle[,'Class'] <- str_remove_all(vehicle$Class," ")
head(vehicle)
```

```
##   Compactness Circularity Distance_circularity Radius_ratio Praxis_aspect_ratio
## 1          95          48                  83          178                  72
## 2          91          41                  84          141                  57
## 3         104          50                 106          209                  66
## 4          93          41                  82          159                  63
## 5          85          44                  70          205                 103
## 6         107          57                 106          172                  50
##   Max_length_aspect_ratio Scatter_ratio Elongatedness Praxis_rectangular
## 1                      10           162            42                 20
## 2                       9           149            45                 19
## 3                      10           207            32                 23
## 4                       9           144            46                 19
## 5                      52           149            45                 19
## 6                       6           255            26                 28
##   Length_rectangular Major_variance Minor_variance Gyration_radius
## 1                159            176            379             184
## 2                143            170            330             158
## 3                158            223            635             220
## 4                143            160            309             127
## 5                144            241            325             188
## 6                169            280            957             264
##   Major_skewness Minor_skewness Minor_kurtosis Major_kurtosis Hollows_ratio
## 1             70              6             16            187           197
## 2             72              9             14            189           199
## 3             73             14              9            188           196
## 4             63              6             10            199           207
## 5            127              9             11            180           183
## 6             85              5              9            181           183
##   Class
## 1   van
## 2   van
## 3  saab
```

```
## 4   van
## 5   bus
## 6   bus
```

Una vez arreglado, pasamos a la limpieza de los datos.

Limpieza de los datos: missing values

Se buscan los missing values y datos duplicados.

```
# Comprobación de missing values
sum(is.na(vehicle))
```

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

```
sum(is.null(vehicle))
```

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

```
# datos duplicados

sum(duplicated(vehicle))
```

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

Limpieza de los datos: anomalías en una dimensión

Para la obtención de las anomalías, se calculan los cuartiles y el rango intercuartílico para determinar los límites superior e inferior. El valor de estos límites determina si el valor de una observación es atípico para una variable determinada.

```
# Outliers en una dimensión
# Calculamos rango intercuartílico de todas las variables
vehicle.IQR <- vehicle %>% select(-Class) %>% apply(2, IQR)
vehicle.Quartiles <- vehicle %>% select(-Class) %>% apply(2, quantile,c(0.25,0.75))
Upper.limit <- vehicle.Quartiles[2,]+1.5*vehicle.IQR
Upper.limit
```

```
##             Compactness              Circularity    Distance_circularity
##                 119.500                   62.500                  140.000
##            Radius_ratio     Praxis_aspect_ratio Max_length_aspect_ratio
##                 276.000                   77.000                   14.500
##           Scatter_ratio           Elongatedness      Praxis_rectangular
##                 275.625                   65.500                   29.000
##      Length_rectangular          Major_variance          Minor_variance
##                 192.000                  292.000                  990.125
##         Gyration_radius          Major_skewness          Minor_skewness
##                 271.500                   87.000                   19.500
##          Minor_kurtosis          Major_kurtosis           Hollows_ratio
##                  40.000                  206.500                  217.125
```

```
Lower.limit <- vehicle.Quartiles[1,]-1.5*vehicle.IQR
Lower.limit
```

```
##             Compactness              Circularity    Distance_circularity
##                  67.500                   26.500                   28.000
##            Radius_ratio     Praxis_aspect_ratio Max_length_aspect_ratio
##                  60.000                   45.000                    2.500
##           Scatter_ratio           Elongatedness      Praxis_rectangular
##                  68.625                   13.500                   13.000
##      Length_rectangular          Major_variance          Minor_variance
```

```
##                104.000                 92.000               -84.875
##         Gyration_radius          Major_skewness          Minor_skewness
##                 75.500                 55.000                 -8.500
##          Minor_kurtosis          Major_kurtosis           Hollows_ratio
##                 -16.000                170.500                174.125
```

Con estos límites podremos determinar los outliers de cada variable, aquellos valores que estén por encima del umbral superior y por debajo del inferior serán considerados outliers. De esta forma calculamos las posiciones de los datos atípicos de cada atributo:

```
# Obtenemos los outliers para cada variable y calculamos sus posiciones
# Compactness
Compactness.outliers <- which(vehicle$Compactness>Upper.limit['Compactness'] | vehicle$Compactness<Lowe
# Circularity
Circularity.outliers <- which(vehicle$Circularity>Upper.limit['Circularity'] | vehicle$Circularity<Lowe
# Distance_circularity
Distance_circularity.outliers <- which(vehicle$Distance_circularity>Upper.limit['Distance_circularity']
# Radius_ratio
Radius_ratio.outliers <- which(vehicle$Radius_ratio>Upper.limit['Radius_ratio'] | vehicle$Radius_ratio<
# Praxis_aspect_ratio
Praxis_aspect_ratio.outliers <- which(vehicle$Praxis_aspect_ratio>Upper.limit['Praxis_aspect_ratio'] | 
# Max_length_aspect_ratio
Max_length_aspect_ratio.outliers <- which(vehicle$Max_length_aspect_ratio>Upper.limit['Max_length_aspec
# Scatter_ratio
Scatter_ratio.outliers <- which(vehicle$Scatter_ratio>Upper.limit['Scatter_ratio'] | vehicle$Scatter_ra
# Elongatedness
Elongatedness.outliers <- which(vehicle$Elongatedness>Upper.limit['Elongatedness'] | vehicle$Elongatedn
# Praxis_rectangular
Praxis_rectangular.outliers <- which(vehicle$Praxis_rectangular>Upper.limit['Praxis_rectangular'] | vehi
# Length_rectangular
Length_rectangular.outliers <- which(vehicle$Length_rectangular>Upper.limit['Length_rectangular'] | vehi
# Major_variance
Major_variance.outliers <- which(vehicle$Major_variance>Upper.limit['Major_variance'] | vehicle$Major_va
# Minor_variance
Minor_variance.outliers <- which(vehicle$Minor_variance>Upper.limit['Minor_variance'] | vehicle$Minor_va
# Gyration_radius
Gyration_radius.outliers <- which(vehicle$Gyration_radius>Upper.limit['Gyration_radius '] | vehicle$Gyra
# Major_skewness
Major_skewness.outliers <- which(vehicle$Major_skewness>Upper.limit['Major_skewness'] | vehicle$Major_sk
# Minor_skewness
Minor_skewness.outliers <- which(vehicle$Minor_skewness>Upper.limit['Minor_skewness'] | vehicle$Minor_sk
# Minor_kurtosis
Minor_kurtosis.outliers <- which(vehicle$Minor_kurtosis>Upper.limit['Minor_kurtosis'] | vehicle$Minor_ku
# Major_kurtosis
Major_kurtosis.outliers <- which(vehicle$Major_kurtosis>Upper.limit['Major_kurtosis'] | vehicle$Major_ku
# Hollows_ratio
Hollows_ratio.outliers <- which(vehicle$Hollows_ratio>Upper.limit['Hollows_ratio'] | vehicle$Hollows_ra
```

Cabe destacar que muchas de las variables no tienen datos atípicos. En el análisis gráfico veremos en detalle que atributos tienen outliers y de qué tipo.

También se ha calculado un vector que incluya las posiciones de todas las observaciones cuyas variables tomen al menos un valor atípico, es decir, se han calculado todas las observaciones atípicas. Veremos también que cantidad de outliers tenemos para tener una idea de la proporción de datos atípicos en el dataset.

```
# Comprobamos cantidad de outliers

vehicle.outliers <- unique(c(Compactness.outliers, Circularity.outliers,
                             Distance_circularity.outliers, Radius_ratio.outliers,
                             Praxis_aspect_ratio.outliers, Max_length_aspect_ratio.outliers,
                             Scatter_ratio.outliers, Elongatedness.outliers,
                             Praxis_rectangular.outliers, Length_rectangular.outliers,
                             Major_variance.outliers, Minor_variance.outliers,
                             Gyration_radius.outliers, Major_skewness.outliers,
                             Minor_skewness.outliers, Minor_kurtosis.outliers,
                             Major_kurtosis.outliers, Hollows_ratio.outliers))

length(vehicle.outliers)
```

```
## [1] 33
```

Tan sólo tenemos 33 observaciones en las que almenos uno de sus atributos tome un valor atípico, en comparación a las 846 observaciones totales de las que disponemos, son menos de un 5% por lo que se considera que prescindir de ellos no afectaría negativamente a la clasificación.

Podemos comprobar también en que proporción están distribuidos los outliers entre las diferentes clases.

```
table(vehicle[vehicle.outliers,"Class"])
```

```
##
##  bus opel saab  van
##   10    4    9   10
```

Teniendo en cuenta la cantidad de datos totales que tenemos por clase:

```
# calculamos número de elementos por clase
counts.Class <- table(vehicle$Class)
counts.Class
```

```
##
##  bus opel saab  van
##  218  212  217  199
```

Quitar los outliers podría desbalancear en cierta medida la clase "van" que ya se encuentra de manera algo menos frecuente que las demás.

Resumen de los datos

En esta sección se realizará el estudio de las variables mediante diferentes medidas de estadística descriptiva y representaciones gráficas, tanto de cada variable individualmente, como de algunas combinaciones entre ellas y la de salida.

Medidas de tendencia central

Nos permiten obtener cierta idea de centro de nuestros datos.

```
# Para obtener estos valores:
vehicle.mean <- vehicle %>% select(-Class) %>% map_dbl(mean)
vehicle.median <- vehicle %>% select(-Class) %>% map_dbl(median)

vehicle.mean
```

```
##             Compactness              Circularity    Distance_circularity
##               93.678487                44.861702                82.088652
##            Radius_ratio     Praxis_aspect_ratio Max_length_aspect_ratio
```

```
##              168.940898              61.693853               8.567376
##           Scatter_ratio           Elongatedness      Praxis_rectangular
##              168.839243              40.933806              20.582742
##      Length_rectangular          Major_variance          Minor_variance
##              147.998818             188.625296             439.911348
##         Gyration_radius          Major_skewness          Minor_skewness
##              174.703310              72.462175               6.377069
##          Minor_kurtosis          Major_kurtosis           Hollows_ratio
##               12.599291             188.932624             195.632388
```

```
vehicle.median
```

```
##             Compactness             Circularity    Distance_circularity
##                    93.0                    44.0                    80.0
##            Radius_ratio     Praxis_aspect_ratio Max_length_aspect_ratio
##                   167.0                    61.0                     8.0
##           Scatter_ratio           Elongatedness      Praxis_rectangular
##                   157.0                    43.0                    20.0
##      Length_rectangular          Major_variance          Minor_variance
##                   146.0                   178.5                   364.0
##         Gyration_radius          Major_skewness          Minor_skewness
##                   173.0                    71.5                     6.0
##          Minor_kurtosis          Major_kurtosis           Hollows_ratio
##                    11.0                   188.0                   197.0
```

Estas dos medídas de tendencia central ya nos permiten tener cierta idea sobre la asimetría de las distribuciones. En este caso todas las variables menos "Hollows ratio" tienen una media mayor a la mediana, por lo que podríamos afirmar que el "skewness" es positivo. Más adelante se calcula esta medida y podremos comprobar si esta afirmación es cierta.

Medidas de dispersión

Estas medidas nos proporcionan información de la dispersión de nuestros datos, y de la distancia a la que se encuentran del centro. Los cálculos para los cuartiles y el rango intercuartílico ya se ha hecho anteriormente para el estudio de los datos atípicos.

```
# Medidas de dispersión
# Para el Rango, máximo y mínimo:
vehicle.min.max <- vehicle %>% select(-Class) %>% apply(2,range)
vehicle.range <- vehicle.min.max[2,]-vehicle.min.max[1,]
vehicle.min.max.range <- rbind(vehicle.min.max, vehicle.range)
rownames(vehicle.min.max.range) <- c('min', 'max', 'range')
vehicle.min.max.range
```

```
##       Compactness Circularity Distance_circularity Radius_ratio
## min            73          33                   40          104
## max           119          59                  112          333
## range          46          26                   72          229
##       Praxis_aspect_ratio Max_length_aspect_ratio Scatter_ratio Elongatedness
## min                    47                      2           112            26
## max                   138                     55           265            61
## range                  91                     53           153            35
##       Praxis_rectangular Length_rectangular Major_variance Minor_variance
## min                   17                118            130            184
## max                   29                188            320           1018
## range                 12                 70            190            834
##       Gyration_radius Major_skewness Minor_skewness Minor_kurtosis
```

```
## min               109             59              0              0
## max               268            135             22             41
## range             159             76             22             41
##       Major_kurtosis Hollows_ratio
## min              176           181
## max              206           211
## range             30            30
```

```
# primer cuartil, tercer cuartil y rango intercuartílico.
vehicle.Quartiles
```

```
##     Compactness Circularity Distance_circularity Radius_ratio
## 25%          87          40                   70          141
## 75%         100          49                   98          195
##     Praxis_aspect_ratio Max_length_aspect_ratio Scatter_ratio Elongatedness
## 25%                  57                       7        146.25            33
## 75%                  65                      10        198.00            46
##     Praxis_rectangular Length_rectangular Major_variance Minor_variance
## 25%                 19                137            167         318.25
## 75%                 23                159            217         587.00
##     Gyration_radius Major_skewness Minor_skewness Minor_kurtosis Major_kurtosis
## 25%             149             67              2              5            184
## 75%             198             75              9             19            193
##     Hollows_ratio
## 25%        190.25
## 75%        201.00
```

```
vehicle.IQR
```

```
##             Compactness              Circularity    Distance_circularity
##                   13.00                     9.00                    28.00
##            Radius_ratio     Praxis_aspect_ratio Max_length_aspect_ratio
##                   54.00                     8.00                     3.00
##           Scatter_ratio           Elongatedness      Praxis_rectangular
##                   51.75                    13.00                     4.00
##      Length_rectangular          Major_variance          Minor_variance
##                   22.00                    50.00                  268.75
##         Gyration_radius          Major_skewness          Minor_skewness
##                   49.00                     8.00                     7.00
##          Minor_kurtosis          Major_kurtosis           Hollows_ratio
##                   14.00                     9.00                    10.75
```

```
# varianza, desviación típica y desviación absoluta de la mediana
vehicle.var <- vehicle %>% select(-Class) %>% map_dbl(var)
vehicle.sd <- vehicle %>% select(-Class) %>% map_dbl(sd)
vehicle.mad <- vehicle %>% select(-Class) %>% map_dbl(mad)

vehicle.var
```

```
##             Compactness              Circularity    Distance_circularity
##               67.806566                38.067242               248.741244
##            Radius_ratio     Praxis_aspect_ratio Max_length_aspect_ratio
##             1120.387035                62.224507                21.171195
##           Scatter_ratio           Elongatedness      Praxis_rectangular
##             1105.228565                61.020465                 6.719181
##      Length_rectangular          Major_variance          Minor_variance
##              210.704141               985.635762            31220.279705
```

```
##         Gyration_radius          Major_skewness          Minor_skewness
##             1059.274001               56.054781               24.190195
##          Minor_kurtosis          Major_kurtosis           Hollows_ratio
##                79.767053               37.994272               55.335707
```

```
vehicle.sd
```

```
##             Compactness             Circularity    Distance_circularity
##                8.234474                6.169866               15.771533
##            Radius_ratio     Praxis_aspect_ratio Max_length_aspect_ratio
##                33.472183                7.888251                4.601217
##           Scatter_ratio          Elongatedness      Praxis_rectangular
##                33.244978                7.811560                2.592138
##      Length_rectangular          Major_variance          Minor_variance
##                14.515652               31.394837              176.692614
##         Gyration_radius          Major_skewness          Minor_skewness
##                32.546490                7.486974                4.918353
##          Minor_kurtosis          Major_kurtosis           Hollows_ratio
##                 8.931240                6.163949                7.438797
```

```
vehicle.mad
```

```
##             Compactness             Circularity    Distance_circularity
##                  8.8956                  7.4130                 17.7912
##            Radius_ratio     Praxis_aspect_ratio Max_length_aspect_ratio
##                 40.0302                  5.9304                  2.9652
##           Scatter_ratio          Elongatedness      Praxis_rectangular
##                 29.6520                  8.8956                  2.9652
##      Length_rectangular          Major_variance          Minor_variance
##                 16.3086                 28.9107                 140.1057
##         Gyration_radius          Major_skewness          Minor_skewness
##                 36.3237                  6.6717                  5.9304
##          Minor_kurtosis          Major_kurtosis           Hollows_ratio
##                  8.8956                  5.9304                  7.4130
```

Las mayores varianzas se encuentran en las variables "Minor_variance", "Major_variance", "Gyration_radiusRadius_ratio" y "Scatter_ratio", esto será aparente al estudiar las distribuciones de estos atributos en el análisis gráfico por la escala que utilizan.

Medidas de forma

Como su nombre indica, las medidas de forma nos proporcionan información sobre la forma de la distribución para una variable. Ya sea una medida de asimetría (skewness) o el tipo de pico que presenta (kurtosis).

```
# Medidas de forma
vehicle.skewness <- vehicle %>% select(-Class) %>% map_dbl(skewness)
vehicle.kurtosis <- vehicle %>% select(-Class) %>% map_dbl(kurtosis)

vehicle.skewness
```

```
##             Compactness             Circularity    Distance_circularity
##              0.38059429              0.26233259              0.10703038
##            Radius_ratio     Praxis_aspect_ratio Max_length_aspect_ratio
##              0.39001338              3.81478096              6.76636927
##           Scatter_ratio          Elongatedness      Praxis_rectangular
##              0.60470440              0.04776018              0.76931732
##      Length_rectangular          Major_variance          Minor_variance
##              0.25590440              0.65065760              0.83435412
```

```
##         Gyration_radius          Major_skewness          Minor_skewness
##              0.27973343              2.06890653              0.77241926
##          Minor_kurtosis          Major_kurtosis           Hollows_ratio
##              0.68810263              0.24809990             -0.22593977
```

```
vehicle.kurtosis
```

```
##             Compactness             Circularity    Distance_circularity
##                2.460799                2.073385                2.020218
##            Radius_ratio     Praxis_aspect_ratio Max_length_aspect_ratio
##                3.292961               32.653111               61.023934
##           Scatter_ratio           Elongatedness      Praxis_rectangular
##                2.380682                2.133946                2.602175
##      Length_rectangular          Major_variance          Minor_variance
##                2.227362                3.110492                2.778336
##         Gyration_radius          Major_skewness          Minor_skewness
##                2.505560               14.298613                3.080743
##          Minor_kurtosis          Major_kurtosis           Hollows_ratio
##                2.852791                2.402333                2.184281
```

Como se había mencionado anteriormente, efectivamente las distribuciones de todas las variables menos "Hollows_ratio" poseen un "skewness" positivo. En cuanto a la medida de kurtosis, parece ser que las distribuciones con un pico más pronunciado son "Praxis_aspect_ratio" y "Max_length_aspect_ratio".

Comprobación de normalidad

Ciertos algoritmos de clasificación y muchos procesos estadísticos asumen que los datos se distribuyen normalemente, por ejemplo ciertos tests estadísticos requieren que nuestras variables se distribuyan normalmente. Para comprobar la normalidad de nuestros datos, se aplicará el test de Shapiro-Wilk. Más adelante se hará el estudio de normalidad por clase para comprobar las suposiciones que se hacen de cara a los clasificadores LDA y QDA.

```
# Comprobamos normalidad con shapiro

vehicle.shapiro <- apply(vehicle[,-19],2,shapiro.test)
vehicle.shapiro
```

```
## $Compactness
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.97712, p-value = 2.99e-10
##
##
## $Circularity
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.96404, p-value = 1.415e-13
##
##
## $Distance_circularity
##
##  Shapiro-Wilk normality test
##
```

```
## data:  newX[, i]
## W = 0.95792, p-value = 7.422e-15
##
##
## $Radius_ratio
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.96983, p-value = 3.194e-12
##
##
## $Praxis_aspect_ratio
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.75081, p-value < 2.2e-16
##
##
## $Max_length_aspect_ratio
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.4661, p-value < 2.2e-16
##
##
## $Scatter_ratio
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.93051, p-value < 2.2e-16
##
##
## $Elongatedness
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.95968, p-value = 1.68e-14
##
##
## $Praxis_rectangular
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.89373, p-value < 2.2e-16
##
##
## $Length_rectangular
##
```

```
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.9764, p-value = 1.846e-10
##
##
## $Major_variance
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.9469, p-value < 2.2e-16
##
##
## $Minor_variance
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.8993, p-value < 2.2e-16
##
##
## $Gyration_radius
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.98434, p-value = 7.267e-08
##
##
## $Major_skewness
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.87384, p-value < 2.2e-16
##
##
## $Minor_skewness
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.9368, p-value < 2.2e-16
##
##
## $Minor_kurtosis
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.94865, p-value < 2.2e-16
##
##
```

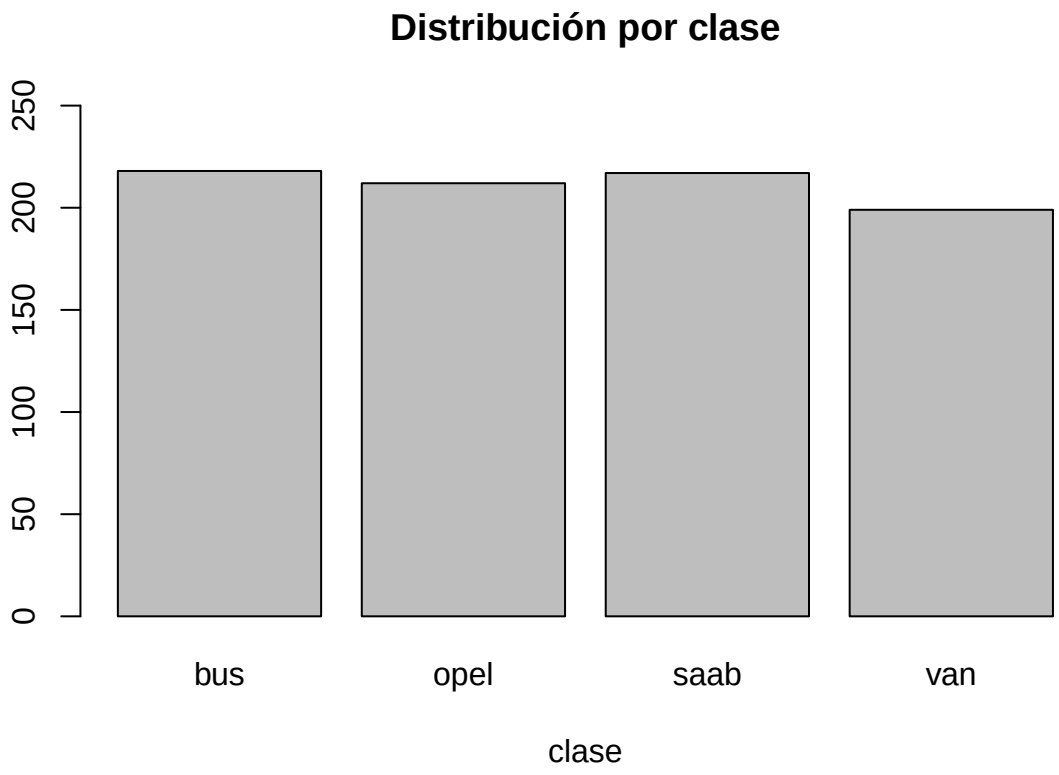
```
## $Major_kurtosis
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.98148, p-value = 7.101e-09
##
##
## $Hollows_ratio
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.96429, p-value = 1.606e-13
```

El p-valor que se obtiene para cada variable es muy bajo, menor al 0.05, por lo que el test nos informa a más de un 95% de confianza de que nuestras distribuciones son diferentes a una distribución normal.

Gráficas univariables de los atributos

Para esta sección, se han realizado dos diagramas para cada variable: un histograma con curva de densidadpara apreciar mejor la distribución y un diagrama de cajas para visualizar la naturaleza de los datos atípicos. Empezando por la variable de salida, que es la única categórica podemos representar la cantidad de elementos por categoría:

```
# Para la variable de salida Class
barplot(counts.Class, main='Distribución por clase', xlab='clase',ylim = c(0,250))
```



Ya se ha mencionado anteriormente que la clase menos representada es la de "van", pero aún así no hay

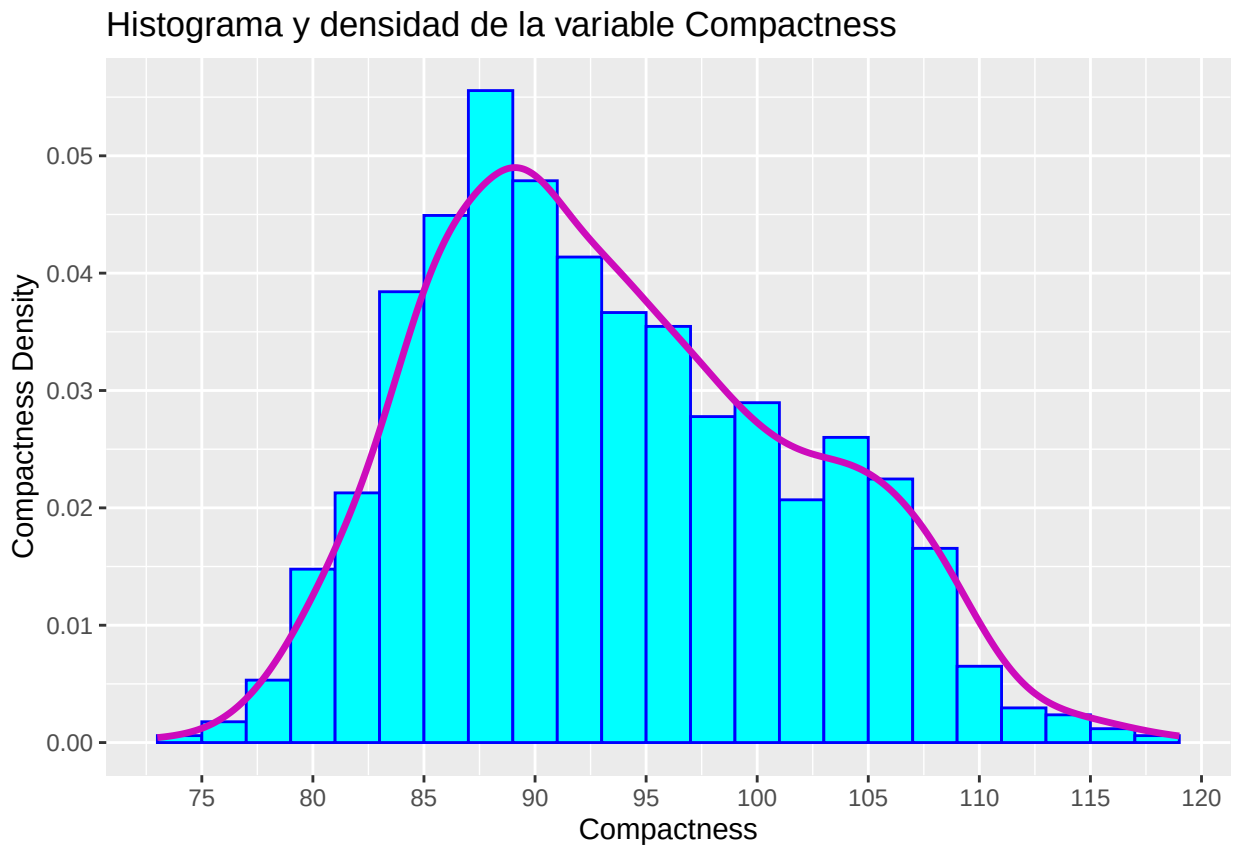
grandes desbalances entre las clases.

Pasamos pues a estudiar las variables cuantitativas discretas, empezando por “Compactness”:

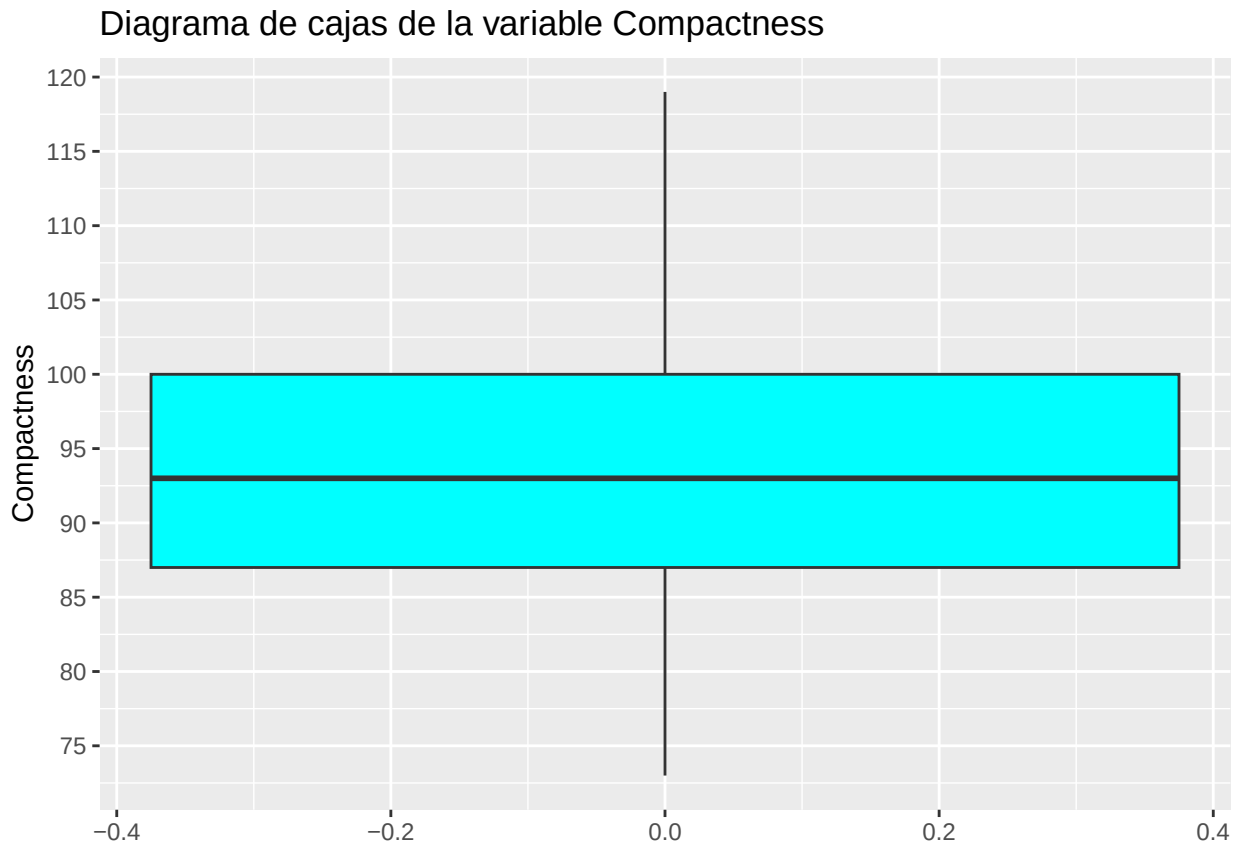
```
# Para Compactness
ggplot(vehicle, aes(x=Compactness)) +
  geom_histogram(aes(y=stat(density)),bins=24, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Compactness Density', title = 'Histograma y densidad de la variable Compactness')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=120, by=5))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```

```
## Warning: `stat(density)` was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
```



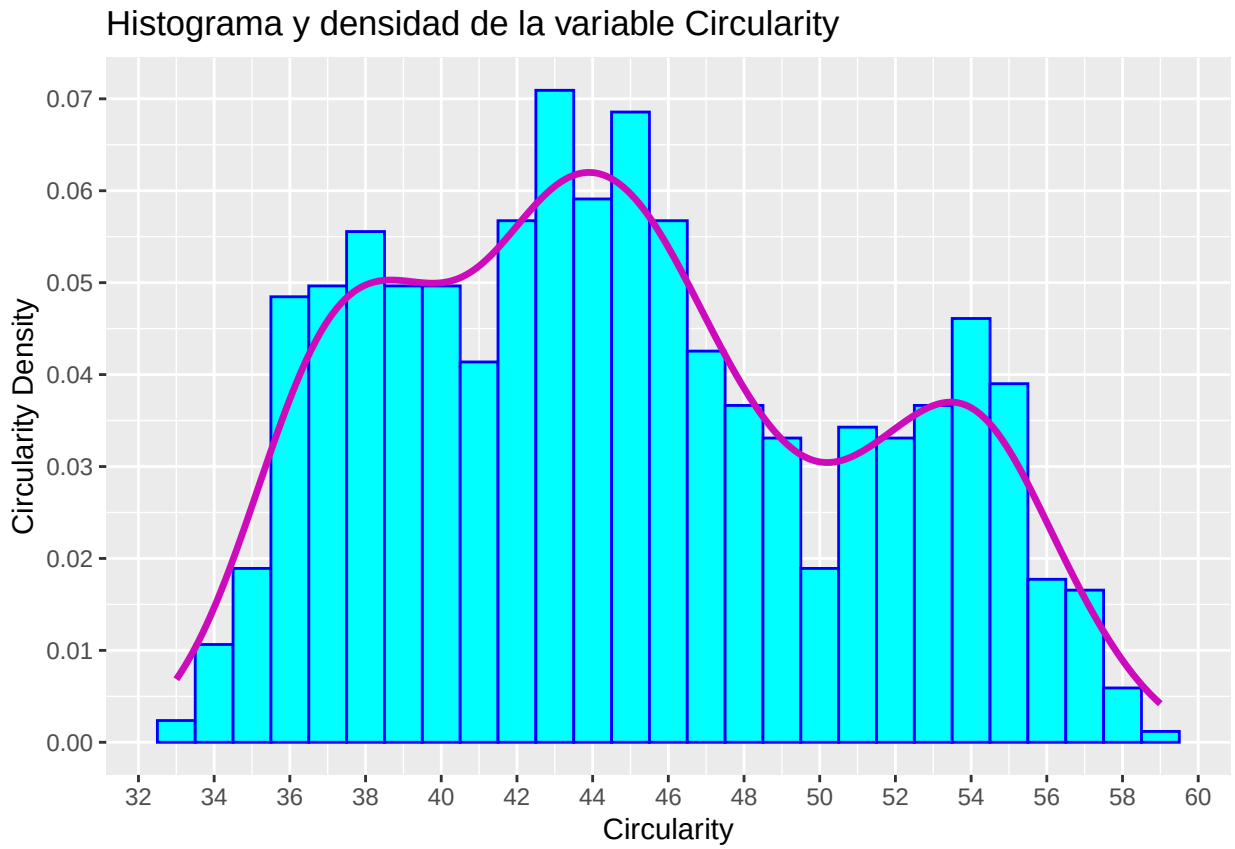

```
ggplot(vehicle, aes(y=Compactness)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Compactness', title = 'Diagrama de cajas de la variable Compactness')+
  scale_y_continuous(breaks = seq(from=0, to=120, by=5))
```



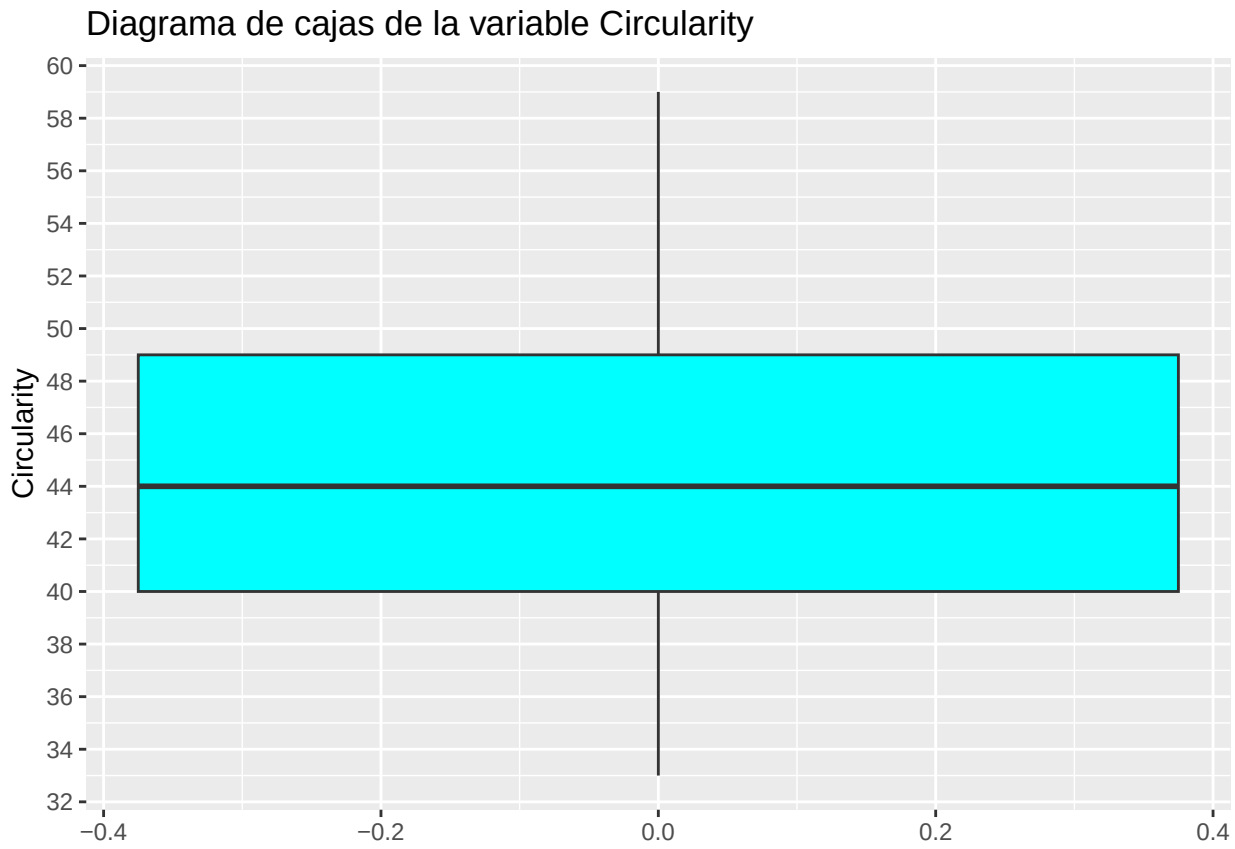
Se ha mencionado anteriormente que había ciertas variables que no tenían outliers, si nos fijamos en el diagrama de cajas, podemos ver que efectivamente esta es una de ellas. En cuanto a la distribucción se ve claramente el "skewness" positivo del que hemos hablado.

Analizamos la variable "Circularity":

```
# Para Circularity
ggplot(vehicle, aes(x=Circularity)) +
  geom_histogram(aes(y=stat(density)),bins=27, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Circularity Density', title = 'Histograma y densidad de la variable Circularity')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=60, by=2))
```



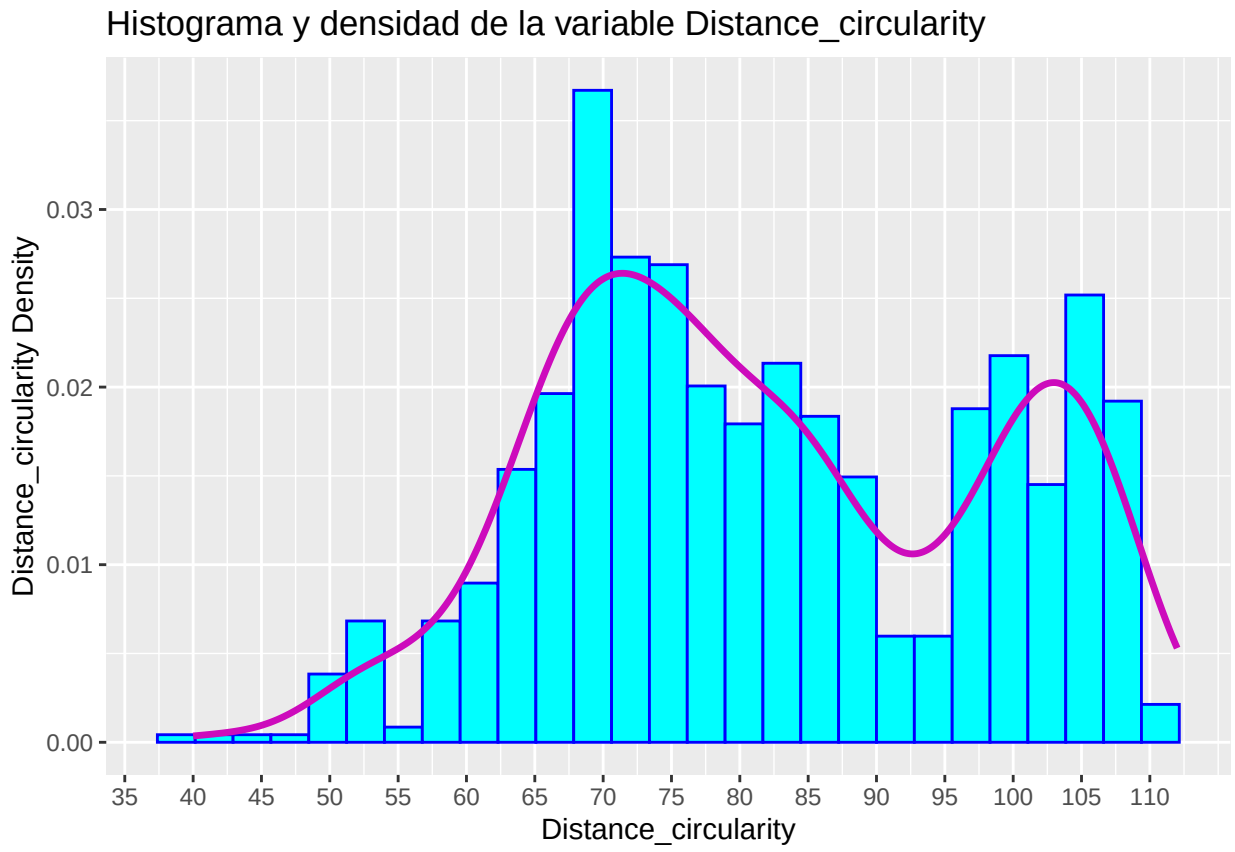

```
ggplot(vehicle, aes(y=Circularity)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Circularity', title = 'Diagrama de cajas de la variable Circularity')+
  scale_y_continuous(breaks = seq(from=0, to=60, by=2))
```



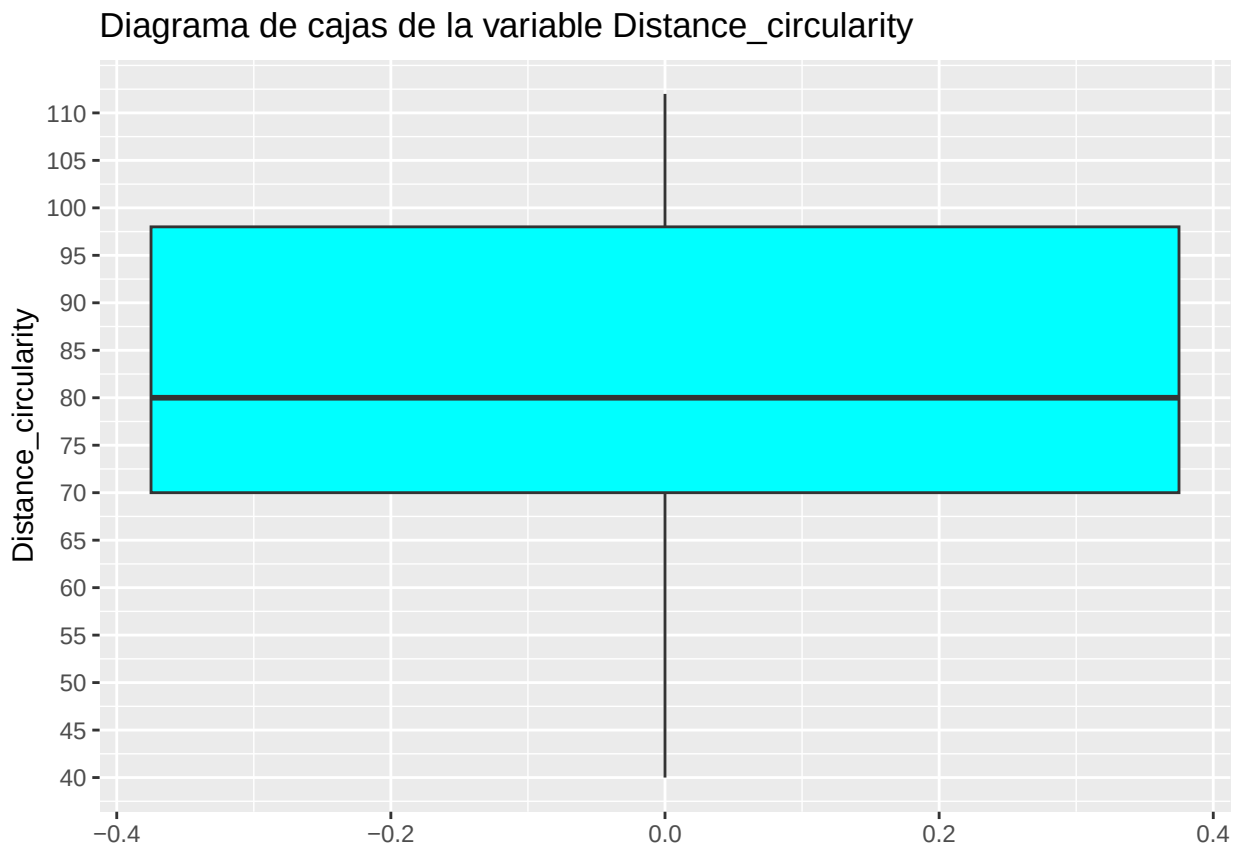
Tiene una distribución interesante, donde parece que se distingue una combinación de tres distribuciones por los tres picos que se observan, más adelante se descompondrá esta distribución por clases, con lo que se podrá estudiar esta afirmación. Tampoco parece tener outliers según el diagrama de cajas obtenido.

Para la variable "Distance_circularity":

```
# Para Distance_circularity
ggplot(vehicle, aes(x=Distance_circularity)) +
  geom_histogram(aes(y=stat(density)),bins=27, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Distance_circularity Density', title = 'Histograma y densidad de la variable Distance_circula
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=110, by=5))
```



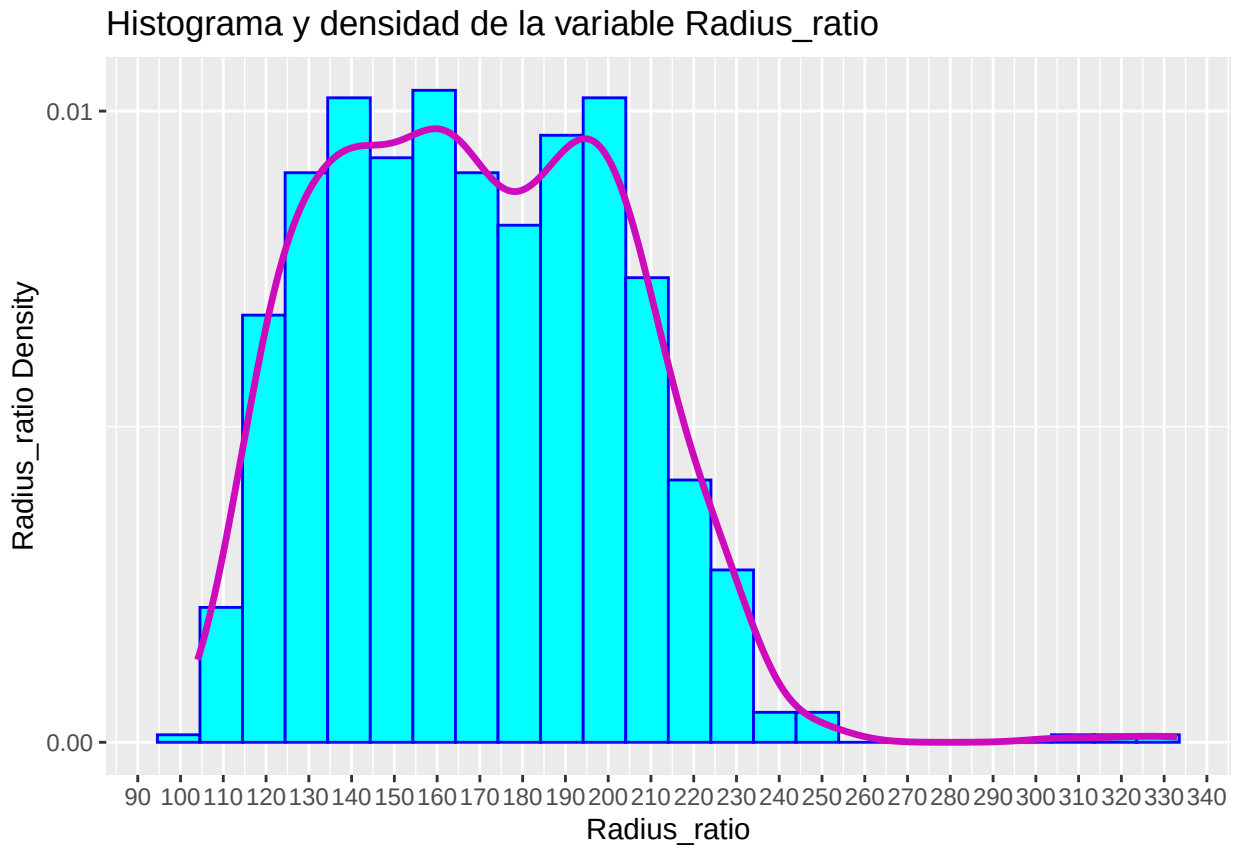

```
ggplot(vehicle, aes(y=Distance_circularity)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Distance_circularity', title = 'Diagrama de cajas de la variable Distance_circularity')+
  scale_y_continuous(breaks = seq(from=0, to=110, by=5))
```



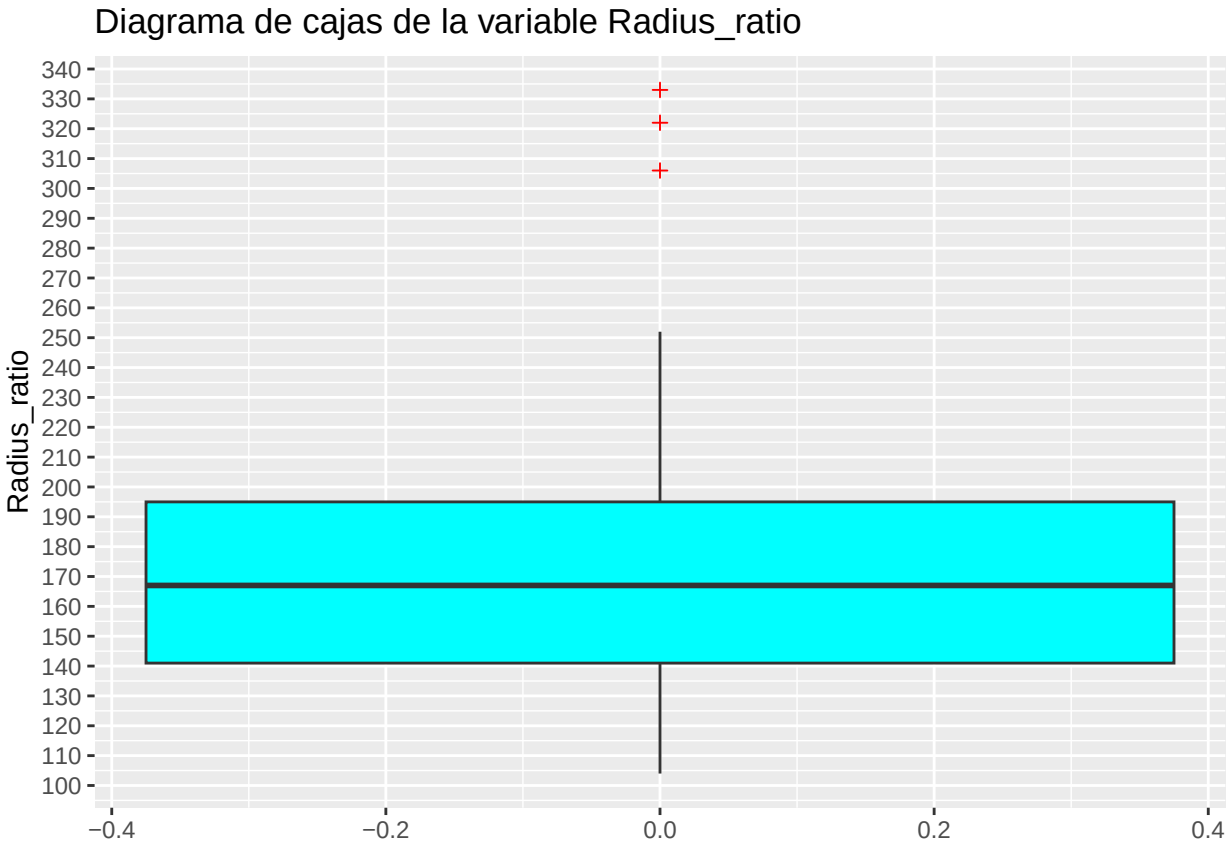
Distribución con dos picos, que anque por el gráfico parezca que pueda tener un "skewness" negativo, en los cálculos se ha visto que es ligeramente positivo. Tampoco tiene outliers.

Representamos "Para Radius_ratio":

```
# Para Radius_ratio
ggplot(vehicle, aes(x=Radius_ratio)) +
  geom_histogram(aes(y=stat(density)),bins=24, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Radius_ratio Density', title = 'Histograma y densidad de la variable Radius_ratio')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=400, by=10))
```



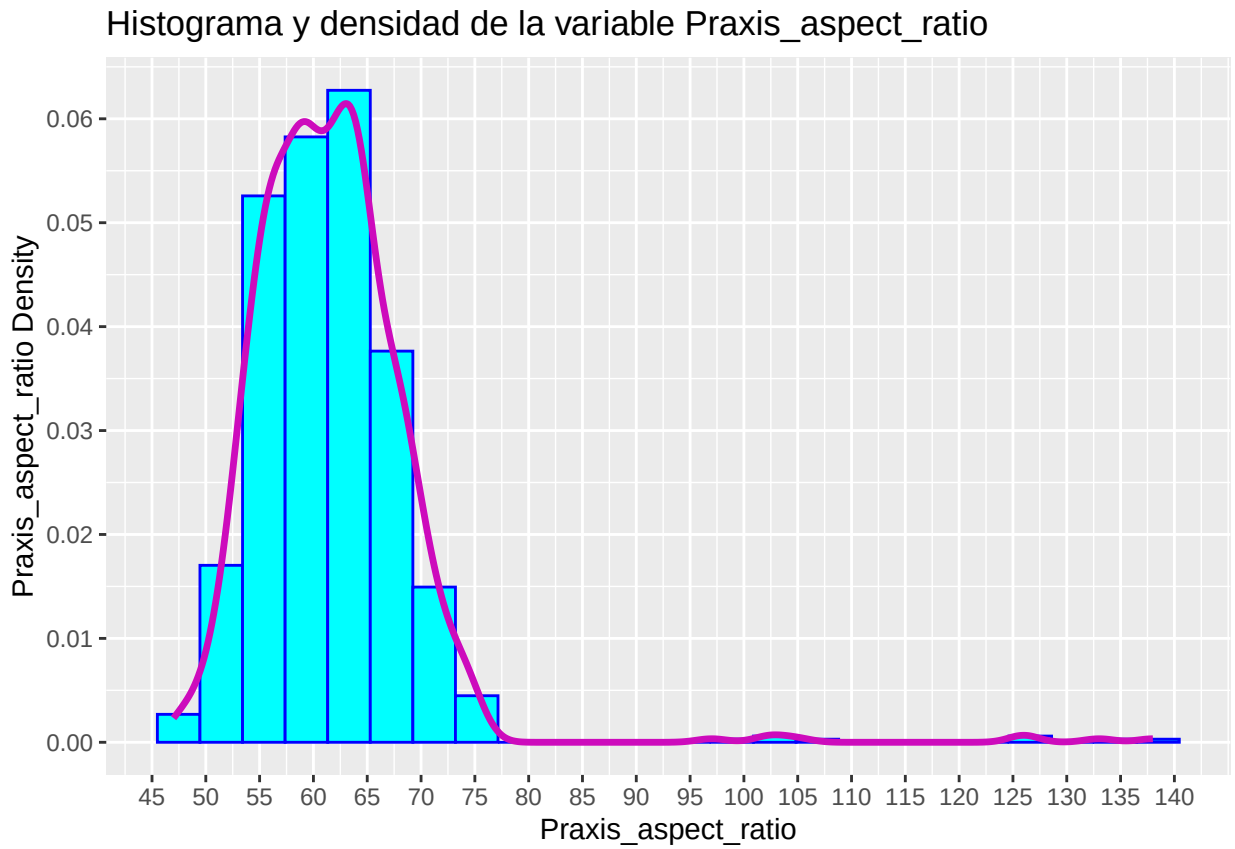

```
ggplot(vehicle, aes(y=Radius_ratio)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Radius_ratio', title = 'Diagrama de cajas de la variable Radius_ratio')+
  scale_y_continuous(breaks = seq(from=0, to=400, by=10))
```



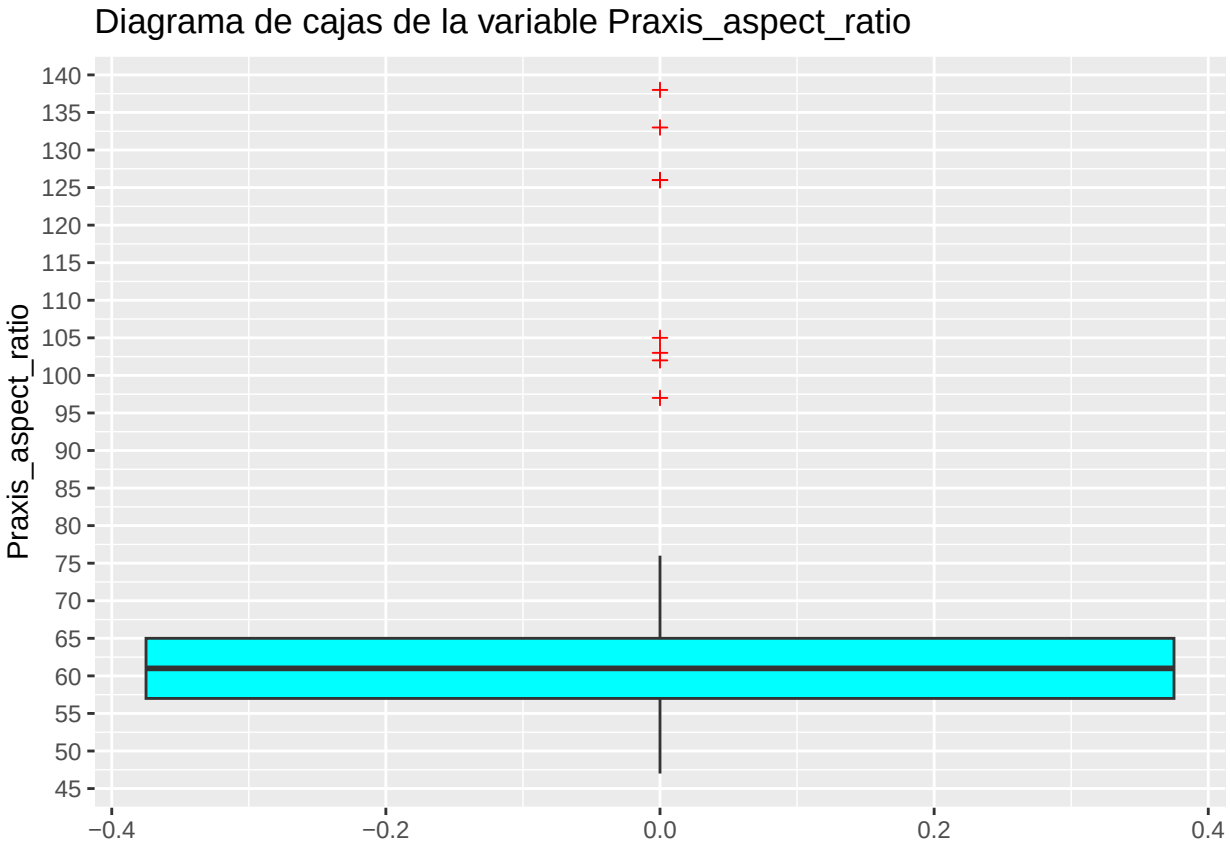
Esta distribución parece estar más centrada, pero tiene outliers de valores bastante altos (casi el doble de la mediana).

En el caso de "Praxis_aspect_ratio" tenemos:

```
# Para Praxis_aspect_ratio
ggplot(vehicle, aes(x=Praxis_aspect_ratio)) +
  geom_histogram(aes(y=stat(density)),bins=24, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Praxis_aspect_ratio Density', title = 'Histograma y densidad de la variable Praxis_aspect_rati
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=140, by=5))
```



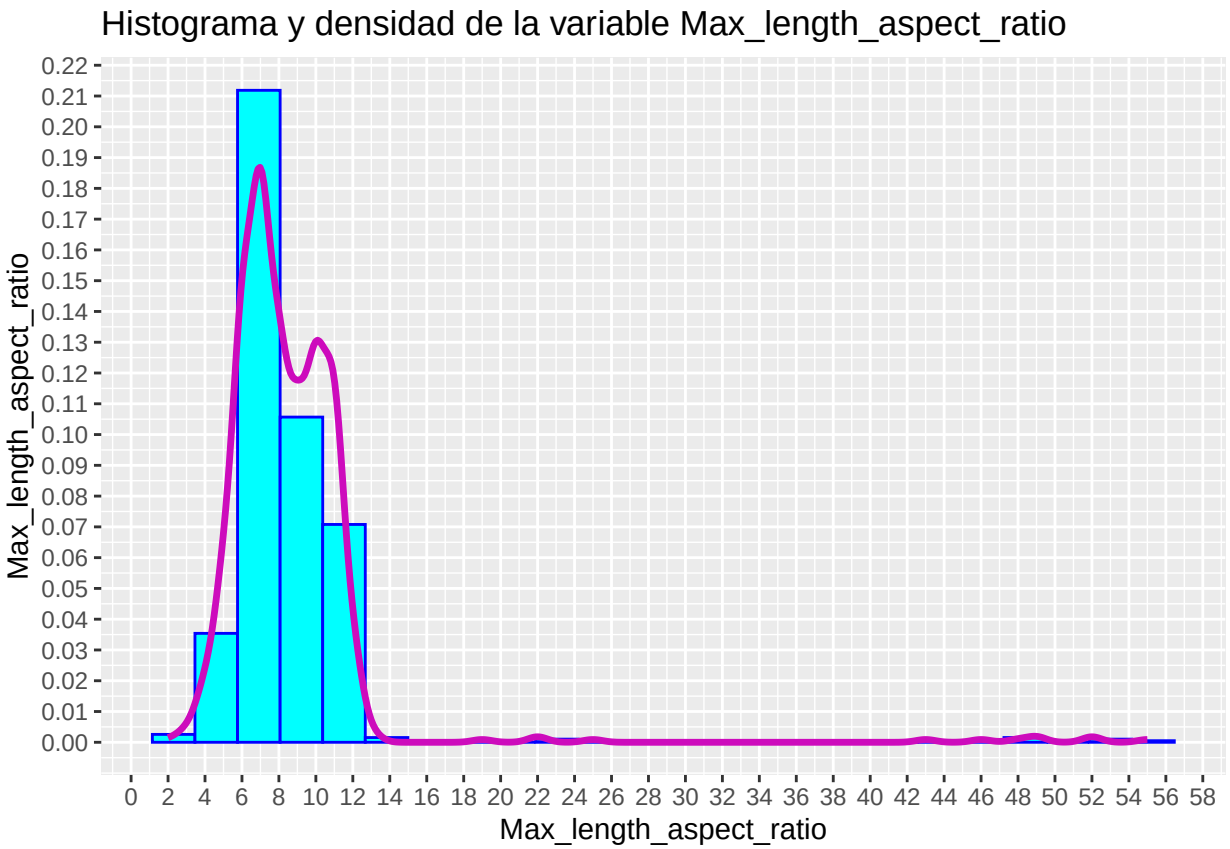

```
ggplot(vehicle, aes(y=Praxis_aspect_ratio)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Praxis_aspect_ratio', title = 'Diagrama de cajas de la variable Praxis_aspect_ratio')+
  scale_y_continuous(breaks = seq(from=0, to=140, by=5))
```



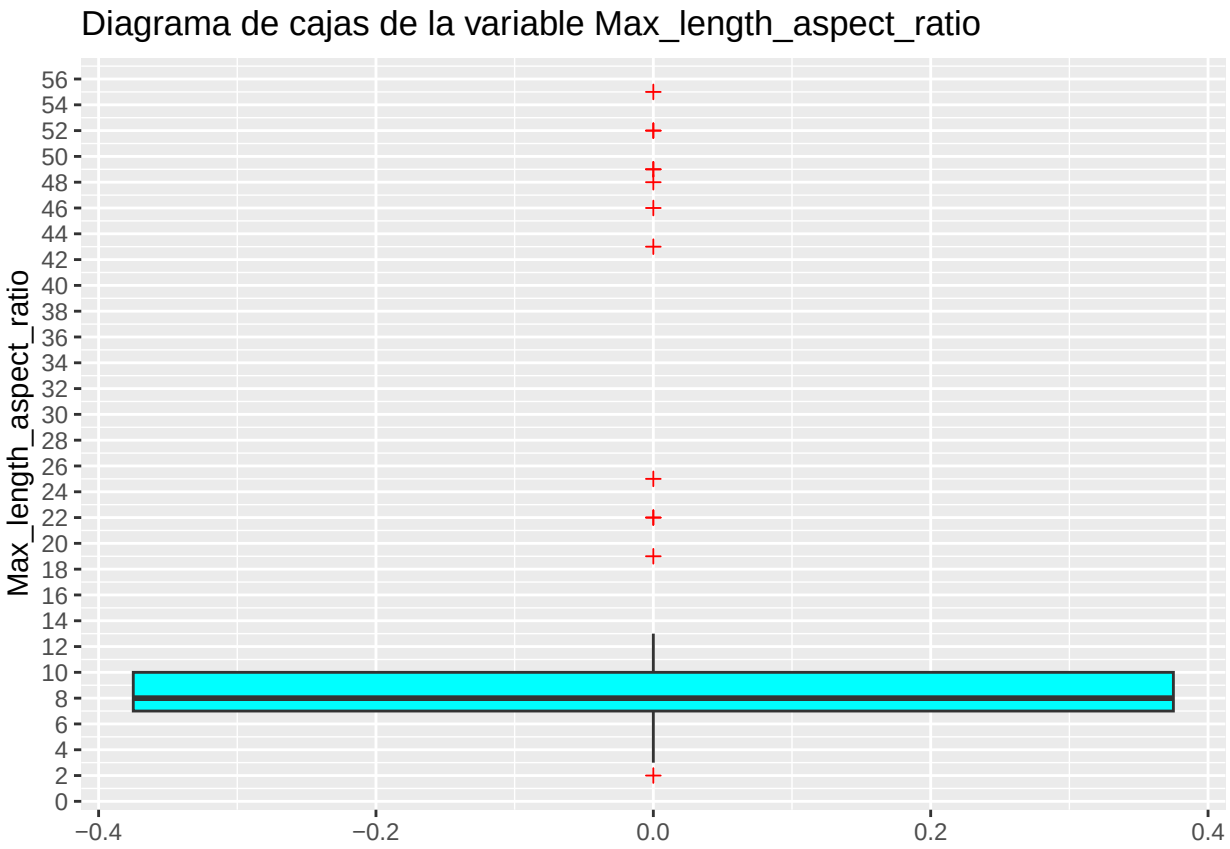
Podemos apreciar una distribución con un pico bastante pronunciado, justo como se había calculado en el apartado del cálculo de medidas de forma. En cuanto a los outliers parece que se pueden agrupar en dos grupos diferentes, unos con valores más extremos (en torno al doble del tercer cuartil) y el otro grupo con valores menos alejados de la distribución.

Para "Max_length_aspect_ratio":

```
# Para Max_length_aspect_ratio
ggplot(vehicle, aes(x=Max_length_aspect_ratio)) +
  geom_histogram(aes(y=stat(density)),bins=24, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Max_length_aspect_ratio', title = 'Histograma y densidad de la variable Max_length_aspect_rati
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=60, by=2))
```



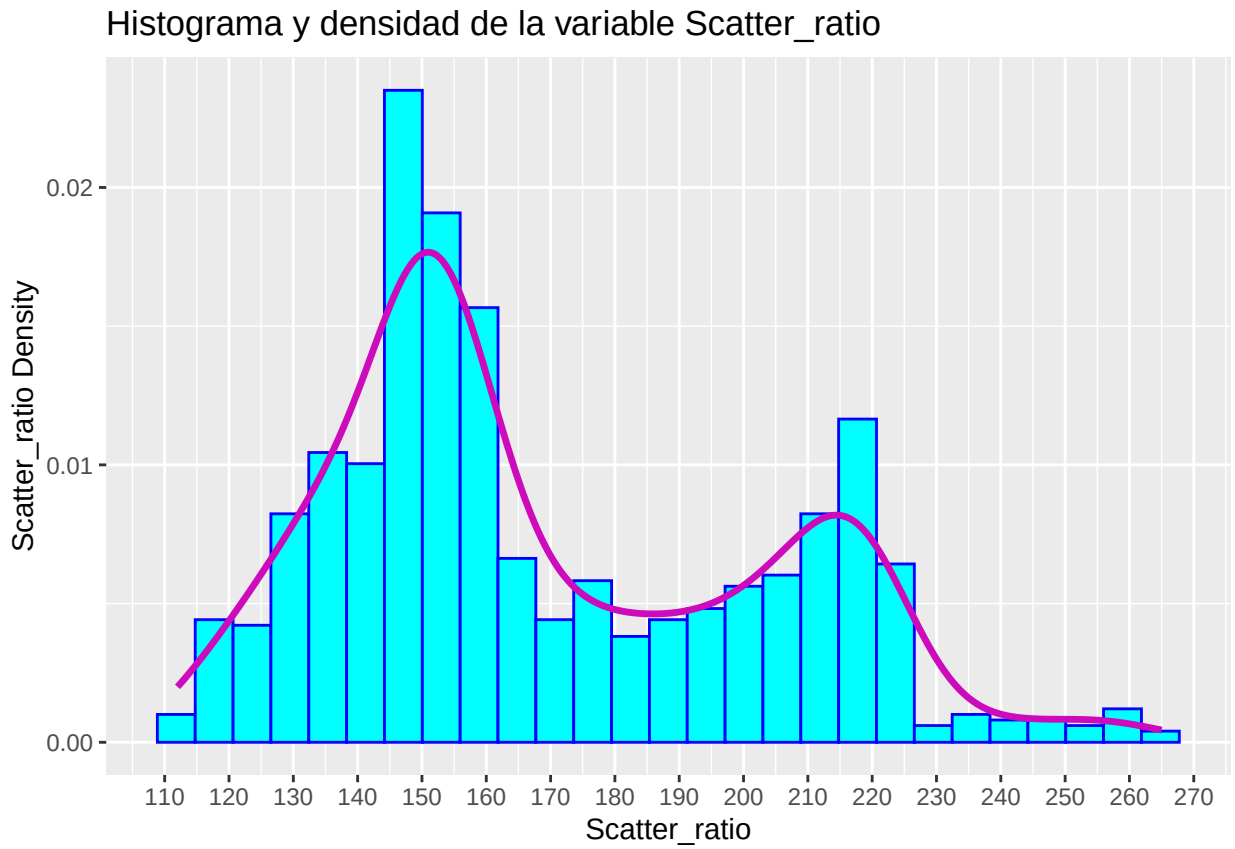

```
ggplot(vehicle, aes(y=Max_length_aspect_ratio)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Max_length_aspect_ratio', title = 'Diagrama de cajas de la variable Max_length_aspect_ratio')+
  scale_y_continuous(breaks = seq(from=0, to=60, by=2))
```



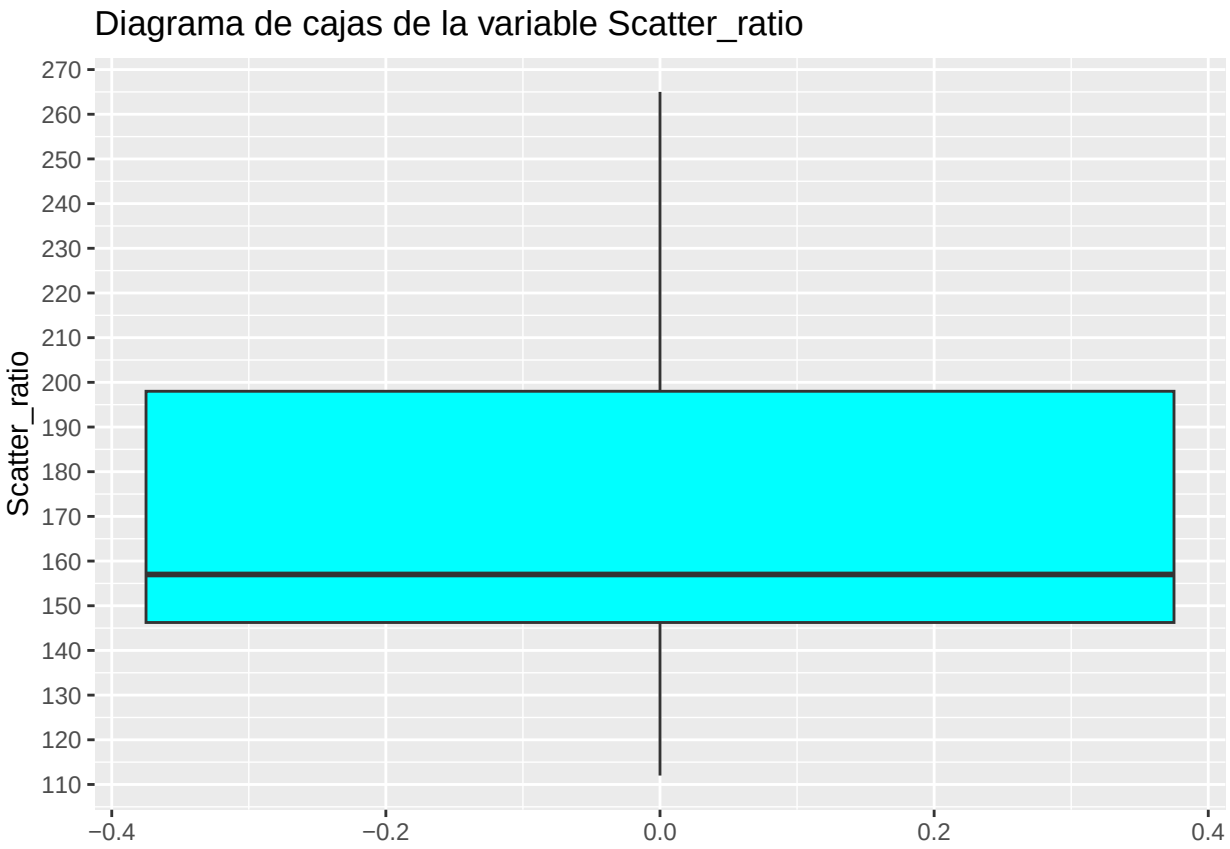
Un pico muy pronunciado en la distribución como se comentó con la medida de kurtosis y en este caso, esta distribución tiene el mayor número de outliers de las que se han visto hasta ahora.

Con "Scatter_ratio":

```
# Para Scatter_ratio
ggplot(vehicle, aes(x=Scatter_ratio)) +
  geom_histogram(aes(y=stat(density)),bins=27, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Scatter_ratio Density', title = 'Histograma y densidad de la variable Scatter_ratio')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=300, by=10))
```



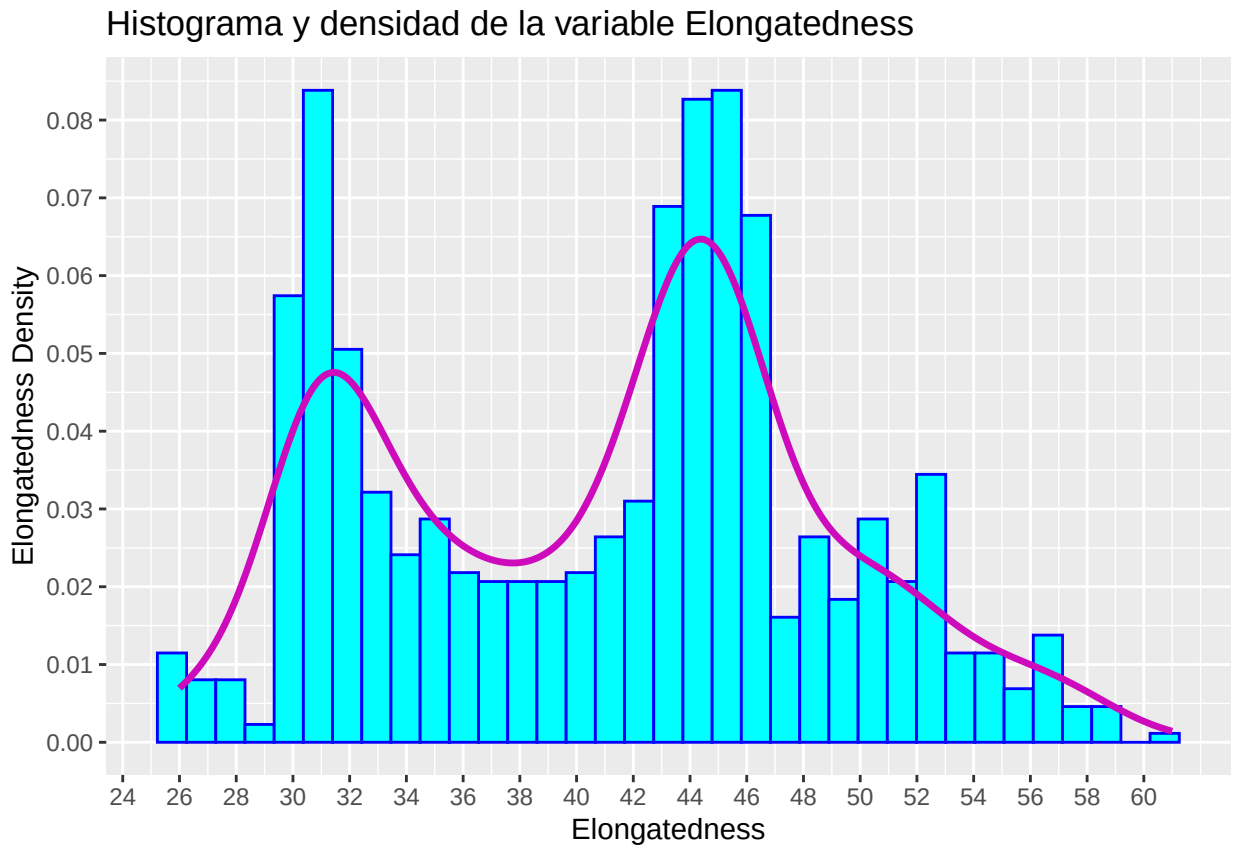

```
ggplot(vehicle, aes(y=Scatter_ratio)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Scatter_ratio', title = 'Diagrama de cajas de la variable Scatter_ratio')+
  scale_y_continuous(breaks = seq(from=0, to=300, by=10))
```



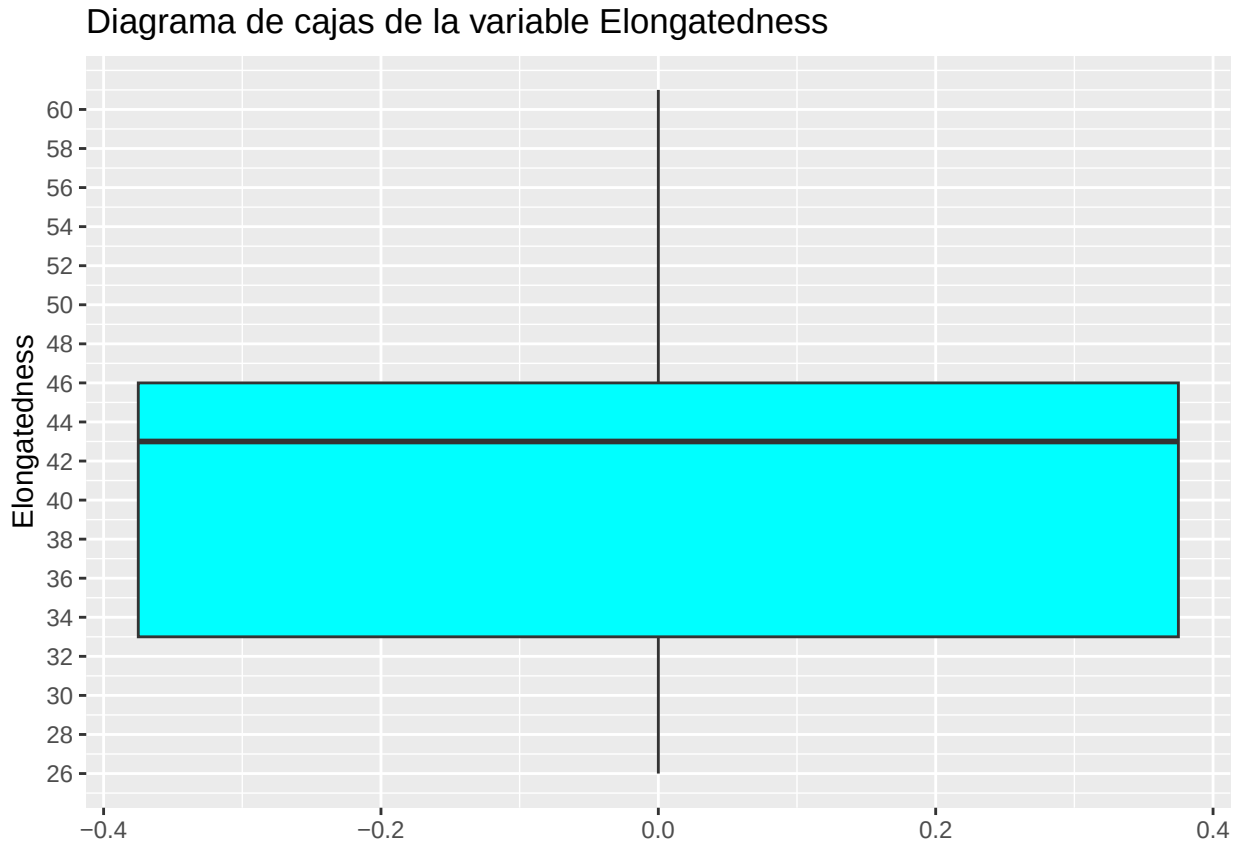
La forma de la distribución parece que se pueda descomponer en la suma de dos diferentes, esto me lleva a preguntarme si este tipo de distribuciones alargadas y con varios picos, jugarán un papel importante para descartar diferentes clases de cara a la clasificación. Podremos ver esto en detalle cuando se analice cada variable con respecto a las distintas categorías de la variable "Class". También cabe destacar que esta distribución no tiene datos atípicos.

Representamos "Elongatedness":

```
# Para Elongatedness
ggplot(vehicle, aes(x=Elongatedness)) +
  geom_histogram(aes(y=stat(density)),bins=35, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Elongatedness Density', title = 'Histograma y densidad de la variable Elongatedness')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=60, by=2))
```



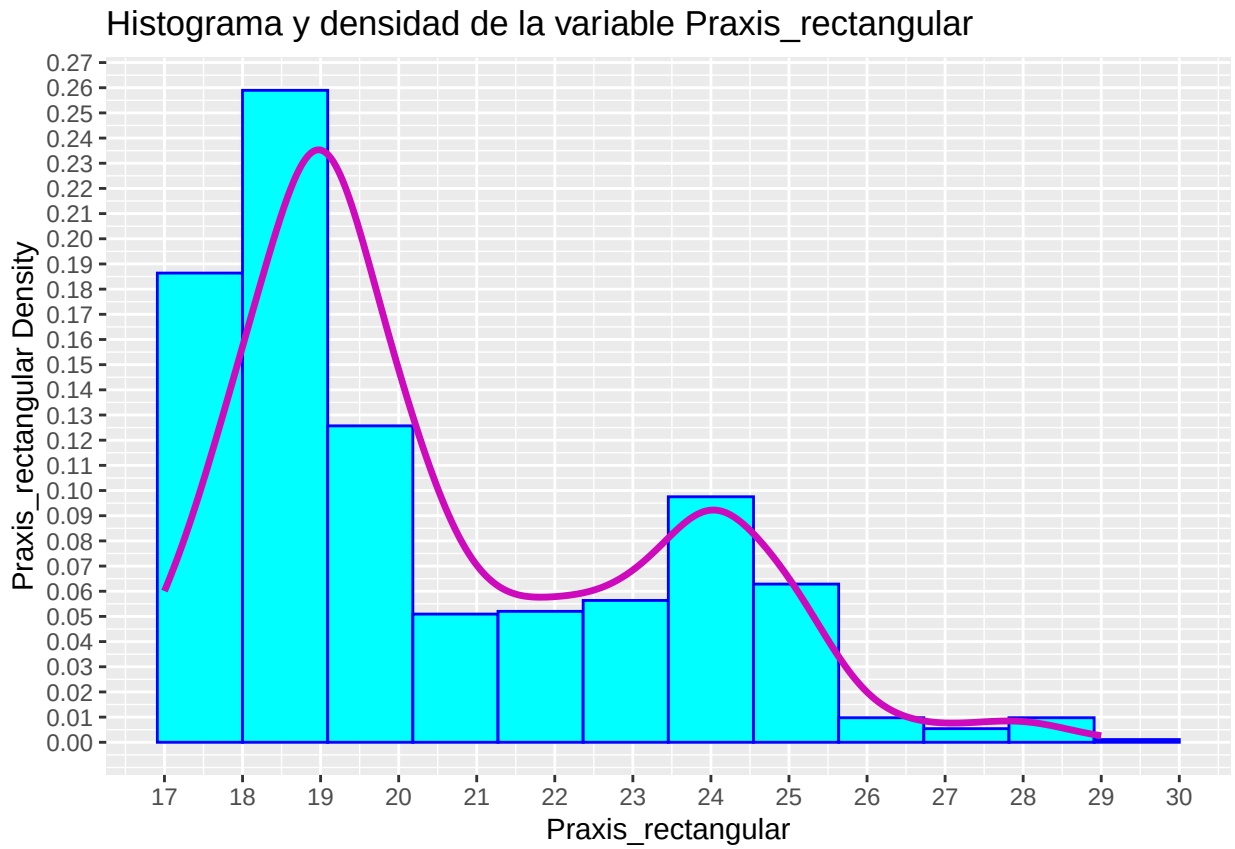

```
ggplot(vehicle, aes(y=Elongatedness)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Elongatedness', title = 'Diagrama de cajas de la variable Elongatedness')+
  scale_y_continuous(breaks = seq(from=0, to=60, by=2))
```



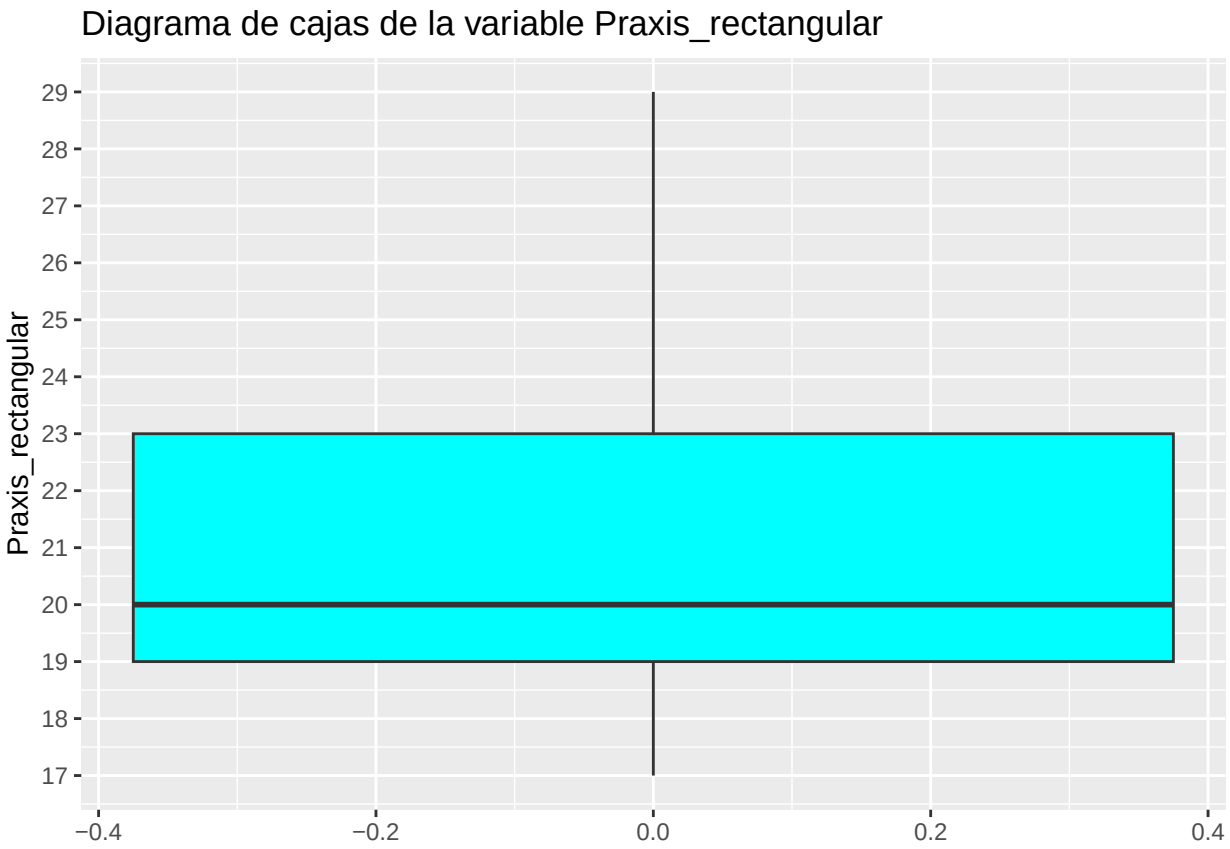
Otra distribución alargada con dos picos, en este caso tampoco tenemos outliers.

Para “Praxis_rectangular”:

```
# Para Praxis_rectangular
ggplot(vehicle, aes(x=Praxis_rectangular)) +
  geom_histogram(aes(y=stat(density)),bins=12, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Praxis_rectangular Density', title = 'Histograma y densidad de la variable Praxis_rectangular
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=30, by=1))
```



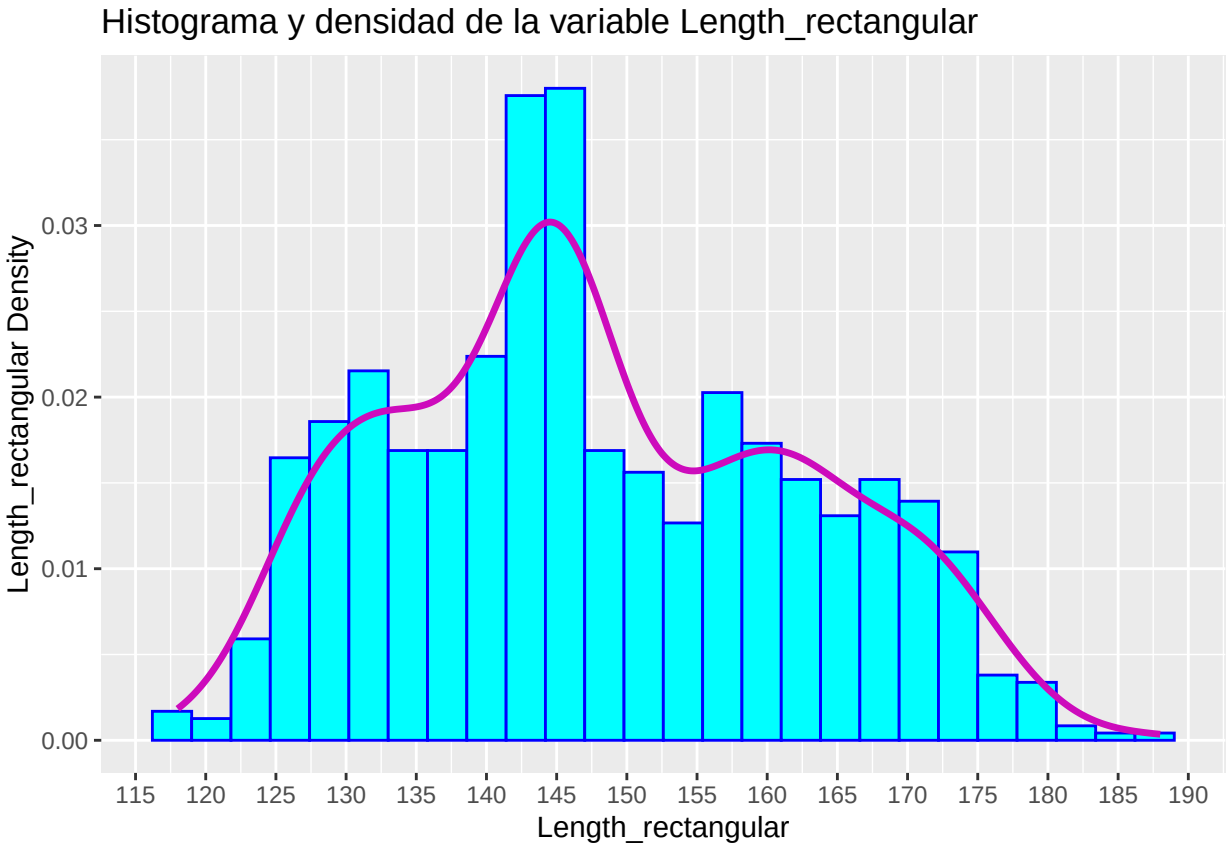

```
ggplot(vehicle, aes(y=Praxis_rectangular)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Praxis_rectangular', title = 'Diagrama de cajas de la variable Praxis_rectangular')+
  scale_y_continuous(breaks = seq(from=0, to=30, by=1))
```



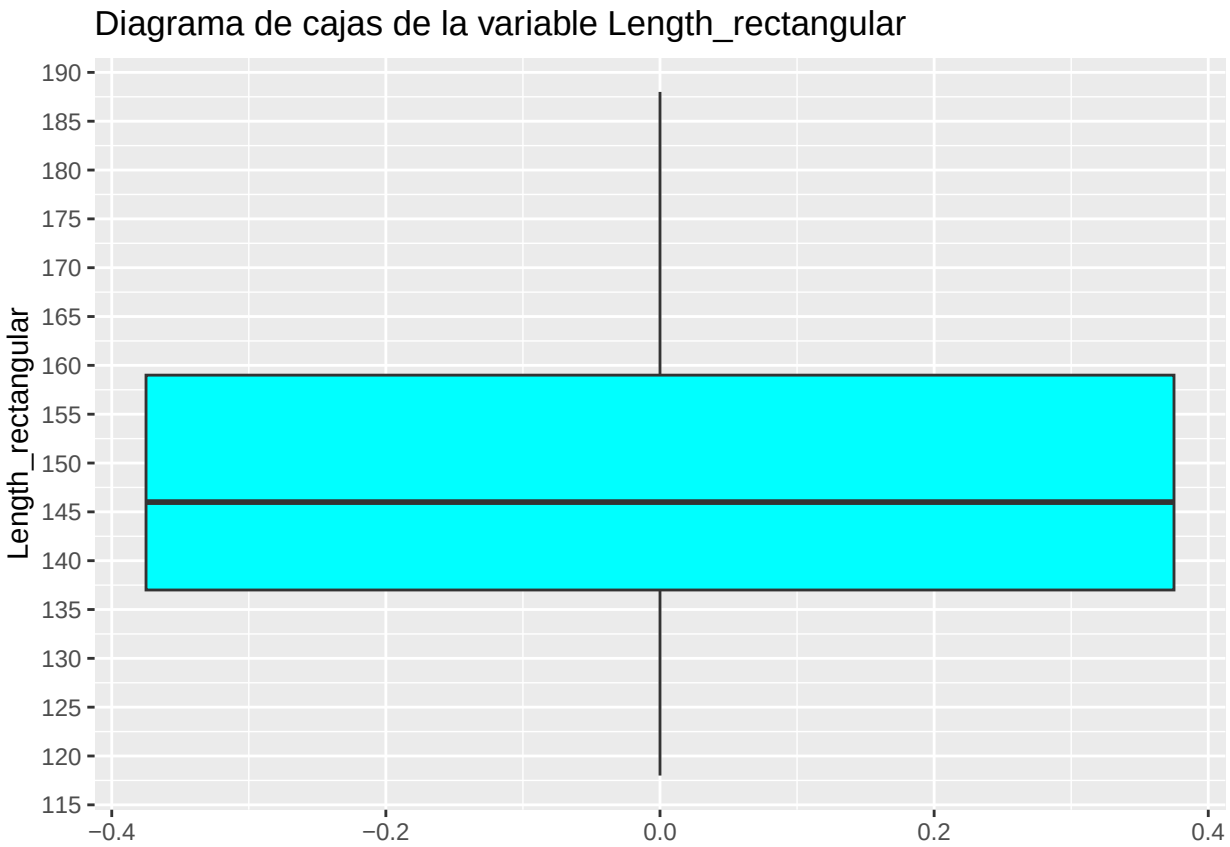
Se ha escogido un número de bins pequeño debido a que esta variable tiene la menor varianza de todas, y por lo tanto los valores que toma están muy próximos. Esta distribución no tiene outliers tampoco. Distribución con dos picos.

En el caso de "Length_rectangular":

```
# Para Length_rectangular
ggplot(vehicle, aes(x=Length_rectangular)) +
  geom_histogram(aes(y=stat(density)),bins=26, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Length_rectangular Density', title = 'Histograma y densidad de la variable Length_rectangular
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=200, by=5))
```



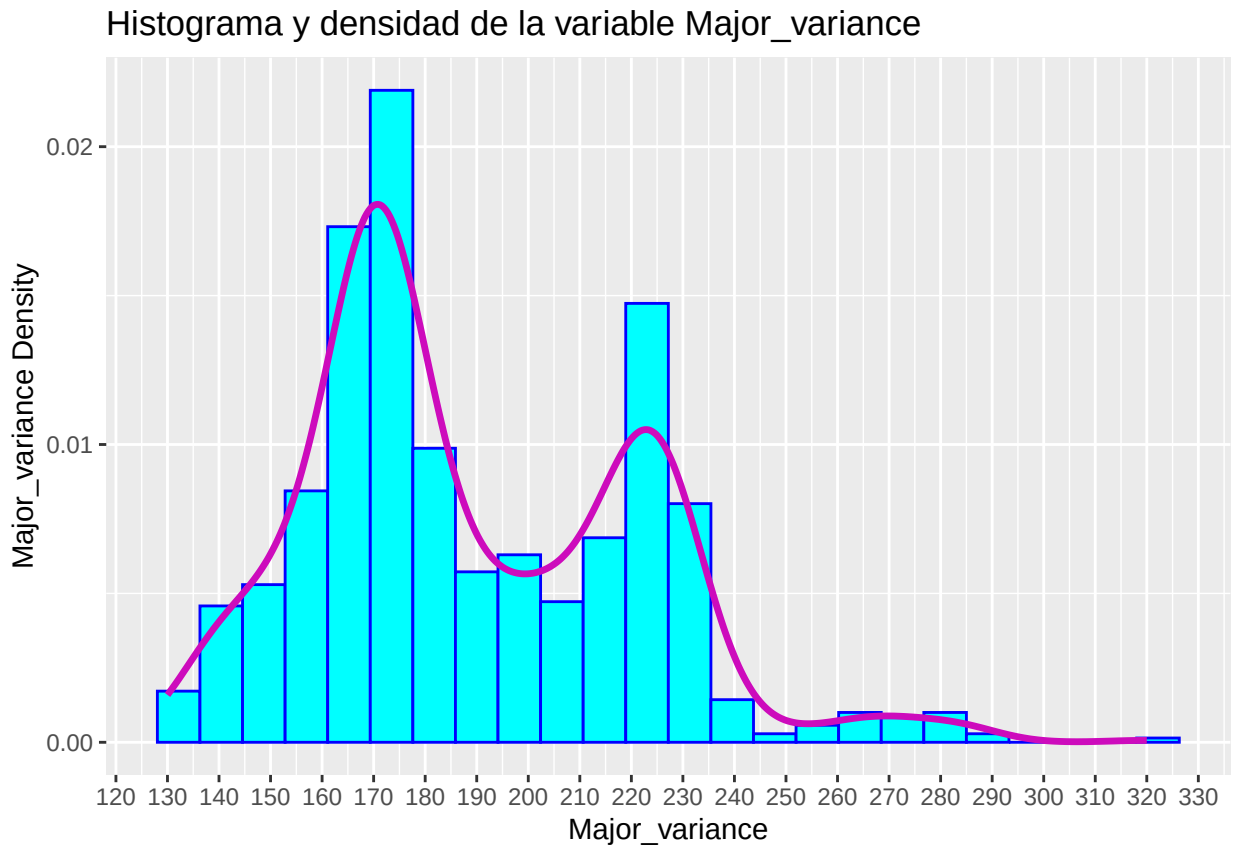

```
ggplot(vehicle, aes(y=Length_rectangular)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Length_rectangular', title = 'Diagrama de cajas de la variable Length_rectangular')+
  scale_y_continuous(breaks = seq(from=0, to=200, by=5))
```



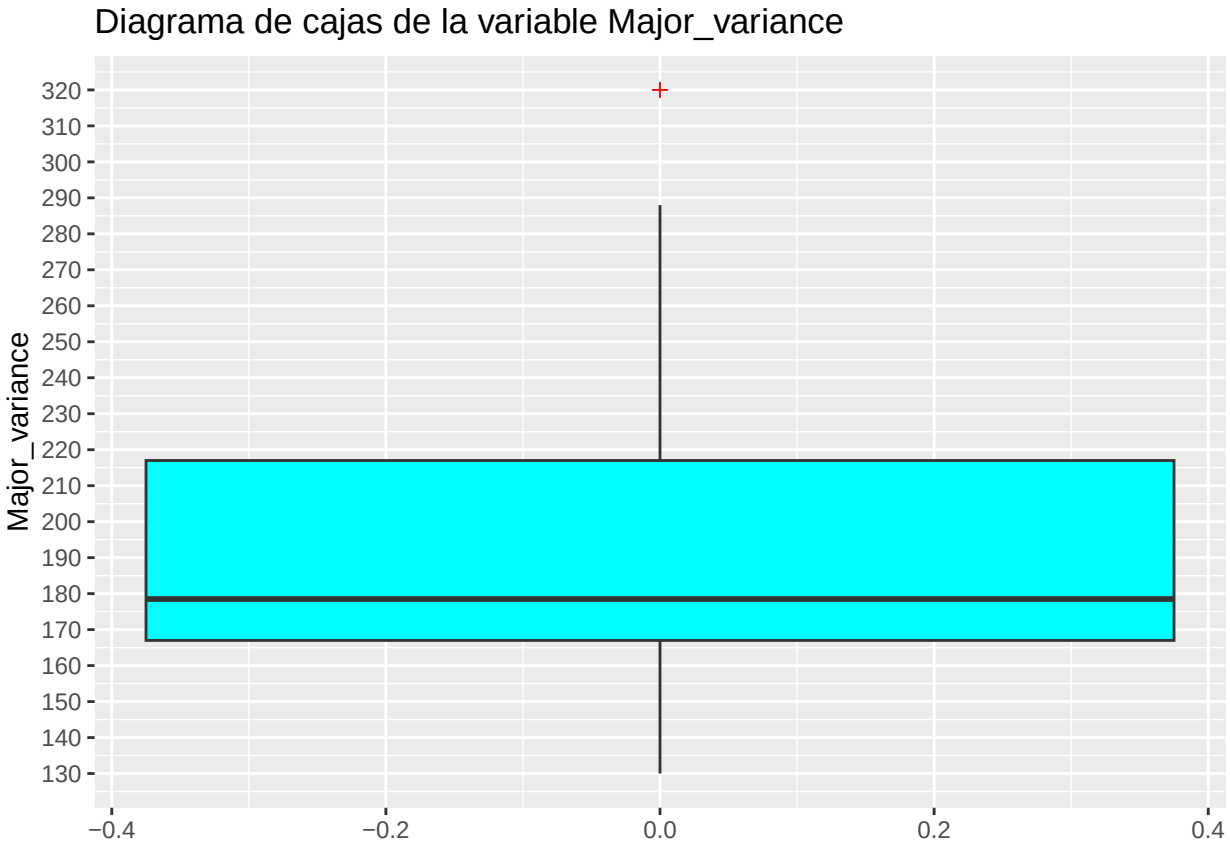
También se trata de una distribución alargada con un pico muy pronunciado y sin outliers.

Para "Major_variance":

```
# Para Major_variance
ggplot(vehicle, aes(x=Major_variance)) +
  geom_histogram(aes(y=stat(density)),bins=24, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Major_variance Density', title = 'Histograma y densidad de la variable Major_variance')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=350, by=10))
```



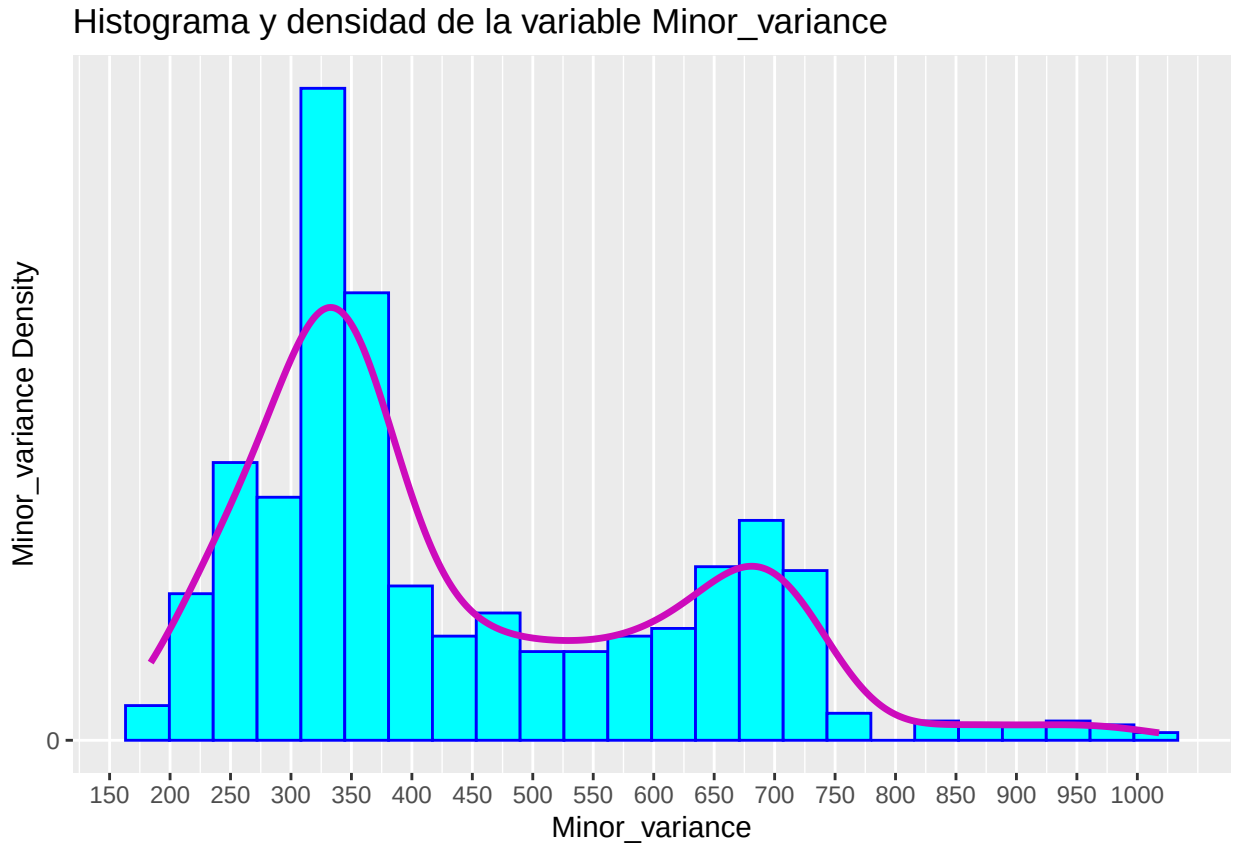

```
ggplot(vehicle, aes(y=Major_variance)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Major_variance', title = 'Diagrama de cajas de la variable Major_variance')+
  scale_y_continuous(breaks = seq(from=0, to=350, by=10))
```



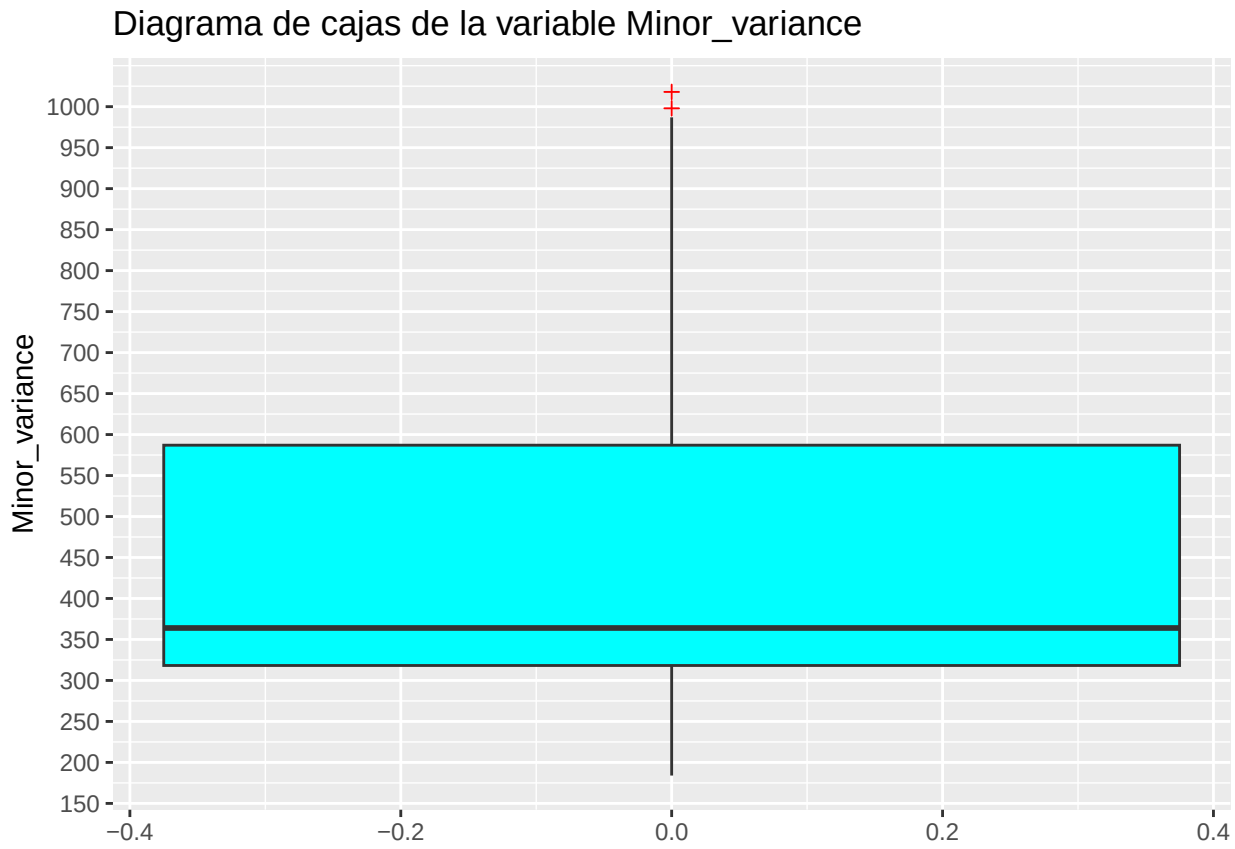
Es una distribución con tres picos y un outlier, el "skewness" es positivo como en el resto de las distribuciones vistas hasta ahora.

Estudiando "Minor_variance":

```
# Para Minor_variance
ggplot(vehicle, aes(x=Minor_variance)) +
  geom_histogram(aes(y=stat(density)),bins=24, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Minor_variance Density', title = 'Histograma y densidad de la variable Minor_variance')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=1000, by=50))
```



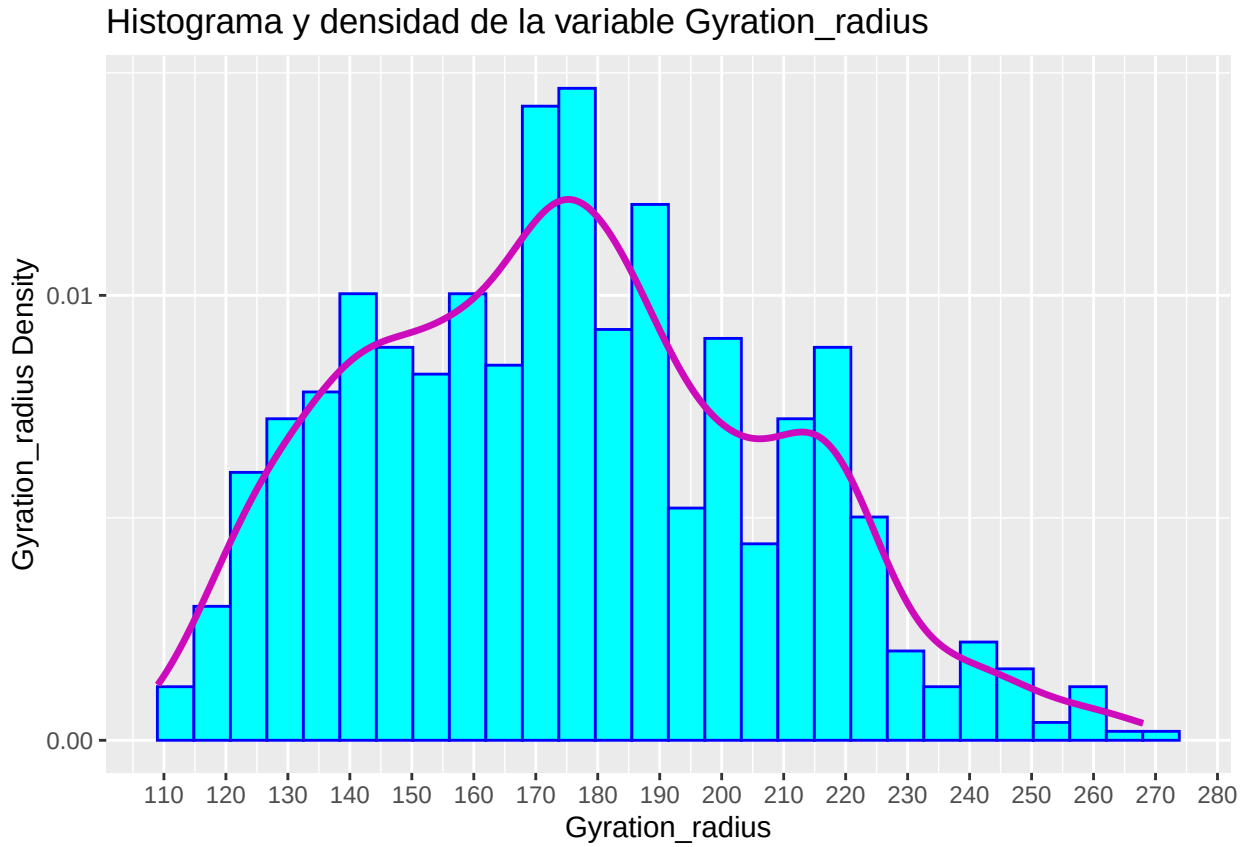

```
ggplot(vehicle, aes(y=Minor_variance)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Minor_variance', title = 'Diagrama de cajas de la variable Minor_variance')+
  scale_y_continuous(breaks = seq(from=0, to=1000, by=50))
```



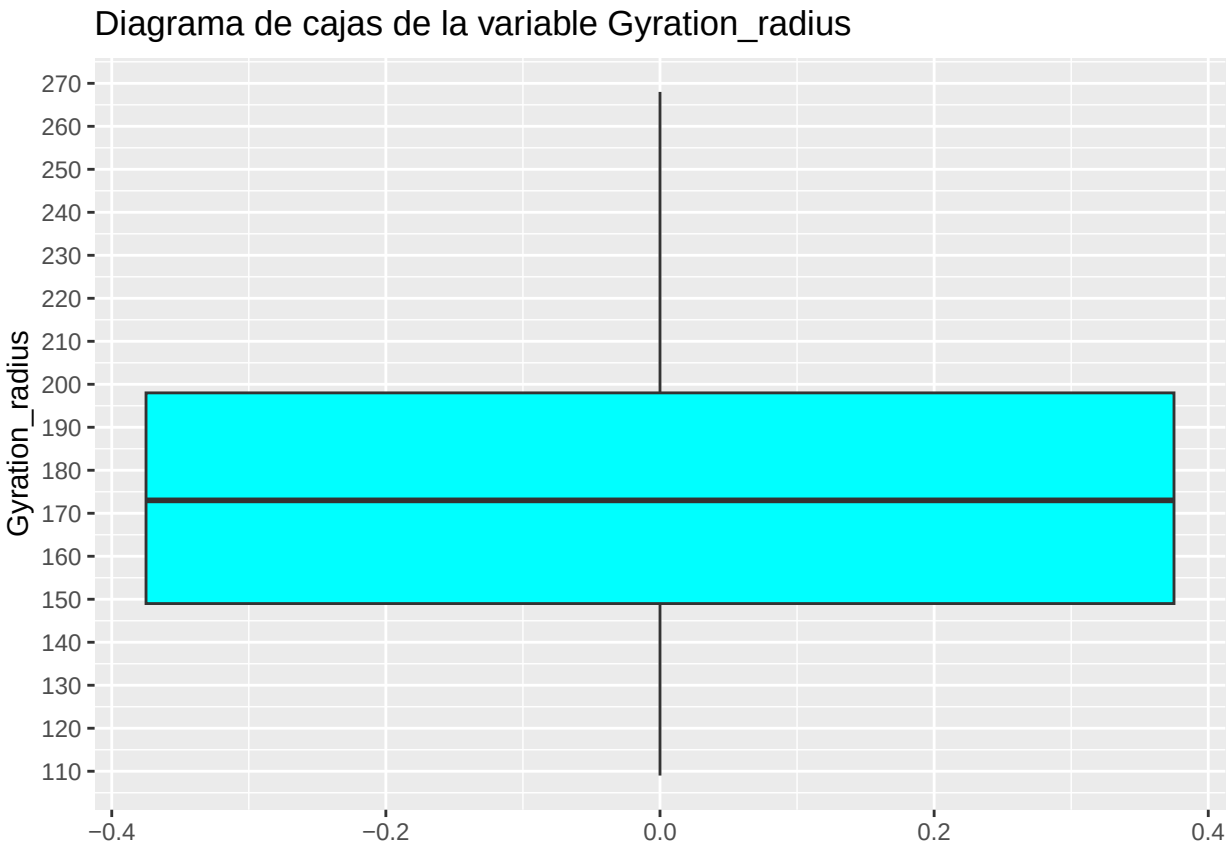
Tiene una forma muy parecida a la distribución de la variable "Major_variance" solo que más alargada (es la que tiene una mayor varianza) y por lo tanto los picos son menos pronunciados. En este caso se tienen dos outliers muy cerca del conjunto de datos.

Con "Gyration_radius":

```
# Para Gyration_radius
ggplot(vehicle, aes(x=Gyration_radius)) +
  geom_histogram(aes(y=stat(density)),bins=28, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Gyration_radius Density', title = 'Histograma y densidad de la variable Gyration_radius')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=300, by=10))
```



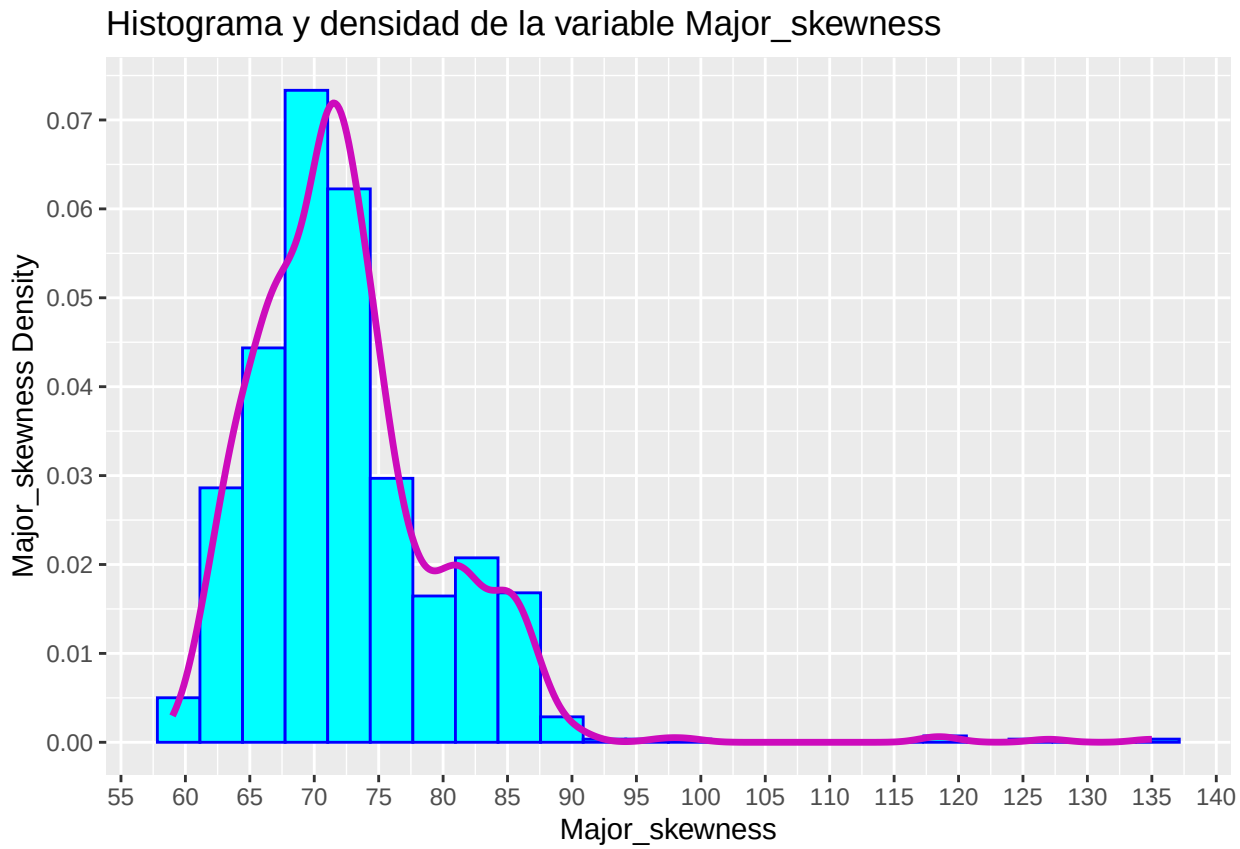

```
ggplot(vehicle, aes(y=Gyration_radius)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Gyration_radius', title = 'Diagrama de cajas de la variable Gyration_radius')+
  scale_y_continuous(breaks = seq(from=0, to=300, by=10))
```



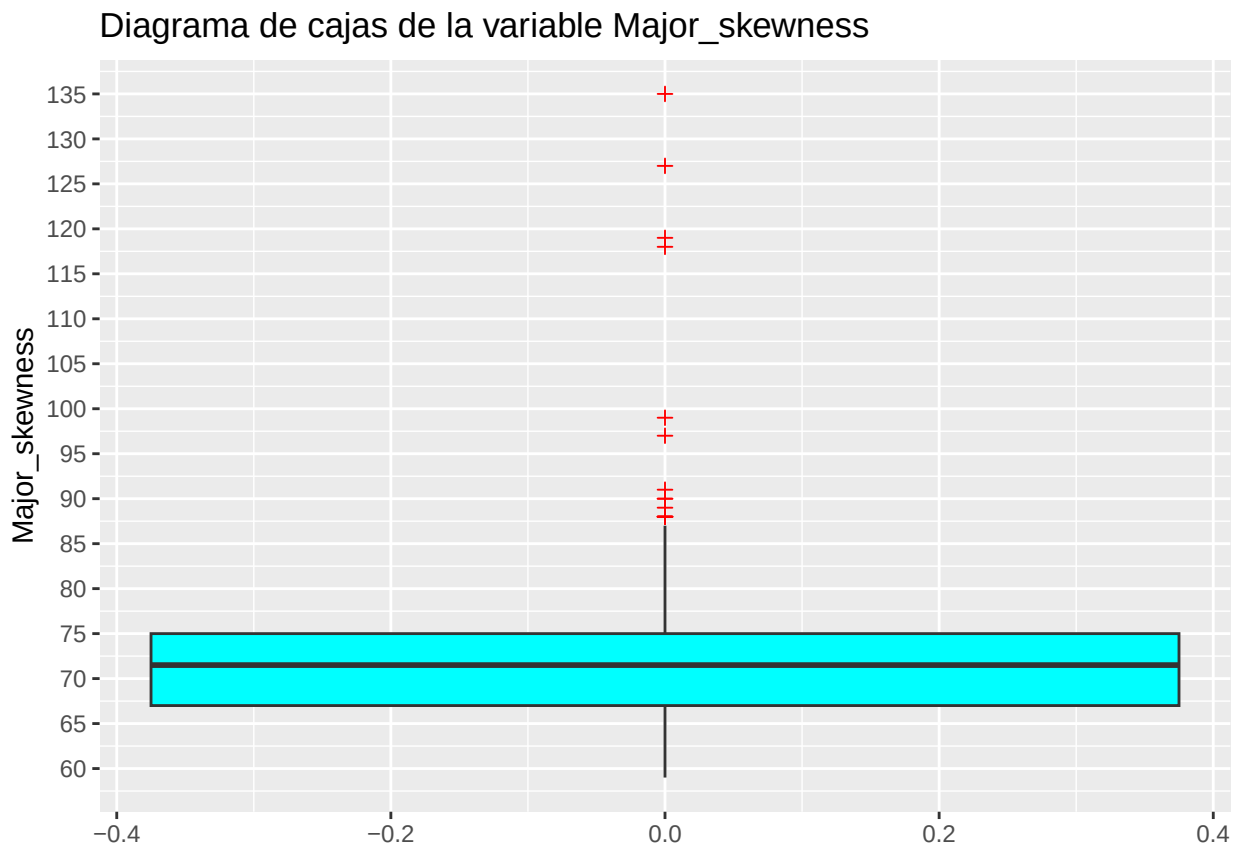
Obtenemos otra distribución alargada con varios puntos de inflexión. Esta no tiene outliers tampoco.

Para "Major_skewness"

```
# Para Major_skewness
ggplot(vehicle, aes(x=Major_skewness)) +
  geom_histogram(aes(y=stat(density)),bins=24, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Major_skewness Density', title = 'Histograma y densidad de la variable Major_skewness')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=140, by=5))
```



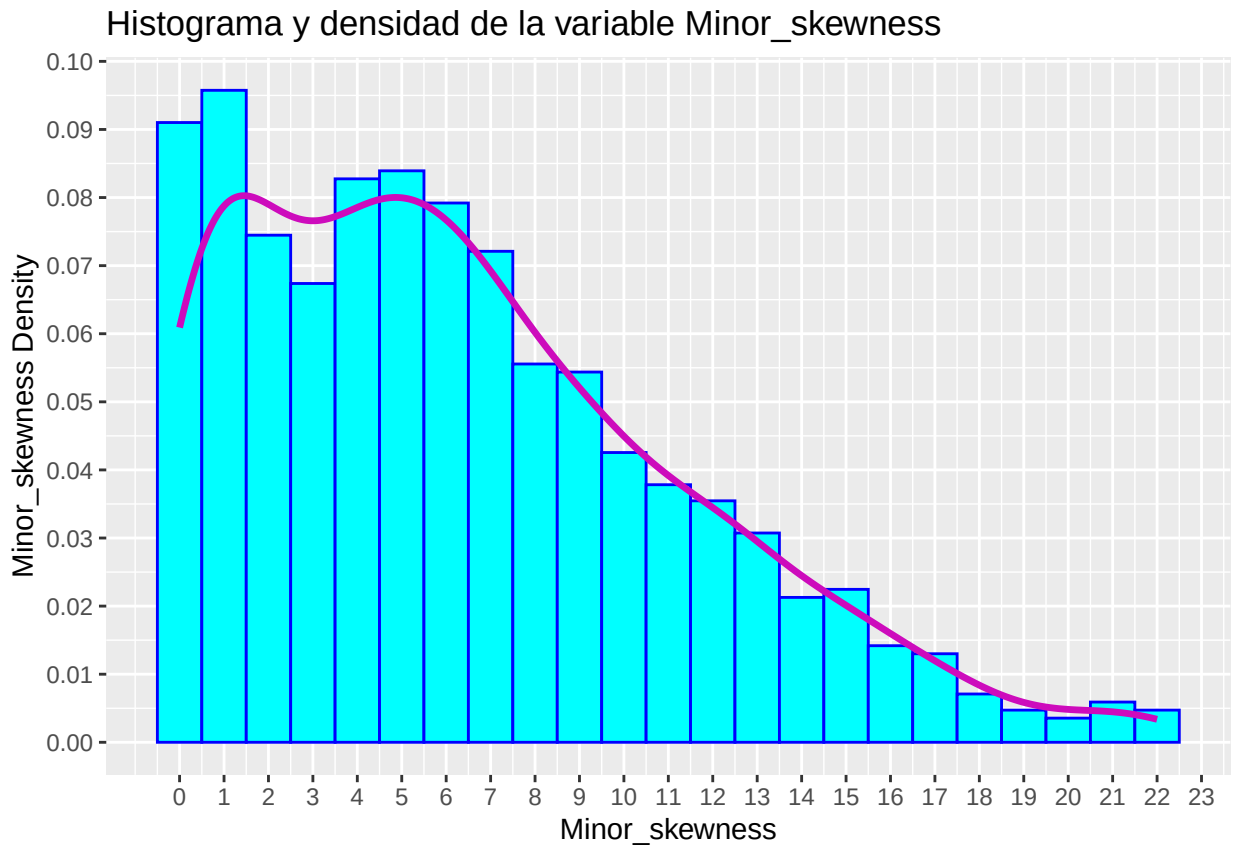

```
ggplot(vehicle, aes(y=Major_skewness)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Major_skewness', title = 'Diagrama de cajas de la variable Major_skewness')+
  scale_y_continuous(breaks = seq(from=0, to=140, by=5))
```



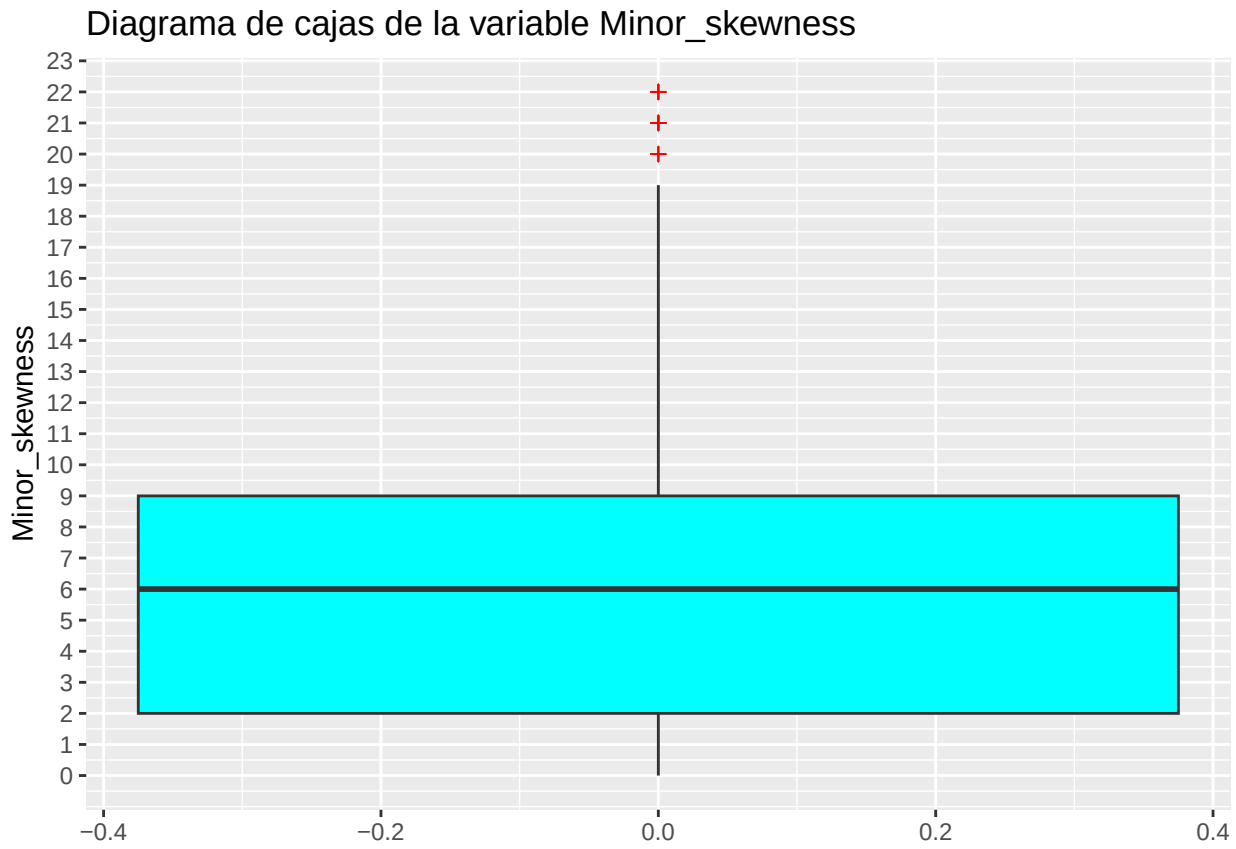
Tiene una distribución bastante puntiaguda con grandes pendientes. También consta de varios outliers.

En el caso de "Minor_skewness":

```
# Para Minor_skewness
ggplot(vehicle, aes(x=Minor_skewness)) +
  geom_histogram(aes(y=stat(density)),bins=23, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Minor_skewness Density', title = 'Histograma y densidad de la variable Minor_skewness')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=25, by=1))
```



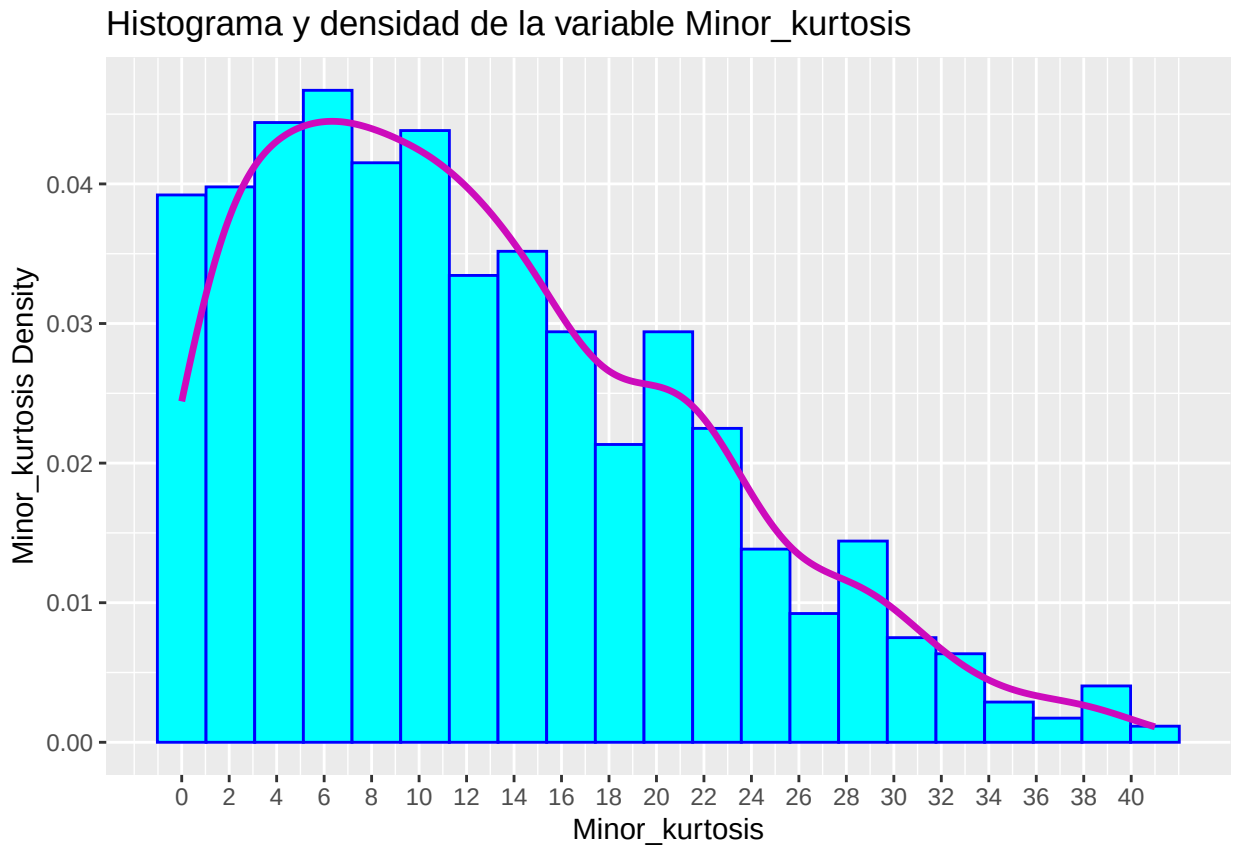

```
ggplot(vehicle, aes(y=Minor_skewness)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Minor_skewness', title = 'Diagrama de cajas de la variable Minor_skewness')+
  scale_y_continuous(breaks = seq(from=0, to=25, by=1))
```



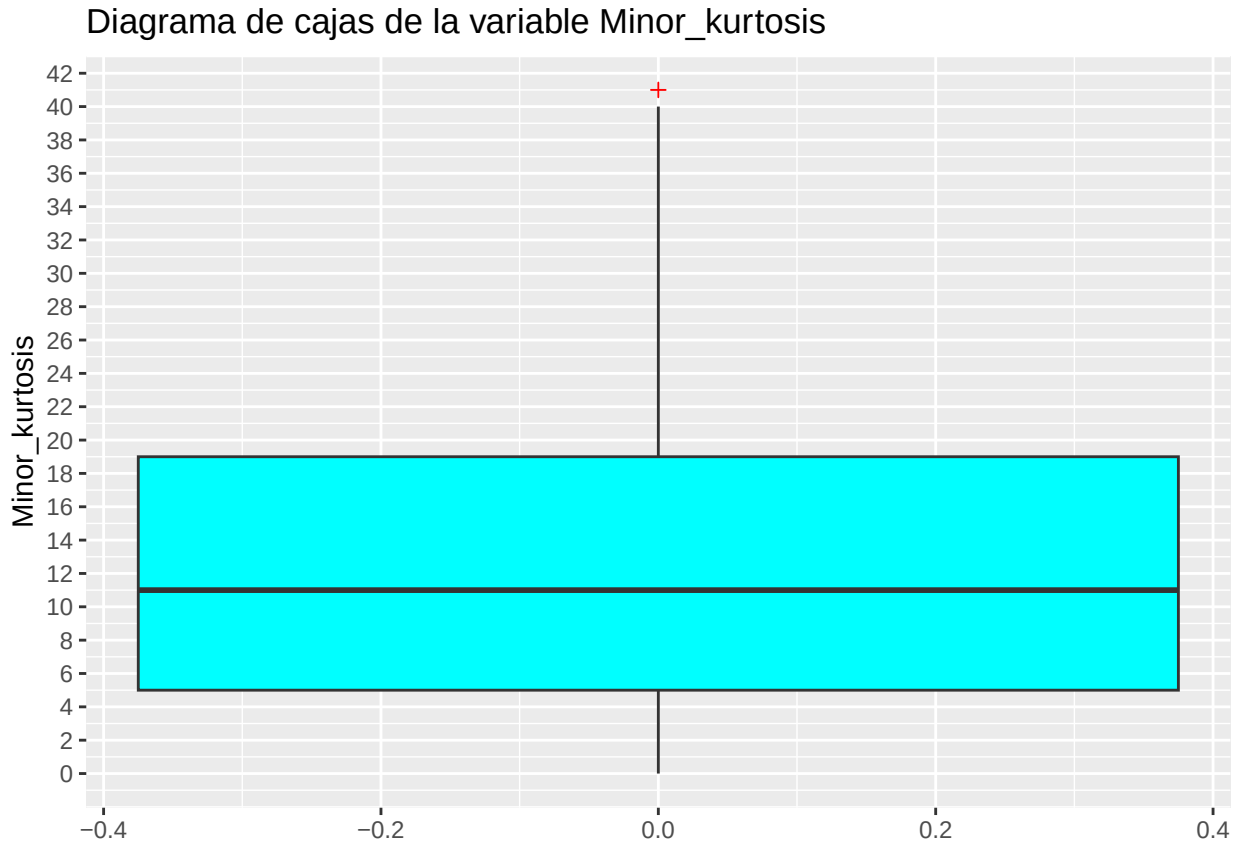
Es una distribución muy diferente a la de "major_skewness", evidentemente asimétrica con un "skewness" positivo aunque el coeficiente calculado no sea muy alto, y con pendientes suaves. Consta de tres outliers con valores poco superiores a los de la distribución.

Para "Minor_kurtosis":

```
# Para Minor_kurtosis
ggplot(vehicle, aes(x=Minor_kurtosis)) +
  geom_histogram(aes(y=stat(density)),bins=21, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Minor_kurtosis Density', title = 'Histograma y densidad de la variable Minor_kurtosis')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=40, by=2))
```



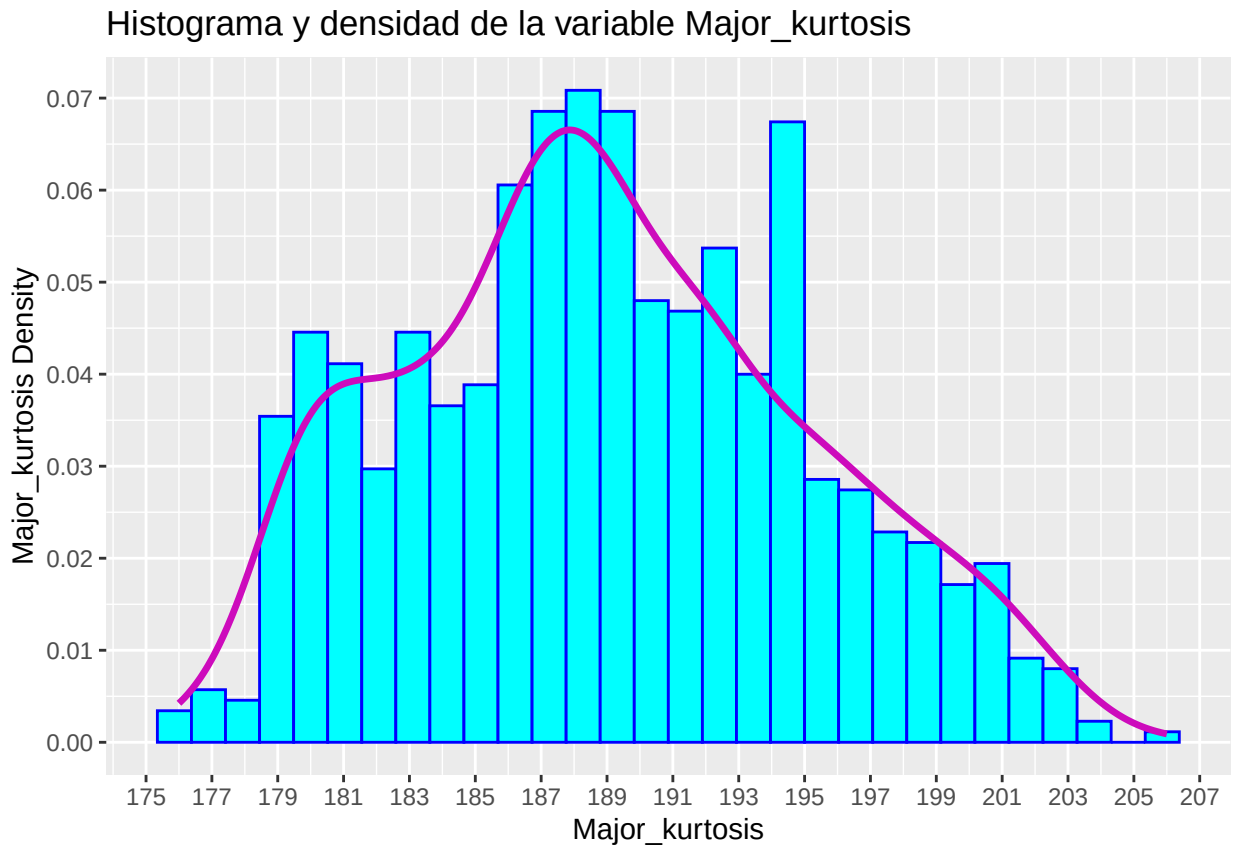

```
ggplot(vehicle, aes(y=Minor_kurtosis)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Minor_kurtosis', title = 'Diagrama de cajas de la variable Minor_kurtosis')+
  scale_y_continuous(breaks = seq(from=0, to=48, by=2))
```



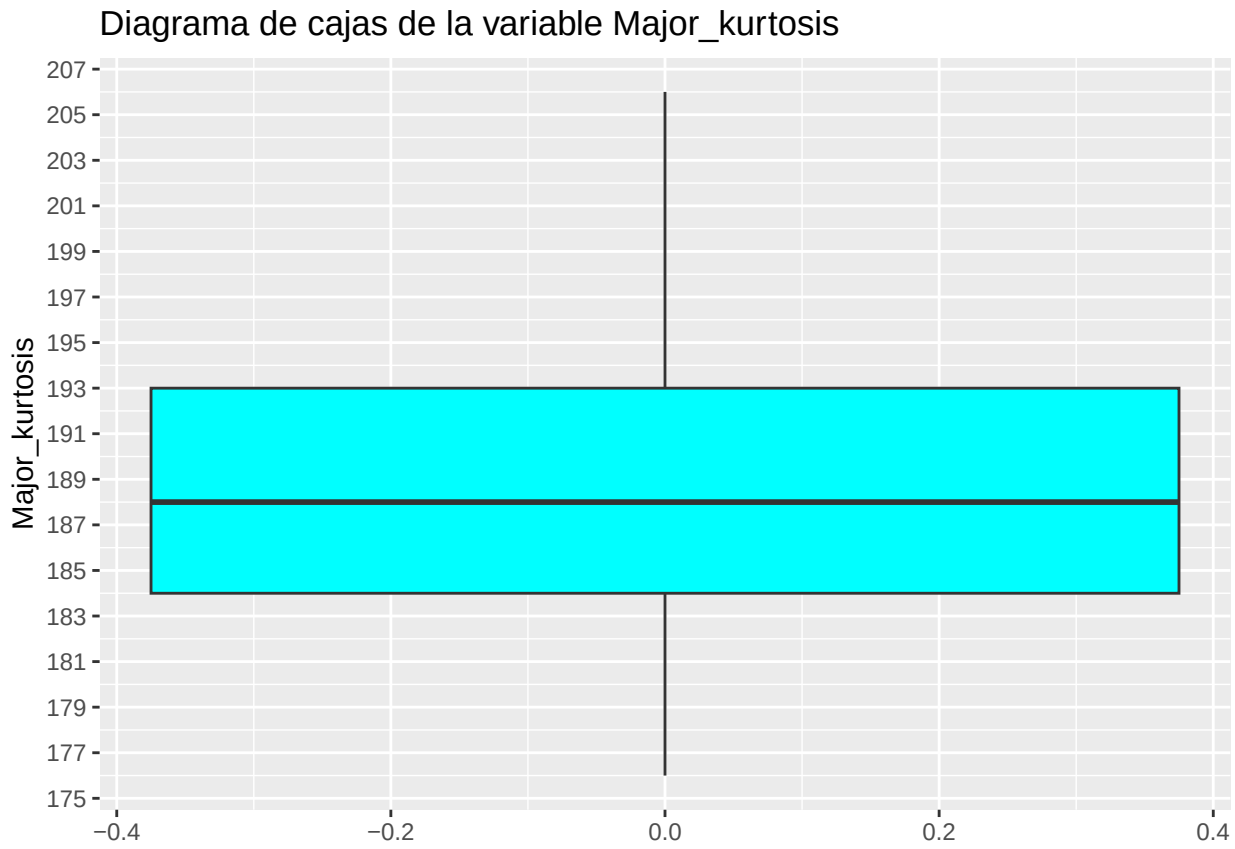
Obtenemos una distribución algo parecida a la anterior, con un outlier de valor no muy extremo.

Con "Major_kurtosis" tenemos:

```
# Para Major_kurtosis
ggplot(vehicle, aes(x=Major_kurtosis)) +
  geom_histogram(aes(y=stat(density)),bins=30, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Major_kurtosis Density', title = 'Histograma y densidad de la variable Major_kurtosis')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=175, to=210, by=2))
```



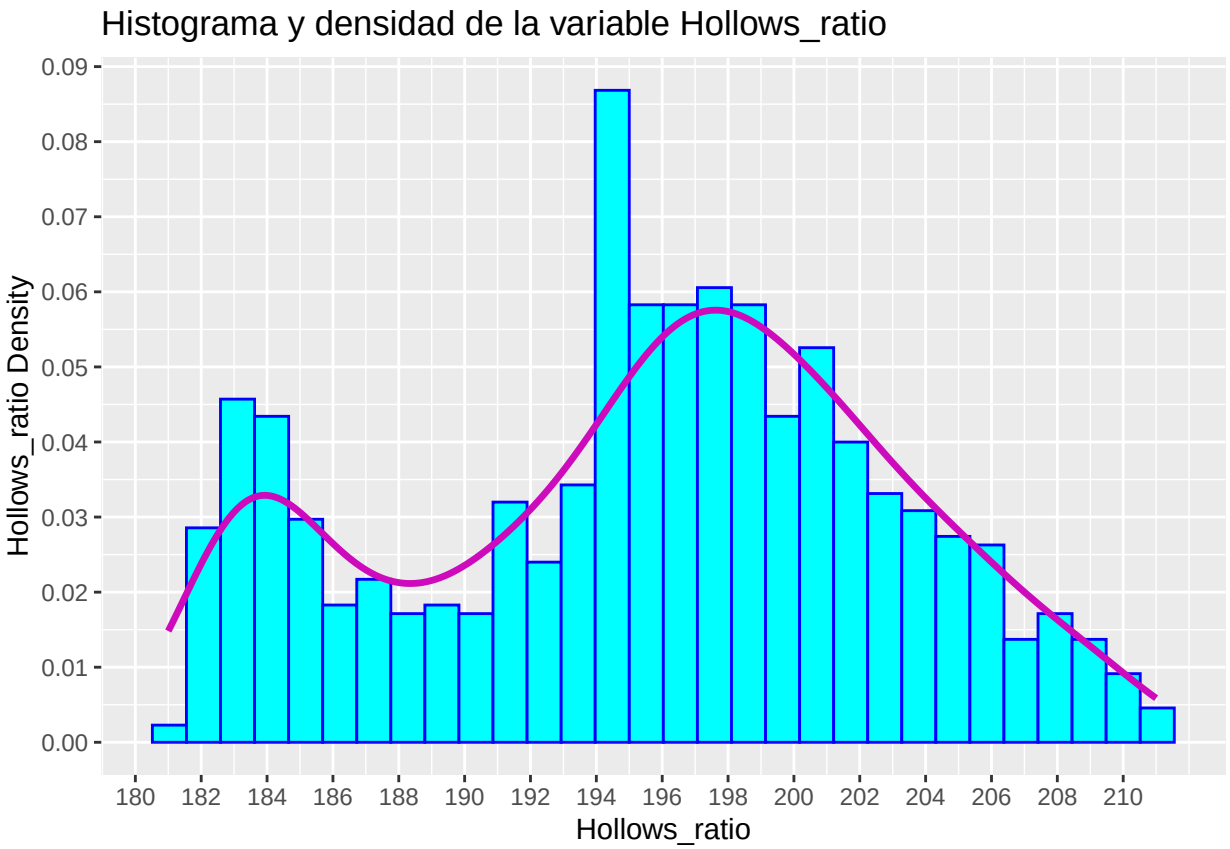

```
ggplot(vehicle, aes(y=Major_kurtosis)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Major_kurtosis', title = 'Diagrama de cajas de la variable Major_kurtosis')+
  scale_y_continuous(breaks = seq(from=175, to=210, by=2))
```



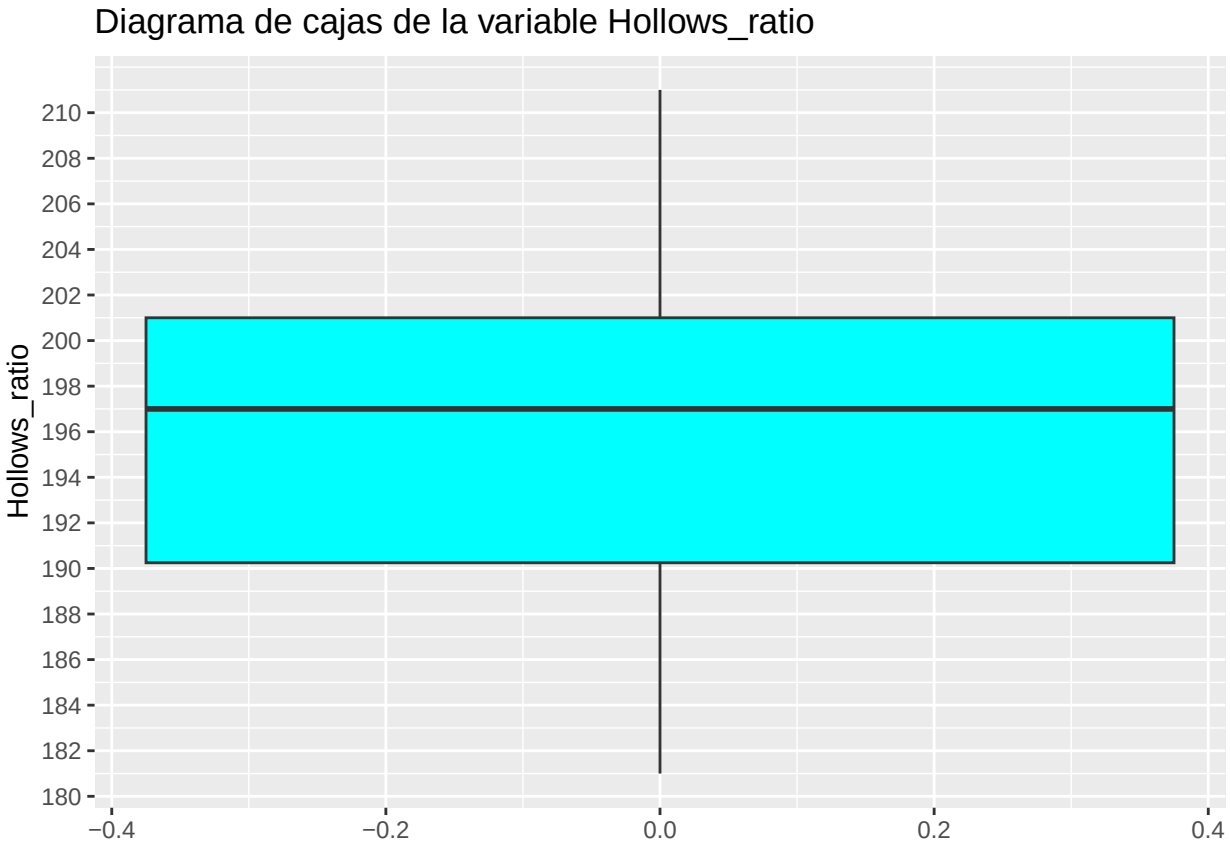
Distribución larga que parece que se puede descomponer en varias, sin outliers.

Finalmente, para "Hollows_ratio":

```
# Para Hollows_ratio
ggplot(vehicle, aes(x=Hollows_ratio)) +
  geom_histogram(aes(y=stat(density)),bins=30, color='Blue',fill='Cyan')+
  geom_density(lwd = 1.2,linetype = 1,colour = 6)+
  labs(y='Hollows_ratio Density', title = 'Histograma y densidad de la variable Hollows_ratio')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=180, to=210, by=2))
```




```
ggplot(vehicle, aes(y=Hollows_ratio)) +
  geom_boxplot(fill='Cyan',outlier.colour = 'red', outlier.shape = 3)+
  labs(y='Hollows_ratio', title = 'Diagrama de cajas de la variable Hollows_ratio')+
  scale_y_continuous(breaks = seq(from=180, to=210, by=2))
```



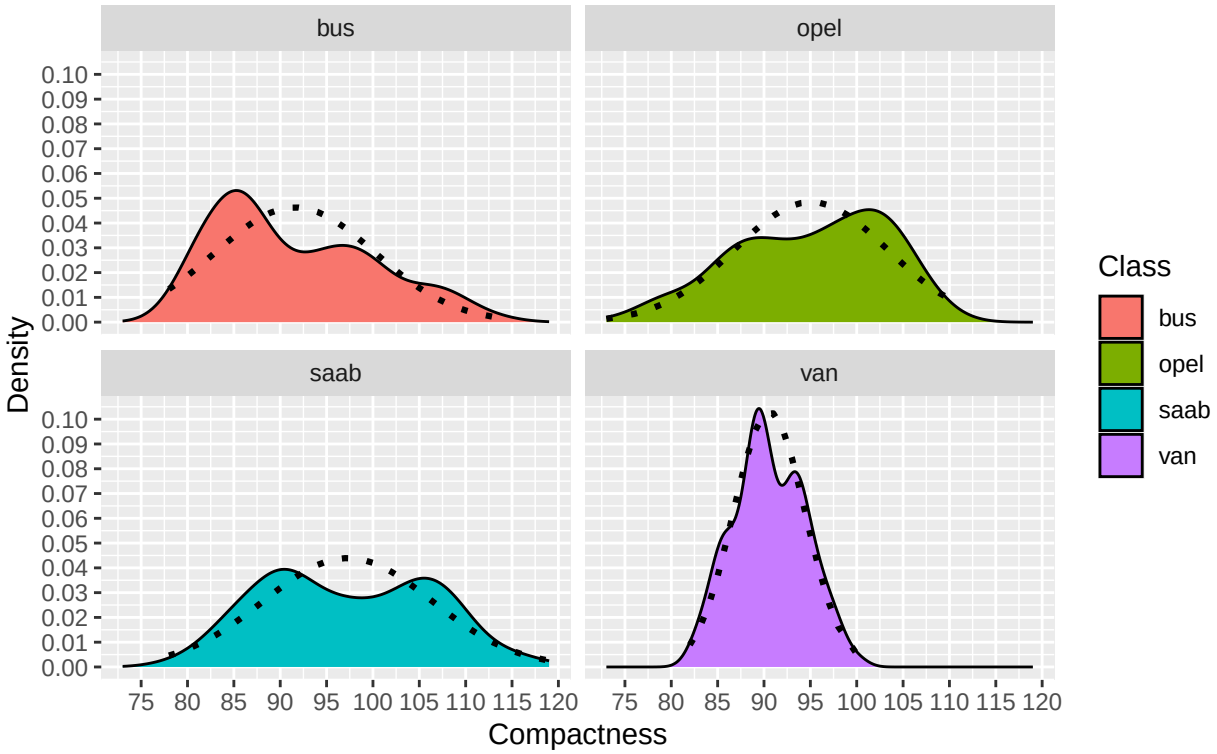
Se obtiene una distribución con dos picos y sin outliers. Para esta variable era la única en la que se había obtenido un coeficiente de "skewness" negativo.

Gráficas bivariables entre los atributos y la variable de salida

Puesto que la salida es categórica, se ha optado por descomponer las distribuciones por clase para cada par de variable de entrada-variable de salida. El objetivo de la descomposición es ver las distintas distribuciones que podrían conformar las distribuciones que hemos estudiado en las gráficas univariantes de cada atributo. También se ha representado la distribución normal superpuesta sobre cada distribución, ya que esto nos sirve para estudiar gráficamente la suposición de normalidad para los algortitmos de clasificación LDA y QDA.

Empezamos por "Compactnes":

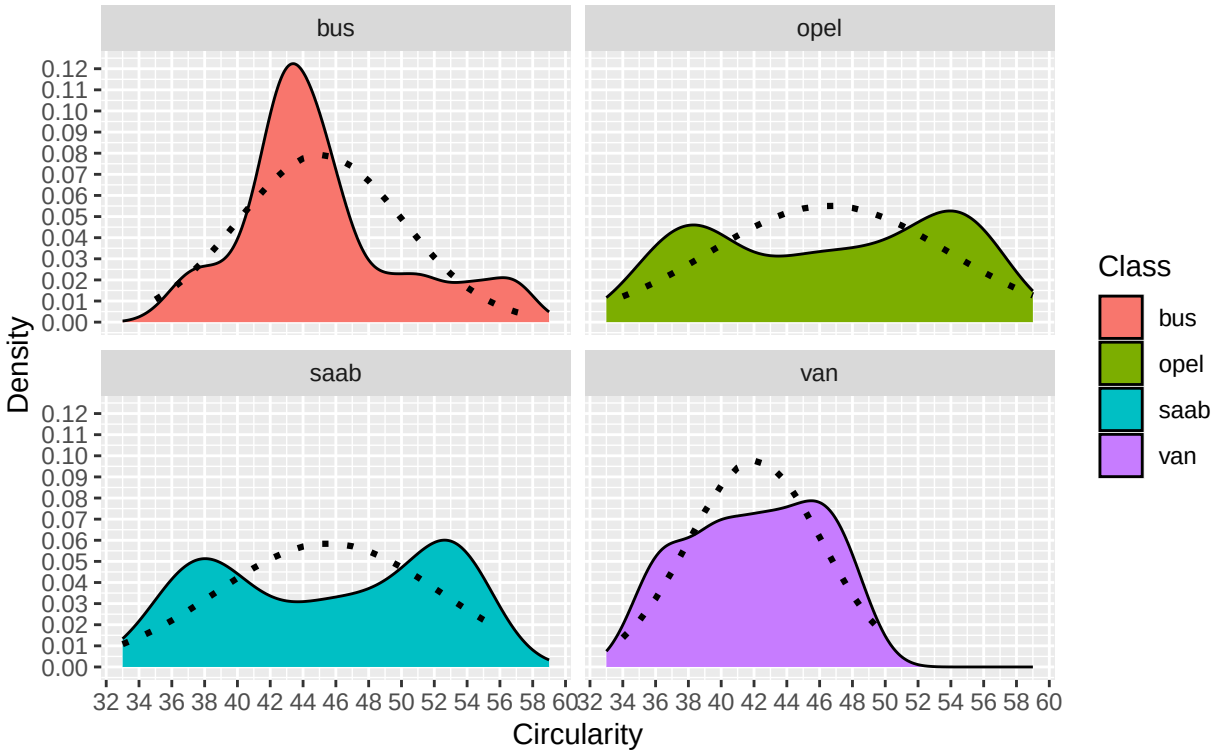
```
# Compactness-Class

vehicle %>%
ggplot(aes(x=Compactness, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Compactness,
                          mean = tapply(Compactness, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Compactness, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Compactness',y='Density',
       title = 'Distribuciones por clase de la variable Compactness',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=120, by=5))
```



Obtenemos las disntintas distribuciones de la variable por clase. Como podemos apreciar la línea negra discontinua correspondería con lo que sería una normal de la misma media y varianza de cada distribución. La que más se podría acercar sería la distribución de la categoría "van". Más adelante comprobaremos con tests estadísticos la normalidad de cada categoría por variables.

Para el par "Circularity-Class":

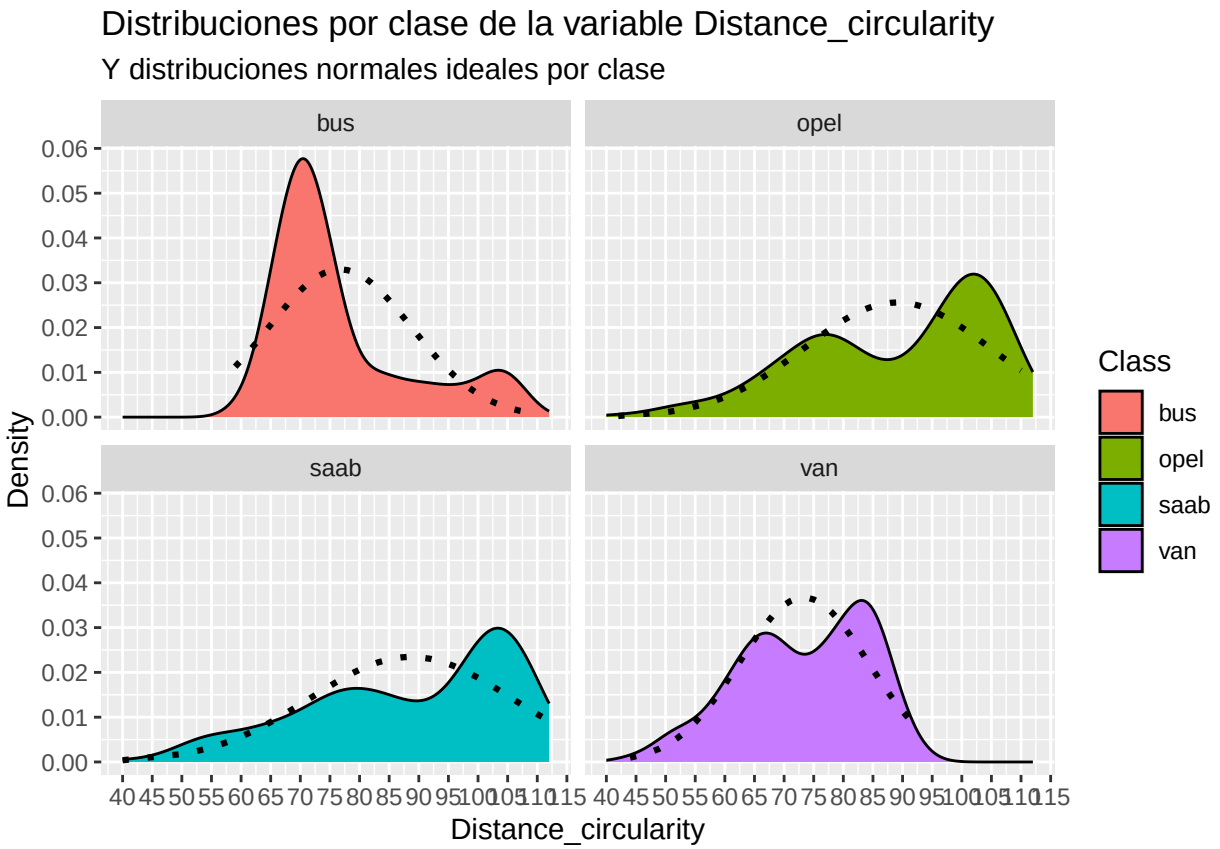
```
# Circularity-Class

vehicle %>%
  ggplot(aes(x=Circularity, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Circularity,
                          mean = tapply(Circularity, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Circularity, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Circularity',y='Density',
       title = 'Distribuciones por clase de la variable Circularity',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=60, by=2))
```



Para esta medida, hay diferencias evidentes entre las categorías "bus" y "van" pero las del "opel" y "saab" son casi idénticas. Esto se debe a que por la silueta, es más sencillo distinguir un autobus, una furgoneta o un coche. Pero cuando se tienen dos coches con una silueta más parecida presenta una mayor dificultad.

Con el par "Distance_circularity-Class" tenemos:

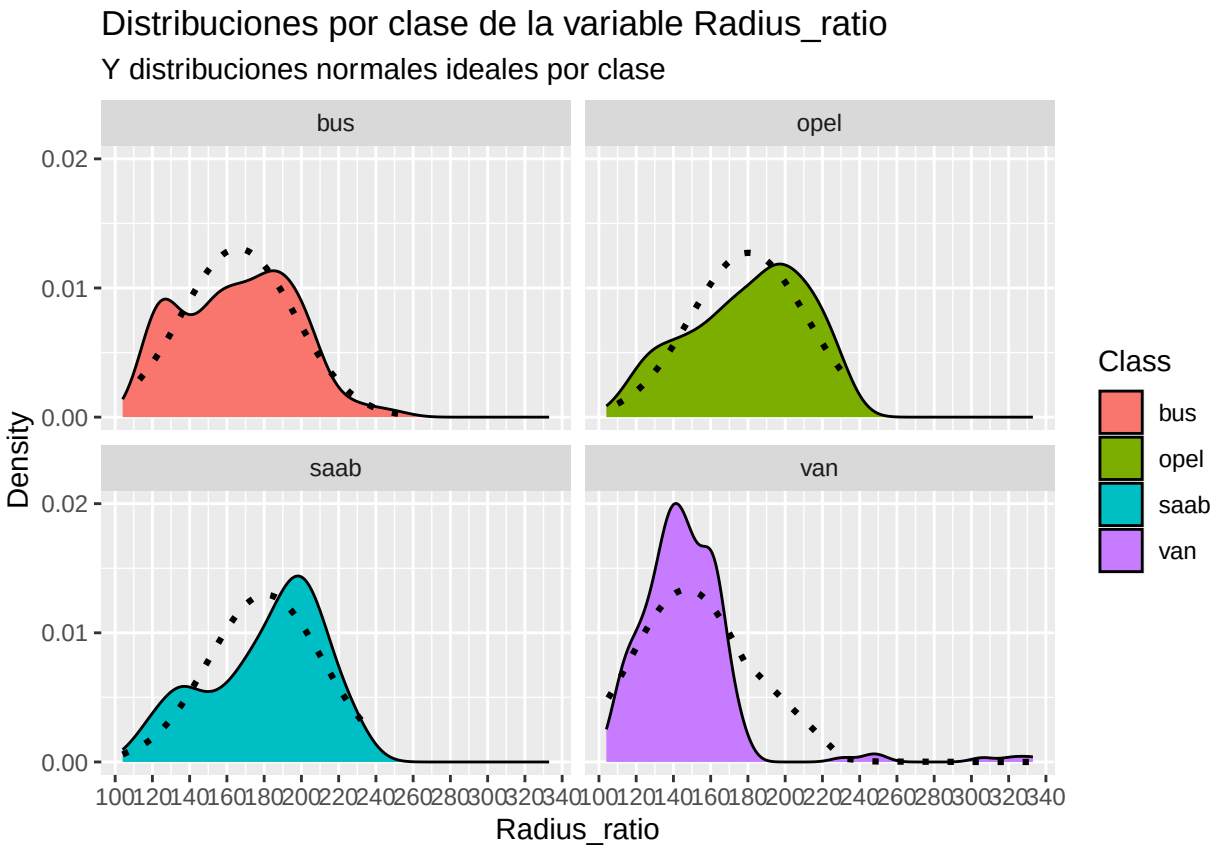
```
# Distance_circularity-Class

vehicle %>%
  ggplot(aes(x=Distance_circularity, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Distance_circularity,
                          mean = tapply(Distance_circularity, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Distance_circularity, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Distance_circularity',y='Density',
       title = 'Distribuciones por clase de la variable Distance_circularity',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=120, by=5))
```



En este caso la distribución de "bus" es la que más dista en parecido de las otras. La distribución de la categoría "van" tiene una forma parecida pero con menor dispersión y valores más concentrados que las otras dos, que son muy parecidas entre ellas.

Para el par "Radius_ratio-Class":

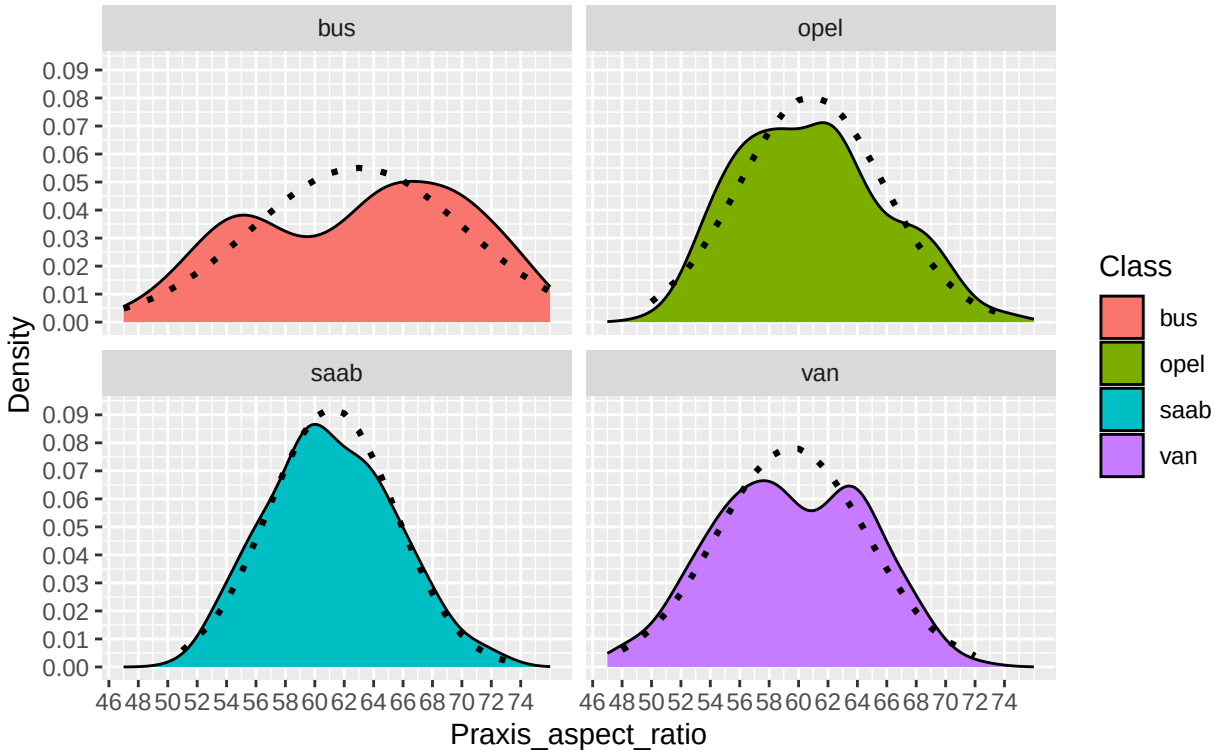
```
# Radius_ratio-Class

vehicle %>%
  ggplot(aes(x=Radius_ratio, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Radius_ratio,
                          mean = tapply(Radius_ratio, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Radius_ratio, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Radius_ratio',y='Density',
       title = 'Distribuciones por clase de la variable Radius_ratio',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=350, by=20))
```



En esta variable las distribuciones son algo más diferentes entre ellas, siendo la más diferente la categoría "van". Además como vimos anteriormente, esta variable presentaba una distribución con outliers bastante extremos, en está gráfica podemos comprobar que estos outliers pertenecen a la categoría "van". Cabe destacar que por la forma que toma la distribución normal en esta categoría, puede ser que tenga dificultad en pasar el test de normalidad debido a estos outliers.

En el caso de "Praxis_aspect_ratio-Class":

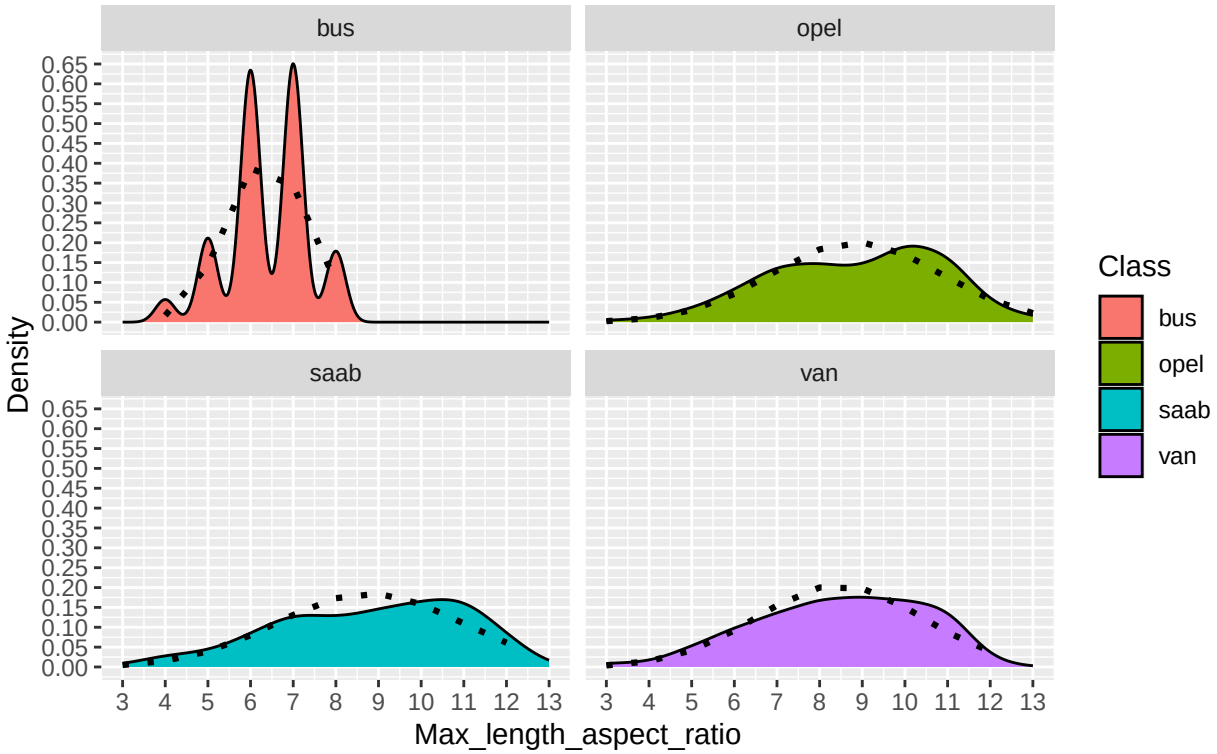
```
# Praxis_aspect_ratio-Class

vehicle[-Praxis_aspect_ratio.outliers,] %>%
  ggplot(aes(x=Praxis_aspect_ratio, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Praxis_aspect_ratio,
                          mean = tapply(Praxis_aspect_ratio, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Praxis_aspect_ratio, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Praxis_aspect_ratio',y='Density',
       title = 'Distribuciones por clase de la variable Praxis_aspect_ratio',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=75, by=2))
```



Como vimos en el análisis de cada distribución, para esta variable se tenía una distribución con un pico muy pronunciado, esto se puede deber a que las distribuciones que la componen están más o menos centradas. Entre ellas la que parece más normal es la de "saab" y las de "bus" y "van" presentan dos picos ambas.

Estudiamos "Max_length_aspect_ratio-Class":

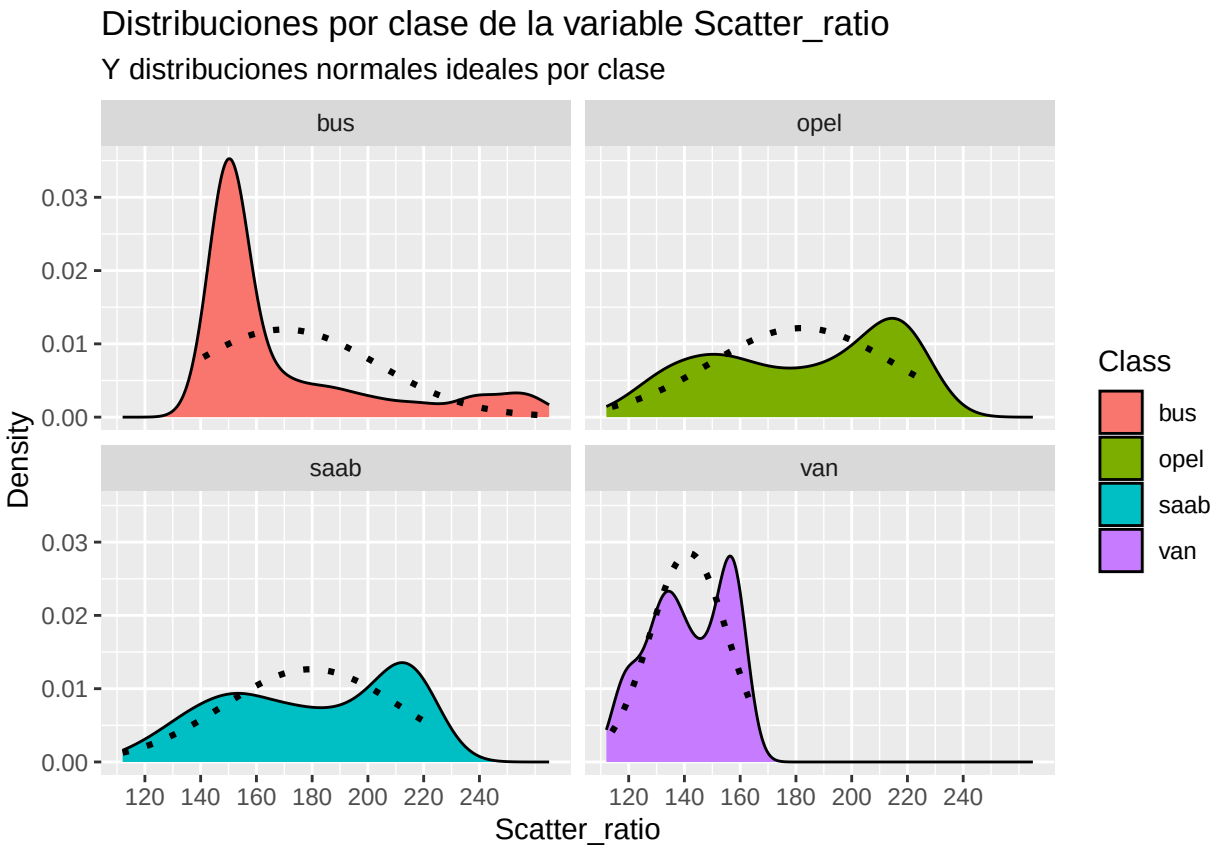
```
# Max_length_aspect_ratio-Class

vehicle[-Max_length_aspect_ratio.outliers,] %>%
  ggplot(aes(x=Max_length_aspect_ratio, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Max_length_aspect_ratio,
                          mean = tapply(Max_length_aspect_ratio, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Max_length_aspect_ratio, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Max_length_aspect_ratio',y='Density',
       title = 'Distribuciones por clase de la variable Max_length_aspect_ratio',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.05))+
  scale_x_continuous(breaks = seq(from=0, to=15, by=1))
```



En esta distribución se obtienen valores curiosos para la clase "bus" que tiene una pequeña distribución con cinco picos muy abruptos, mientras que el resto de las categorías presentan distribuciones similares con pendientes más suaves y mayor dispersión en los datos.

En "Scatter_ratio-Class":

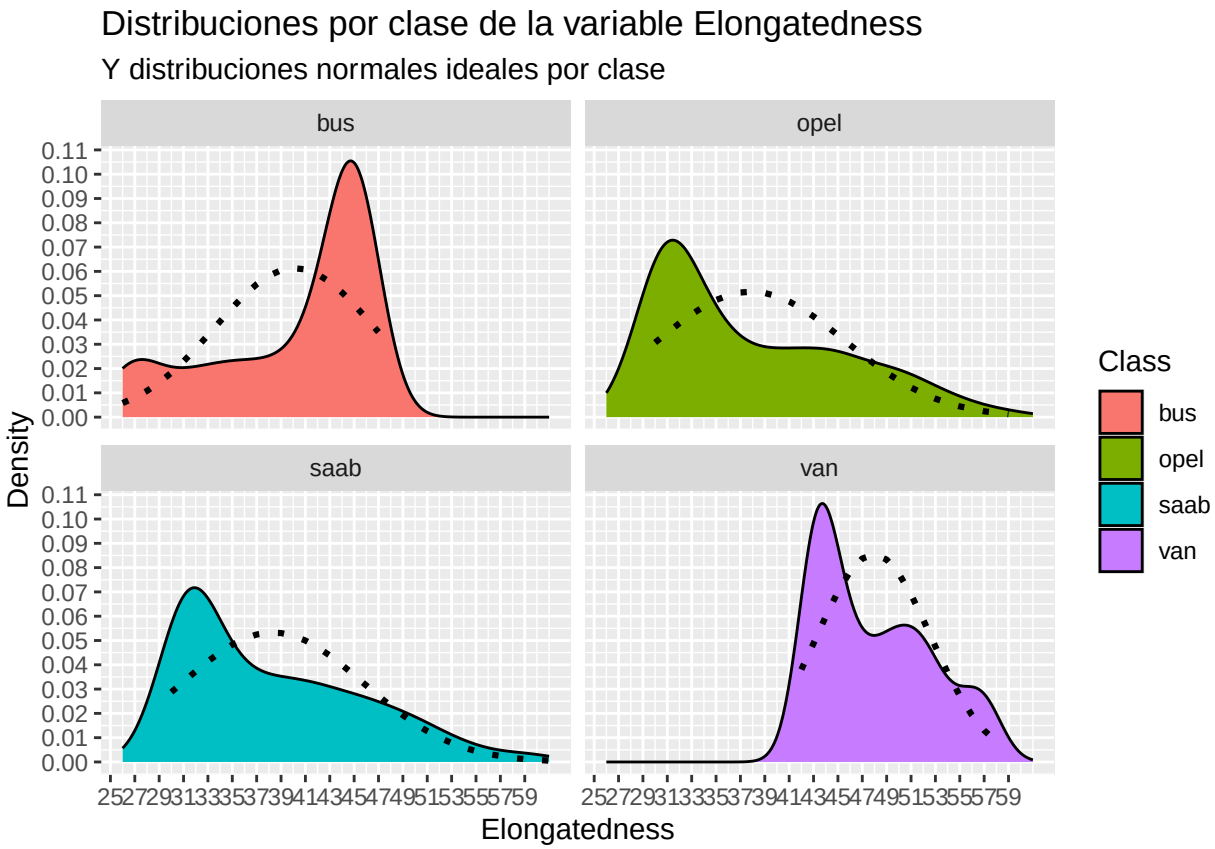
```
# Scatter_ratio-Class

vehicle %>%
  ggplot(aes(x=Scatter_ratio, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Scatter_ratio,
                          mean = tapply(Scatter_ratio, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Scatter_ratio, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Scatter_ratio',y='Density',
       title = 'Distribuciones por clase de la variable Scatter_ratio',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=120, to=250, by=20))
```



Tenemos distribuciones bien distinguidas para las categorías de "bus" y "van", mientras que las otras dos categorías tienen una distribución muy similar entre ellas.

Para "Elongatedness-Class":

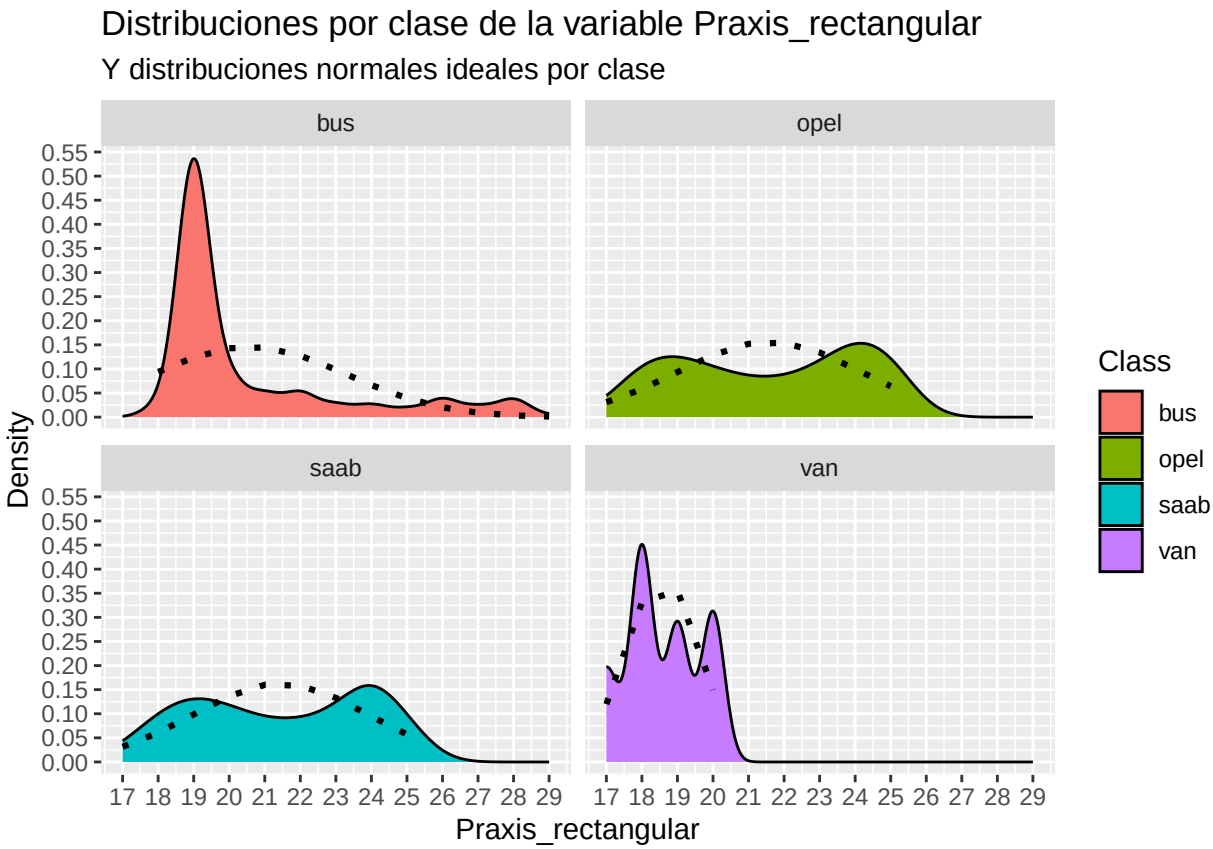
```
# Elongatedness-Class

vehicle %>%
  ggplot(aes(x=Elongatedness, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Elongatedness,
                          mean = tapply(Elongatedness, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Elongatedness, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Elongatedness',y='Density',
       title = 'Distribuciones por clase de la variable Elongatedness',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=25, to=60, by=2))
```



Obtenemos distribuciones parecidas entre las clases "saab" y "opel" mientras que las otras dos presentan un pico grande entre los mismos datos, pero el resto de los datos se encuentran en valores menores para la clase "bus" y valores más altos en "van" (uno a la izquierda del pico y el otro a la derecha).

Para "Praxis_rectangular-Class":

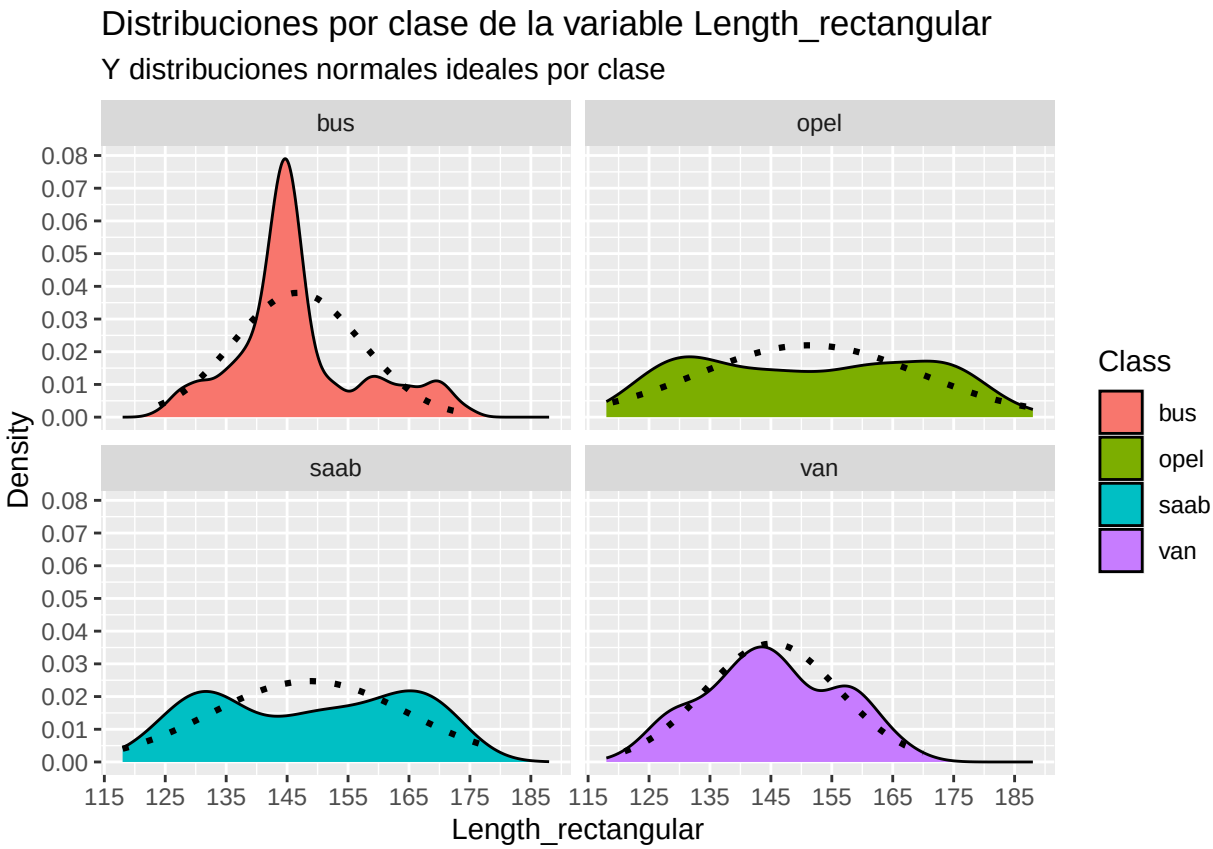
```
# Praxis_rectangular-Class

vehicle %>%
  ggplot(aes(x=Praxis_rectangular, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Praxis_rectangular,
                          mean = tapply(Praxis_rectangular, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Praxis_rectangular, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Praxis_rectangular',y='Density',
       title = 'Distribuciones por clase de la variable Praxis_rectangular',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.05))+
  scale_x_continuous(breaks = seq(from=0, to=30, by=1))
```



Para esta variable tenemos otra vez distribuciones distinguidas entre "bus" y "van", mientras que las otras dos categorías tienen distribuciones muy parecidas.

En el caso del par "Length_rectangular-Class":

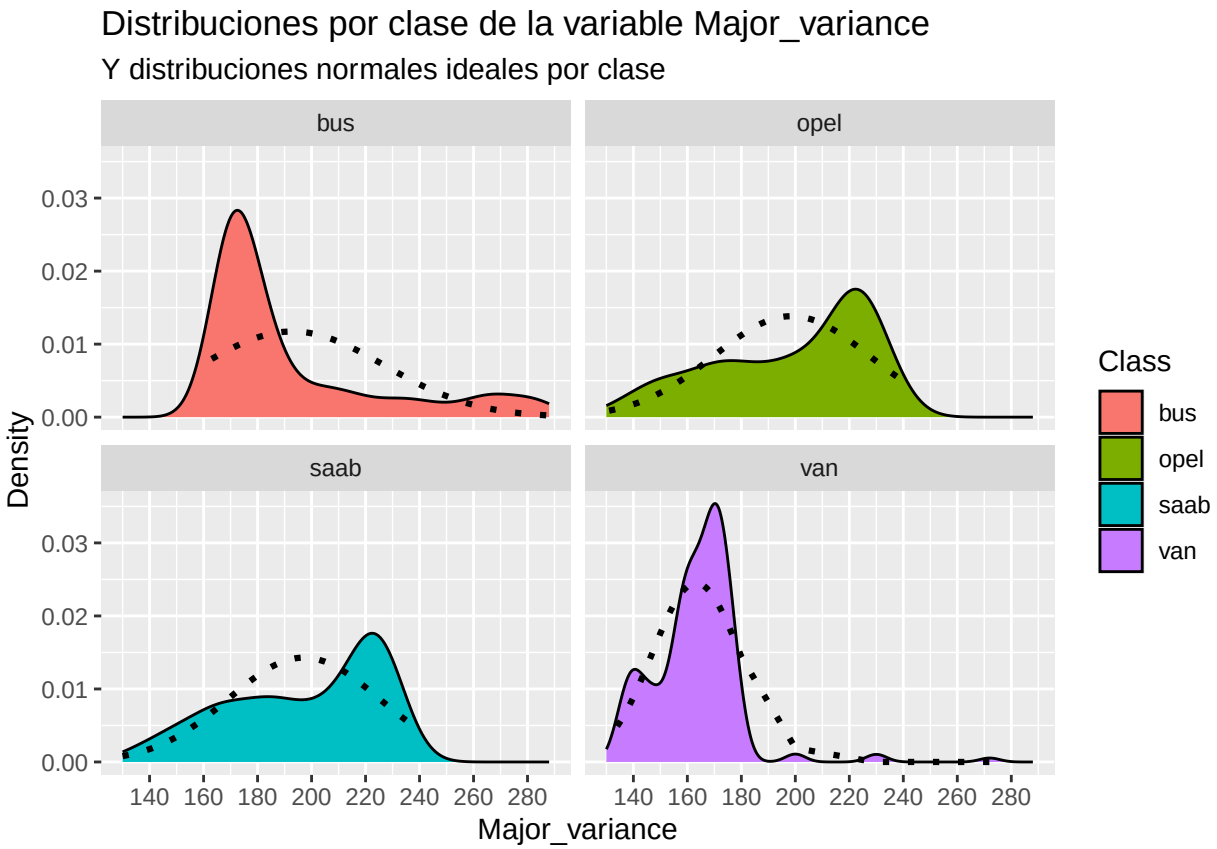
```
# Length_rectangular-Class

vehicle %>%
  ggplot(aes(x=Length_rectangular, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Length_rectangular,
                          mean = tapply(Length_rectangular, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Length_rectangular, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Length_rectangular',y='Density',
       title = 'Distribuciones por clase de la variable Length_rectangular',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=115, to=200, by=10))
```



Se obtienen distribuciones similares entre "opel" y "saab" mientras que la de "bus" es completamente diferente con un pico extremo y la de "van" es la que más se acerca a una distribución normal.

Para el par "Major_variance-Class"

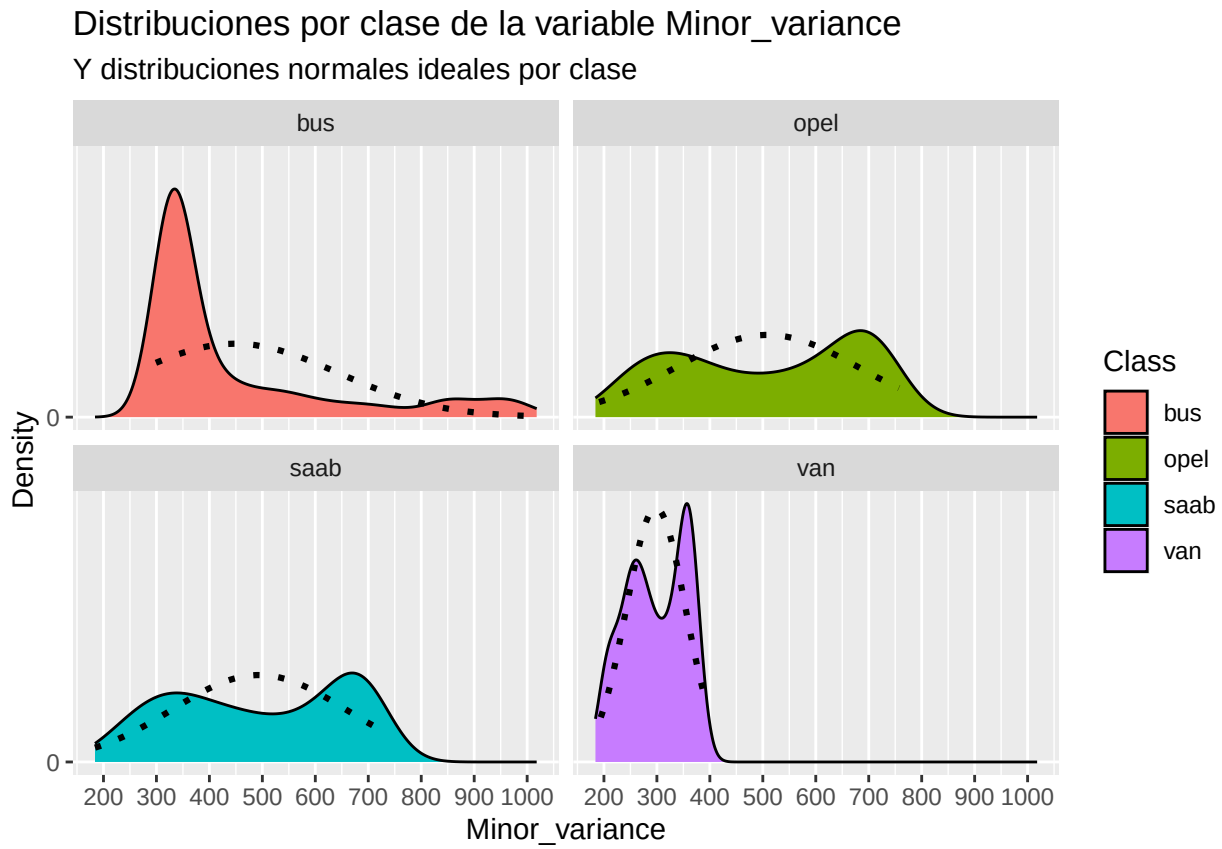
```
# Major_variance-Class

vehicle[-Major_variance.outliers,] %>%
  ggplot(aes(x=Major_variance, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Major_variance,
                          mean = tapply(Major_variance, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Major_variance, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Major_variance',y='Density',
       title = 'Distribuciones por clase de la variable Major_variance',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=350, by=20))
```



Se puede apreciar que las categorías "van" y "bus" son bastante únicas, teniendo la de "van" outliers asociados a la distribución de su categoría. Las otras dos categorías en cambio, tienen distribuciones muy parecidas para esta variable. Con la cantidad de atributos que hemos visto en los que pasa esto precisamente, parece que el reto de esta clasificación será distinguir estas dos modelos de coches.

En el par "Minor_variance-Class" tenemos:

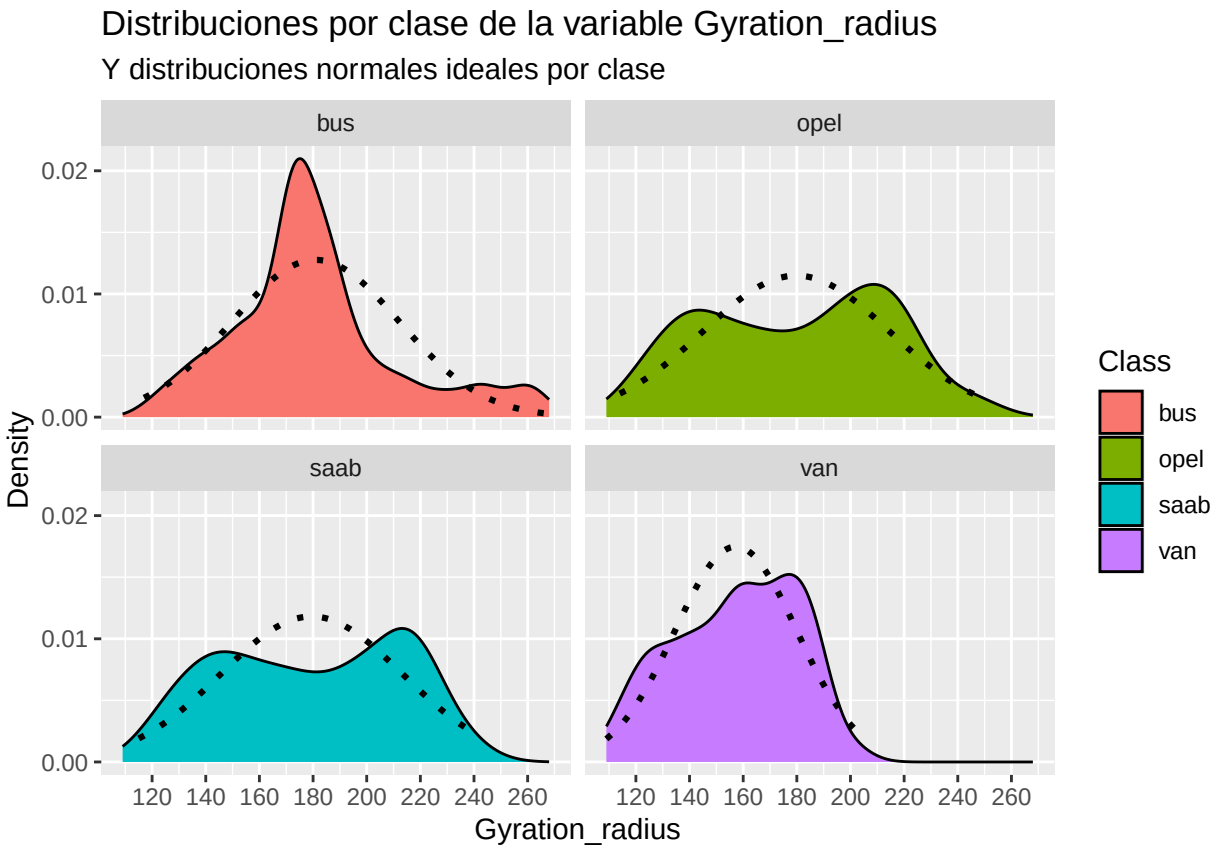
```
# Minor_variance-Class

vehicle %>%
  ggplot(aes(x=Minor_variance, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Minor_variance,
                          mean = tapply(Minor_variance, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Minor_variance, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Minor_variance',y='Density',
       title = 'Distribuciones por clase de la variable Minor_variance',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=200, to=1000, by=100))
```



Un pico bastante pronunciado para la distribución de "bus" y dos picos con los datos muy concentrados para la de "van", mientras que los dos coches tienen distribuciones casi idénticas.

Para el par "Gyration_radius-Class":

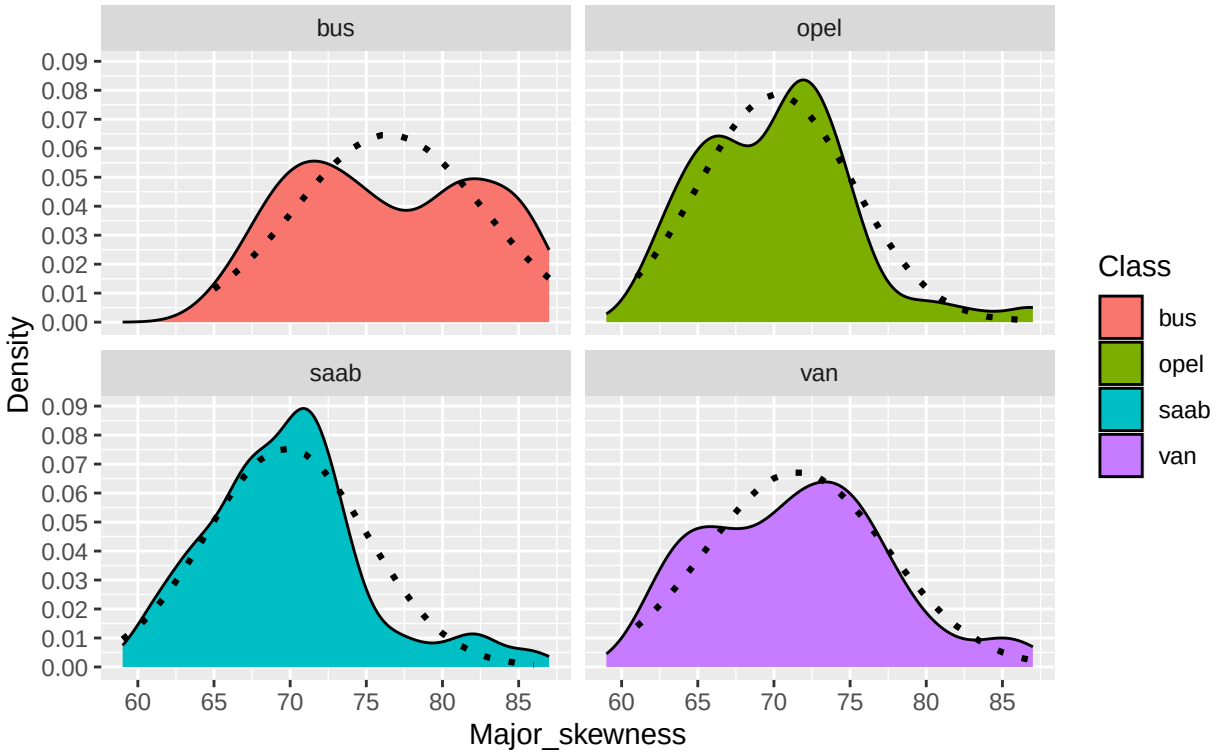
```
# Gyration_radius-Class

vehicle %>%
  ggplot(aes(x=Gyration_radius, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Gyration_radius,
                          mean = tapply(Gyration_radius, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Gyration_radius, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Gyration_radius',y='Density',
       title = 'Distribuciones por clase de la variable Gyration_radius',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=120, to=260, by=20))
```



De nuevo, distribuciones bastante diferentes con el autobús y la furgoneta, mientras que los dos coches ofrecen una distribución casio igual. La que más se acercaría a una normal en este caso sería la categoría "van", veremos más adelante si pasa el test.

Con el par "Major_skewness-Class":

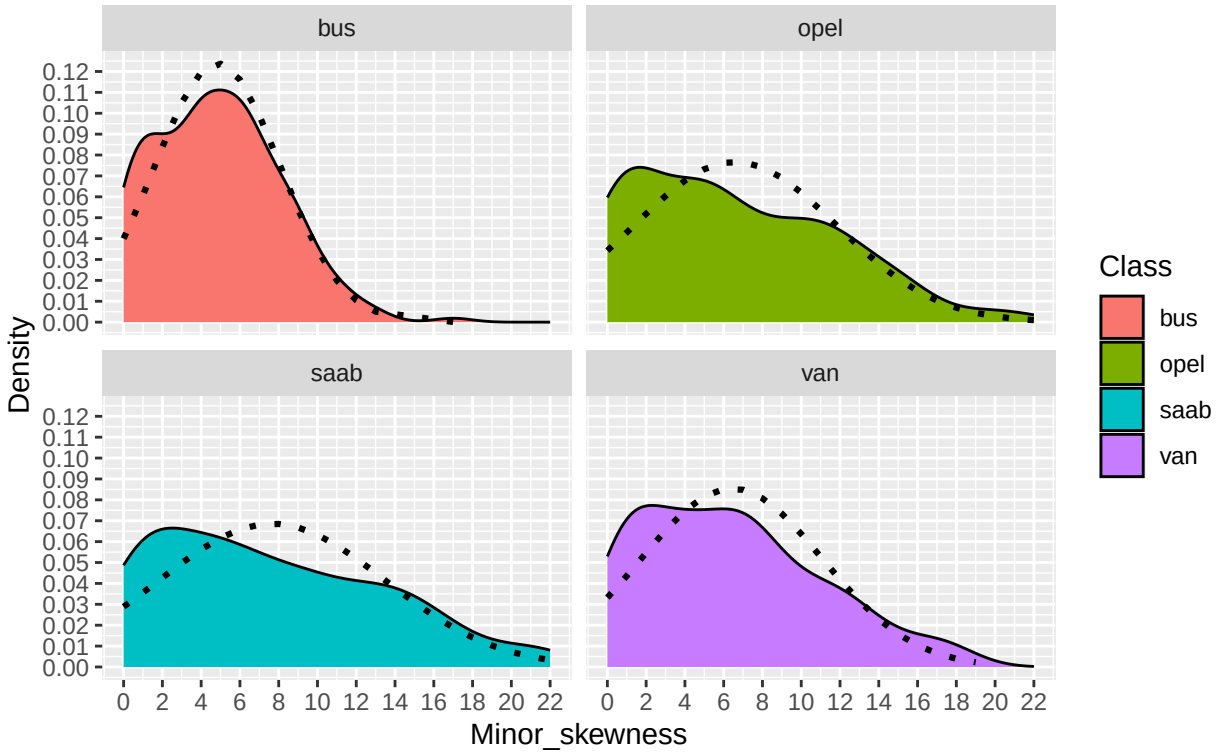
```
# Major_skewness-Class

vehicle[-Major_skewness.outliers,] %>%
  ggplot(aes(x=Major_skewness, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Major_skewness,
                          mean = tapply(Major_skewness, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Major_skewness, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Major_skewness',y='Density',
       title = 'Distribuciones por clase de la variable Major_skewness',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=120, by=5))
```



En esta variable obtenemos diferencias en las distribuciones de las dos clases de coches, mientras que la de "van" no dista mucho de la de "opel". La distribución de bus es muy diferente al resto. Estimo que son estas variables las que vana ser de vital importancia para clasificar correctamente entre "opel" y "saab".

Con "Minor_skewness-Class" se tiene:

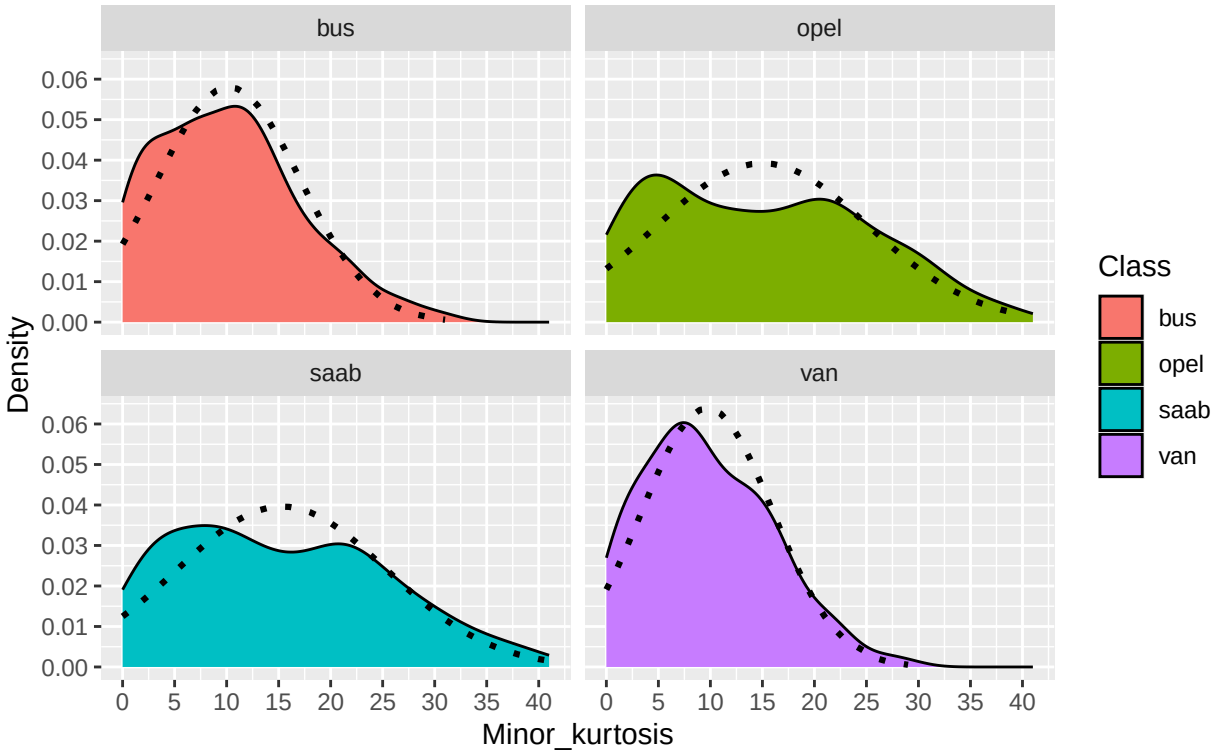
```
# Minor_skewness-Class

vehicle %>%
  ggplot(aes(x=Minor_skewness, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Minor_skewness,
                          mean = tapply(Minor_skewness, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Minor_skewness, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Minor_skewness',y='Density',
       title = 'Distribuciones por clase de la variable Minor_skewness',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=25, by=2))
```



Distribuciones parecidas entre todas las categorías menos la de "bus", por las variables que llevamos estudiadas hasta ahora estimo que la clase más fácil para clasificar será la de "bus" (es evidente si pensamos en el problema ya que un autobus tiene una silueta muy diferente al resto de los vehículos).

Para el par "Minor_kurtosis-Class":

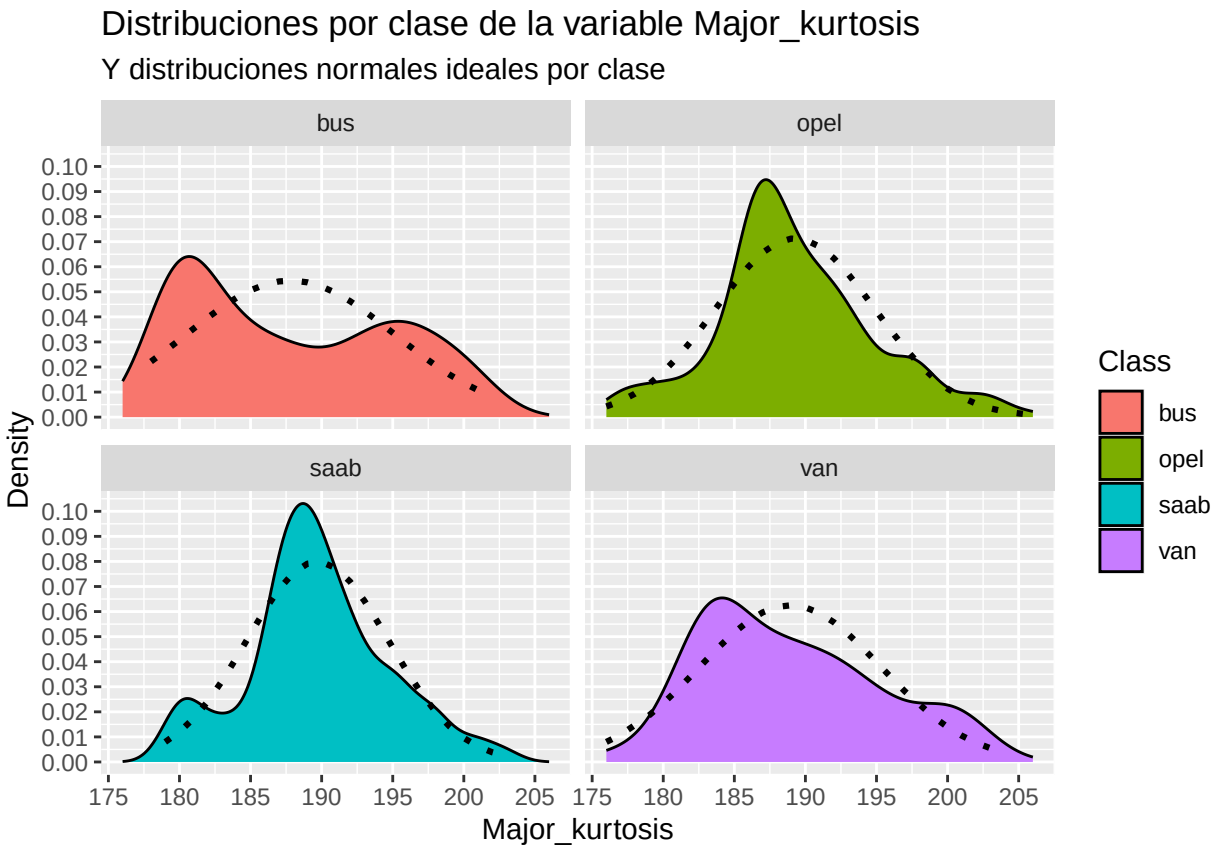
```
# Minor_kurtosis-Class

vehicle %>%
  ggplot(aes(x=Minor_kurtosis, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Minor_kurtosis,
                          mean = tapply(Minor_kurtosis, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Minor_kurtosis, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Minor_kurtosis',y='Density',
       title = 'Distribuciones por clase de la variable Minor_kurtosis',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=0, to=40, by=5))
```



Obtenemos distribuciones de "bus" y "van" parecidas entre ellas, y por otro lado "opel" y "saab" con un parecido muy alto.

Estudiamos "Major_kurtosis-Class":

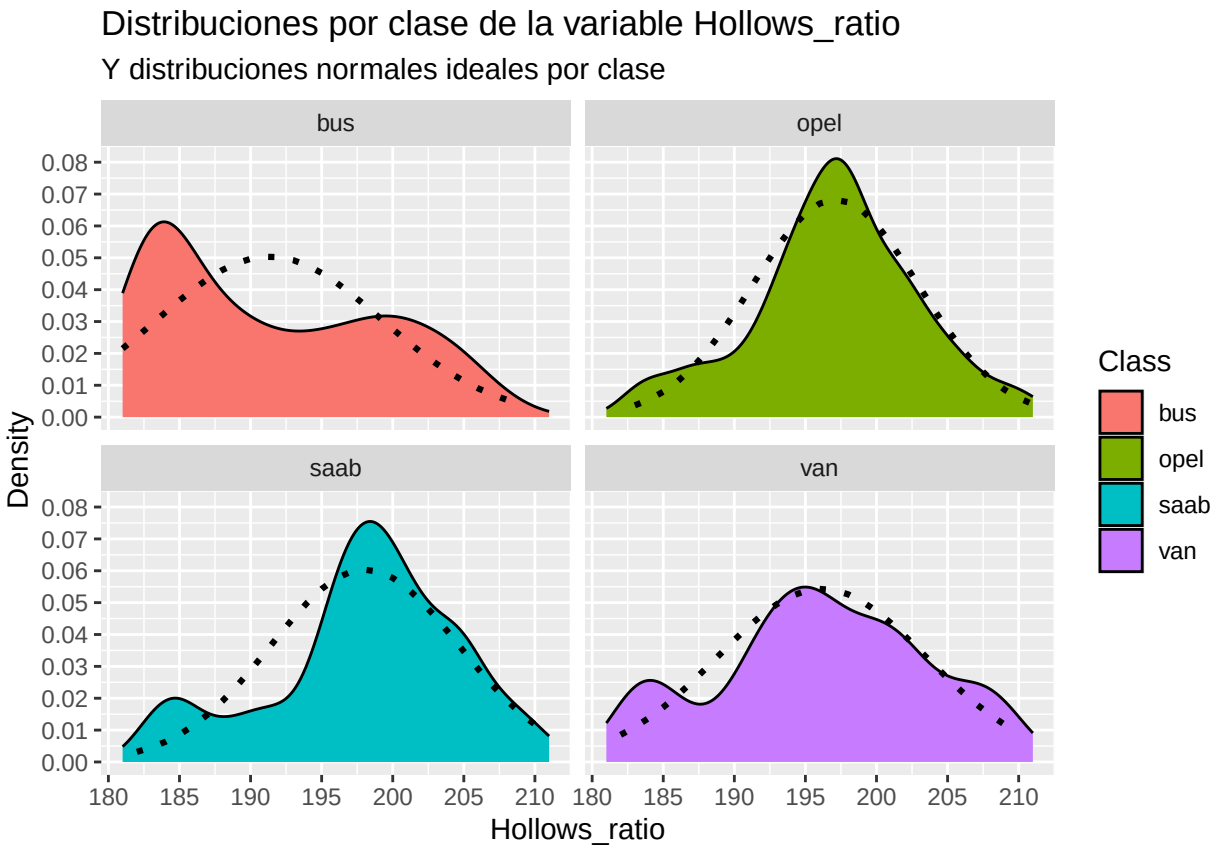
```
# Major_kurtosis-Class

vehicle %>%
  ggplot(aes(x=Major_kurtosis, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Major_kurtosis,
                          mean = tapply(Major_kurtosis, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Major_kurtosis, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Major_kurtosis',y='Density',
       title = 'Distribuciones por clase de la variable Major_kurtosis',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=175, to=210, by=5))
```



Hay ciertas diferencias entre todas las categorías, esta variable puede ser importante para la clasificación.

Finalmente con "Hollows_ratio-Class":

```
# Hollows_ratio-Class

vehicle %>%
  ggplot(aes(x=Hollows_ratio, fill=Class)) +
  geom_density(linetype = 1)+
  facet_wrap(~Class)+
  geom_line(aes(y = dnorm(Hollows_ratio,
                          mean = tapply(Hollows_ratio, Class, mean, na.rm = TRUE)[PANEL],
                          sd = tapply(Hollows_ratio, Class, sd, na.rm = TRUE)[PANEL])),
            color =1, lwd=1.1,linetype = 3)+
  labs(x='Hollows_ratio',y='Density',
       title = 'Distribuciones por clase de la variable Hollows_ratio',
       subtitle ='Y distribuciones normales ideales por clase')+
  scale_y_continuous(breaks = seq(from=0, to=1, by=0.01))+
  scale_x_continuous(breaks = seq(from=180, to=210, by=5))
```



Hay ciertas diferencias en cada distribución, con "saab" teneindo cierto parecido con "van", mientras que "opel" no presenta el segundo pico menor en los valores más bajos. La categoría "bus" es completamente diferente al resto como ha pasado en la mayoría de las variables.

Descomposición de atributos complicados

En este dataset ha trabajado con variables cuantitativas discretas, por lo que no haría falta nincuna transformación. Como mucho codificar la variable de salida categórica con valores "dummy" para algoritmos basados en distancia. Aunque la mayoría de los paquetes acepta variables categóricas, si quisiésemos utilizar un algoritmo propio (por ejemplo la función my_knn que hicimos como ejercicio en la eval. continua) si sería importante descomponer esta variable categórica en su codificación binaria.

```
# Creamos variables dummy
vehicle.dummy <- dummy_cols(vehicle,remove_selected_columns = TRUE)
head(vehicle.dummy)
```

```
##   Compactness Circularity Distance_circularity Radius_ratio Praxis_aspect_ratio
## 1          95          48                  83          178                  72
## 2          91          41                  84          141                  57
## 3         104          50                 106          209                  66
## 4          93          41                  82          159                  63
## 5          85          44                  70          205                 103
## 6         107          57                 106          172                  50
##   Max_length_aspect_ratio Scatter_ratio Elongatedness Praxis_rectangular
## 1                      10           162            42                 20
## 2                       9           149            45                 19
## 3                      10           207            32                 23
```

```
## 4                     9           144            46                19
## 5                    52           149            45                19
## 6                     6           255            26                28
##   Length_rectangular Major_variance Minor_variance Gyration_radius
## 1                159            176            379             184
## 2                143            170            330             158
## 3                158            223            635             220
## 4                143            160            309             127
## 5                144            241            325             188
## 6                169            280            957             264
##   Major_skewness Minor_skewness Minor_kurtosis Major_kurtosis Hollows_ratio
## 1             70              6             16            187           197
## 2             72              9             14            189           199
## 3             73             14              9            188           196
## 4             63              6             10            199           207
## 5            127              9             11            180           183
## 6             85              5              9            181           183
##   Class_bus Class_opel Class_saab Class_van
## 1         0          0          0         1
## 2         0          0          0         1
## 3         0          0          1         0
## 4         0          0          0         1
## 5         1          0          0         0
## 6         1          0          0         0
```

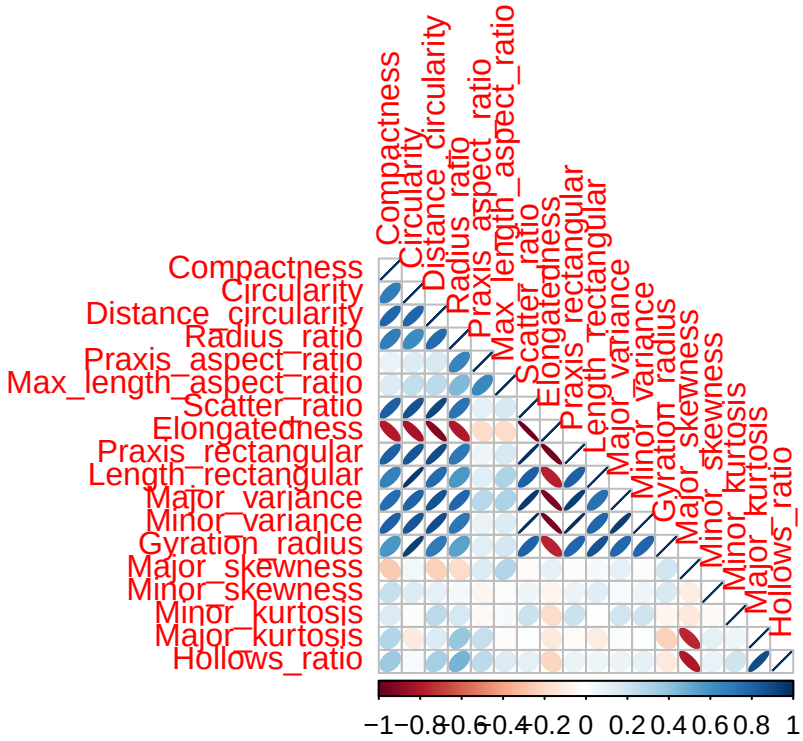
Así quedaría el dataset.

Búsqueda de datos redundantes

La reducción de dimensionalidad es un aspecto importante a la hora de aplicar un algoritmo de clasificación, por un lado nos permite contrarestar el problema de alta dimensionalidad, y por otro lado permite simplificar el modelo seleccionando las variables que mayor influencia tengan en la variable objetivo.

Calculamos la matriz de correlación para estudiar la correlación que tenemos entre nuestra variables.

```
# Estudiamos la correlación entre las variables mediante su matriz de correlacción
vehicle.CorMatrix <- cor(vehicle[,-19])
corrplot(vehicle.CorMatrix,method = "ellipse", type = "lower")
```



Las variables que podríamos descartar observando el gráfico podrían ser "Elongatedness" que tiene una correlación negativa muy alta con 8 variables diferentes, después probaría quitar las variables "Praxis_rectangular", "Length_rectangular", "Major_variance" y "Minor_variance", ya que tienen una correlación positiva alta entre ellas y con otras variables, por lo que considero que no estaría ganando en información.

Transformación de datos

Para aquellos algoritmos basados en distancias es importante realizar un escalado de nuestras variables numéricas, ya que de lo contrario aquellas con valores más extremos y dispersos podrían tener una contribución mayor en el cálculo de la distancia. Es por eso que la normalización es necesaria para asegurarnos de que las variables contribuyen por igual.

El escalado lo aplicaremos sobre el dataframe con valores "dummy" que hemos calculado antes, ya que sería el que usaríamos para este tipo de algoritmos.

```
# Escalado
vehicle.scaled <- vehicle.dummy %>% mutate_if(is.numeric, scale, center = TRUE, scale = TRUE)
head(vehicle.scaled)
```

```
##   Compactness Circularity Distance_circularity Radius_ratio Praxis_aspect_ratio
## 1  0.16048541   0.5086493          0.057784334   0.27064568           1.3065186
## 2 -0.32527723  -0.6258973          0.121189713  -0.83474981          -0.5950436
## 3  1.25345137   0.8328055          1.516108039   1.19678784           0.5458937
## 4 -0.08239591  -0.6258973         -0.005621044  -0.29698984           0.1655813
## 5 -1.05392120  -0.1396630         -0.766485586   1.07728563           5.2364137
## 6  1.61777335   1.9673521          1.516108039   0.09139236          -1.4824393
##   Max_length_aspect_ratio Scatter_ratio Elongatedness Praxis_rectangular
```

```
## 1              0.31135767    -0.2057226     0.1364892         -0.2248114
## 2              0.09402385    -0.5967591     0.5205355         -0.6105933
## 3              0.31135767     1.1478653    -1.1436648          0.9325342
## 4              0.09402385    -0.7471578     0.6485509         -0.6105933
## 5              9.43937817    -0.5967591     0.5205355         -0.6105933
## 6             -0.55797761     2.5916924    -1.9117573          2.8614436
##   Length_rectangular Major_variance Minor_variance Gyration_radius
## 1          0.7578841     -0.4021456     -0.3447306       0.2856434
## 2         -0.3443743     -0.5932598     -0.6220483      -0.5132139
## 3          0.6889930      1.0949159      1.1041132       1.3917535
## 4         -0.3443743     -0.9117835     -0.7408988      -1.4656975
## 5         -0.2754832      1.6682585     -0.6503461       0.4085445
## 6          1.4467957      2.9105010      2.9264871       2.7436658
##   Major_skewness Minor_skewness Minor_kurtosis Major_kurtosis Hollows_ratio
## 1    -0.32886116    -0.07666562      0.3807656    -0.31353666    0.18384857
## 2    -0.06173054     0.53329468      0.1568326     0.01093064    0.45270923
## 3     0.07183477     1.54989517     -0.4030001    -0.15130301    0.04941824
## 4    -1.26381831    -0.07666562     -0.2910336     1.63326713    1.52815188
## 5     7.28436143     0.53329468     -0.1790670    -1.44917221   -1.69817606
## 6     1.67461847    -0.27998571     -0.4030001    -1.28693856   -1.69817606
##    Class_bus Class_opel Class_saab  Class_van
## 1 -0.5888323 -0.5779183  -0.587013  1.8020580
## 2 -0.5888323 -0.5779183  -0.587013  1.8020580
## 3 -0.5888323 -0.5779183   1.701526 -0.5542651
## 4 -0.5888323 -0.5779183  -0.587013  1.8020580
## 5  1.6962691 -0.5779183  -0.587013 -0.5542651
## 6  1.6962691 -0.5779183  -0.587013 -0.5542651
```

Conclusiones

Una vez realizado el análisis exploratorio de datos procedemos a responder a las hipótesis originalmente planteadas:

- Hay diferencias en los parámetros de silueta dependiendo del vehículo. Si las hay, la más evidente es la forma de la distribución de los datos según la clase de vehículo.
- Al tener dos coches entre las clases la dificultad del problema estará en clasificar estas dos clases. ¿Cómo se ve reflejada esa similitud? Efectivamente hemos comprobado que para muchas variables las distribuciones de estas dos categorías es casi idéntica.
- La clase autobús es la más sencilla de clasificar puesto que la silueta difiere mucho de las del resto. Así es, entre las distribuciones, la clase "bus" era la que más diferencias tenía con el resto de clases para todas las variables.

Por lo general la dificultad de este dataset estaba en entender como afectan las variables, ya que sin ser expertos en la materia y con la poca información que había disponible sobre los atributos, además de la cantidad numerosa de ellos, ha hecho complicado analizar en profundidad el conjunto de datos.

Clasificación

Tras profundizar en las variables en el análisis exploratorio de los datos, procedemos a realizar el estudio de clasificación para el dataset.

Lo primero que se debe hacer es cargar los paquetes necesarios para aplicar los algoritmos de clasificación, además de asignar la ruta donde se encuentran los archivos que contienen las particiones proporcionadas de los datos.

```
require(tidyverse)
require(readr)
require(caret)
require(ggplot2)

nombre <- "Input/vehicle/vehicle"
```

Estudio de K-nn con validación cruzada y distintos valores de k

Para este estudio se ha creado una función específica que carga las distintas particiones de entrenamiento y "test" para entrenar el modelo y hacer predicciones con cada subconjunto, el valor medio de cada fold se va almacenando en un dataframe para distintos valores de k.

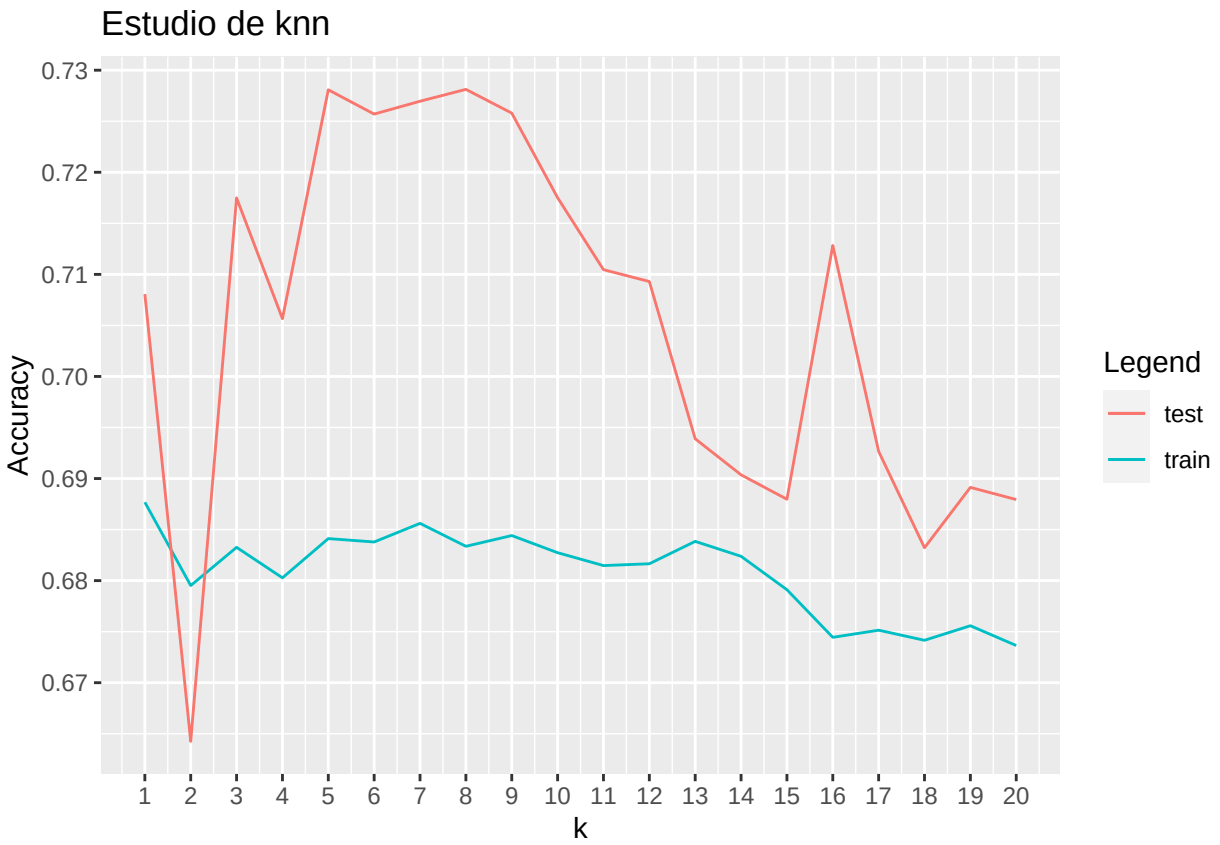
```
# Aplicamos knn con validación cruzada para distintos valores de k

#------------- 10-fold cross-validation KNN todas las variables
run_knn_k_fold <- function(kmax, nfolds, x){
  set.seed(1) # set.seed es necesario si utilizamos caret.
  accuracy.df <- data.frame('k'=1:kmax, 'train'=1:kmax, 'test'=1:kmax)
  for (k in 1:kmax){
    accuracy.train <- 1:nfolds
    accuracy.test <- 1:nfolds
    for (i in 1:nfolds){
      file <- paste(x, "-10-", i, "tra.dat", sep="")
      x_tra <- read.csv(file, comment.char="@", header=FALSE)
      file <- paste(x, "-10-", i, "tst.dat", sep="")
      x_tst <- read.csv(file, comment.char="@", header=FALSE)
      In <- length(names(x_tra)) - 1
      names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
      names(x_tra)[In+1] <- "Y"
      names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
      names(x_tst)[In+1] <- "Y"


      knnModel <- train(x=x_tra %>% select(-Y), y = x_tra[,"Y"],
                        method = "knn", preProc = c("center", "scale"),
                        metric="Accuracy", tuneGrid = data.frame(k=k))

      knnPred <- predict(knnModel, newdata = x_tst %>% select(-Y))
      cfm <- table(knnPred, x_tst[,'Y'])
      accuracy.train[i] <- knnModel$results$Accuracy
      accuracy.test[i] <- sum(diag(cfm))/length(x_tst[,'Y'])
    }
    accuracy.df[k,'train']<- mean(accuracy.train)
    accuracy.df[k,'test']<- mean(accuracy.test)
  }
accuracy.df
}
```

Los argumentos que hay que pasar son: kmax (el número de k máxima que se quiere estudiar), nfolds (el número de folds en el que están particionados los datos, para nuestro caso 10) y x (la ruta del archivo donde se encuentran los datos). Si ejecutamos la función, nos devolverá el dataframe accuracy.df con los resultados.

```
# llamamos a la función
resultados <- run_knn_k_fold(20, 10, nombre)
```

Se ha escogido un valor de k=20, si queremos mostrar los resultados podemos contruir un gráfico:

```
# Mostramos los resultados
colors <- c('train'='red', 'test'='blue')
ggplot(resultados, aes(x=k))+
  geom_line(aes(y=train, color="train")) +
  geom_line(aes(y=test, color="test")) +
  labs(title="Estudio de knn", y="Accuracy",  color = "Legend") +
  scale_x_continuous(breaks=seq(0, 21, 1)) +
  scale_y_continuous(breaks=seq(0, 1, 0.01))
```



Se han obtenido valores más altos de precisión en el conjunto de "test" que en el de entrenamiento, lo cual es extraño y creo que la diferencia está en cómo calcula el paquete la precisión del modelo para "train", mientras que en "test" se ha calculado manualmente usando la matriz de confusión.

Si queremos obtener los valores de k con mayor precisión:

```
resultados[which.max(resultados$train),]
```

```
##   k     train      test
## 1 1 0.6876804 0.7080672
```

```
resultados[which.max(resultados$test),]
```

```
##   k     train      test
## 8 8 0.6833627 0.7281232
```

Utilizar el algoritmo LDA para clasificar

Lo primero que se debe hacer antes de aplicar este algoritmo es comprobar que se cumplen las asunciones. Siendo la primera de estas que nuestros datos se hayan muestreado aleatoriamente, suponemos que así es. La siguiente asunción es que los datos para cada categoría y para cada variable se distribuyen de forma normal. Para ello se realiza el test de Shapiro-Wilk. En este caso ejecutaré todos los test de golpe y veré que variables tienen distribución normal según los resultados.

```
# Realizamos test de shapiro por categorías a cada variable

# Compactness
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Compactness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.93801, p-value = 5.358e-08
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Compactness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.96976, p-value = 0.0001631
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Compactness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.9708, p-value = 0.0001824
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Compactness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.98193, p-value = 0.01157
```

```
# Circularity
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Circularity)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.92716, p-value = 6.479e-09
```

```r
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Circularity)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.92443, p-value = 5.731e-09
```

```r
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Circularity)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.91181, p-value = 4.737e-10
```

```r
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Circularity)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.95977, p-value = 1.961e-05
```

```r
# Distance_circularity
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Distance_circularity)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.8402, p-value = 3.135e-14
```

```r
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Distance_circularity)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.91959, p-value = 2.446e-09
```

```r
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Distance_circularity)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.9154, p-value = 8.617e-10
```

```r
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Distance_circularity)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.95771, p-value = 1.184e-05
```

```
# Radius_ratio
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Radius_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.96905, p-value = 0.0001041
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Radius_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.9631, p-value = 2.524e-05
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Radius_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.95186, p-value = 1.182e-06
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Radius_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.70619, p-value < 2.2e-16
```

```
# Praxis_aspect_ratio
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Praxis_aspect_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.86218, p-value = 4.022e-13
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Praxis_aspect_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  p1[, 1]
## W = 0.97739, p-value = 0.001735
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Praxis_aspect_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.98893, p-value = 0.09299
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Praxis_aspect_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.57651, p-value < 2.2e-16
```

```
# Max_length_aspect_ratio
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Max_length_aspect_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.25494, p-value < 2.2e-16
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Max_length_aspect_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.95852, p-value = 7.692e-06
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Max_length_aspect_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.94558, p-value = 2.828e-07
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Max_length_aspect_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.41757, p-value < 2.2e-16
```

```r
# Scatter_ratio
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Scatter_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.71629, p-value < 2.2e-16
```

```r
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Scatter_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.91815, p-value = 1.912e-09
```

```r
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Scatter_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.92593, p-value = 5.486e-09
```

```r
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Scatter_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.93799, p-value = 1.634e-07
```

```r
# Elongatedness
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Elongatedness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.80862, p-value = 1.197e-15
```

```r
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Elongatedness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.87701, p-value = 4.208e-12
```

```r
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Elongatedness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.89156, p-value = 2.121e-11
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Elongatedness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.90004, p-value = 2.723e-10
```

```
# Praxis_rectangular
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Praxis_rectangular)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.67052, p-value < 2.2e-16
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Praxis_rectangular)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.90256, p-value = 1.548e-10
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Praxis_rectangular)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.91511, p-value = 8.202e-10
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Praxis_rectangular)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.86392, p-value = 2.335e-12
```

```
# Length_rectangular
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Length_rectangular)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  p1[, 1]
## W = 0.92889, p-value = 8.943e-09
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Length_rectangular)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.94486, p-value = 3.175e-07
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Length_rectangular)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.93671, p-value = 4.36e-08
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Length_rectangular)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.98265, p-value = 0.01472
```

```
# Major_variance
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Major_variance)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.74222, p-value < 2.2e-16
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Major_variance)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.91539, p-value = 1.2e-09
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Major_variance)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.91544, p-value = 8.673e-10
```

```r
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Major_variance)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.71385, p-value < 2.2e-16
```

```r
# Minor_variance
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Minor_variance)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.70101, p-value < 2.2e-16
```

```r
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Minor_variance)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.9151, p-value = 1.142e-09
```

```r
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Minor_variance)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.92003, p-value = 1.907e-09
```

```r
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Minor_variance)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.93121, p-value = 4.494e-08
```

```r
# Gyration_radius
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Gyration_radius)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.93929, p-value = 6.981e-08
```

```r
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Gyration_radius)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.95701, p-value = 5.268e-06
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Gyration_radius)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.9472, p-value = 4.055e-07
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Gyration_radius)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.96334, p-value = 4.843e-05
```

```
# Major_skewness
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Major_skewness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.85305, p-value = 1.351e-13
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Major_skewness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.94847, p-value = 7.045e-07
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Major_skewness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.95227, p-value = 1.303e-06
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Major_skewness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  p1[, 1]
## W = 0.80206, p-value = 3.706e-15
```

```
# Minor_skewness
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Minor_skewness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.96219, p-value = 1.513e-05
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Minor_skewness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.93877, p-value = 8.862e-08
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Minor_skewness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.94085, p-value = 1.024e-07
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Minor_skewness)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.94661, p-value = 9.521e-07
```

```
# Minor_kurtosis
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Minor_kurtosis)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.96144, p-value = 1.238e-05
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Minor_kurtosis)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.95315, p-value = 2.072e-06
```

```r
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Minor_kurtosis)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.96095, p-value = 1.138e-05
```

```r
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Minor_kurtosis)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.96842, p-value = 0.0001889
```

```r
# Major_kurtosis
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Major_kurtosis)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.90244, p-value = 9.953e-11
```

```r
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Major_kurtosis)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.97474, p-value = 0.000739
```

```r
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Major_kurtosis)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.97801, p-value = 0.001814
```

```r
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Major_kurtosis)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.95752, p-value = 1.132e-05
```

```r
# Hollows_ratio
p1 <- vehicle %>% filter(Class == "bus") %>% dplyr::select(Hollows_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.9047, p-value = 1.414e-10
```

```
p1 <- vehicle %>% filter(Class == "opel") %>% dplyr::select(Hollows_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.98206, p-value = 0.00845
```

```
p1 <- vehicle %>% filter(Class == "saab") %>% dplyr::select(Hollows_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.95701, p-value = 4.141e-06
```

```
p1 <- vehicle %>% filter(Class == "van") %>% dplyr::select(Hollows_ratio)
shapiro.test(p1[,1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p1[, 1]
## W = 0.96932, p-value = 0.0002429
```

La única en la que el test nos indica normalidad es "Praxis_aspect_ratio" para la clase "saab", el resto obtienen un p-valor muy bajo, por lo que no tenemos evidencia estadística de que las distribuciones se asemejen a una normal. Lo siguiente es comprobar la homogeneidad de las varianzas. Al disponer de un dataset con muchos atributos no se van a mostrar las matrices, pero el código sería el siguiente:

```
# Comprobamos las varianzas
var(vehicle %>% filter(Class == "bus") %>% dplyr::select(-19))
var(vehicle %>% filter(Class == "opel") %>% dplyr::select(-19))
var(vehicle %>% filter(Class == "saab") %>% dplyr::select(-19))
var(vehicle %>% filter(Class == "van") %>% dplyr::select(-19))
```

Para estudiar la homogeneidad podemos aplicar el test de Levene, puesto que ninguna de nuestras variables se distribuye de forma normal (se realizó el test de Shapiro-Wilk en la parte de análisis exploratorio de datos sobre cada variable).

```
# Aplicamos tests para estudiar homogeneidad
leveneTest(Compactness ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  48.334 < 2.2e-16 ***
##       842
## ---
```

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

```
leveneTest(Circularity ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  54.481 < 2.2e-16 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Distance_circularity ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  22.656 4.115e-14 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Radius_ratio ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3   8.046 2.733e-05 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Praxis_aspect_ratio ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3   12.17 8.409e-08 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Max_length_aspect_ratio ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value   Pr(>F)
## group   3  4.9598 0.002035 **
##       842
## ---
```

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

```
leveneTest(Scatter_ratio ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  31.536 < 2.2e-16 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Elongatedness ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  11.163 3.447e-07 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Praxis_rectangular ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3   30.39 < 2.2e-16 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Length_rectangular ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  63.907 < 2.2e-16 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Major_variance ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3   16.78 1.356e-10 ***
##       842
## ---
```

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

```
leveneTest(Minor_variance ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  39.551 < 2.2e-16 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Gyration_radius ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  21.195 3.047e-13 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Major_skewness ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  11.209 3.233e-07 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Minor_skewness ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  25.561 7.856e-16 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Minor_kurtosis ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  34.664 < 2.2e-16 ***
##       842
## ---
```

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

```
leveneTest(Major_kurtosis ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  22.805 3.357e-14 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Hollows_ratio ~ Class, vehicle)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  13.495 1.317e-08 ***
##       842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Todas las variables han obtenido un p-valor pequeño en los tests (menor a 0.05), por lo que se concluye que no hay homogeneidad en las varianzas.

Aunque no se hayan cumplido las asunciones, se ha efectuado igualmente LDA. Para ello se ha utilizado una función parecida a la utilizada para knn:

```
# # función para LDA
run_lda_fold <- function(nfolds, x){
set.seed(1)
accuracy.df <- data.frame('fold'=1:nfolds, 'train'=1:nfolds, 'test'=1:nfolds)
for (i in 1:10){
  file <- paste(x, "-10-", i, "tra.dat", sep="")
  x_tra <- read.csv(file, comment.char="@", header=FALSE)
  file <- paste(x, "-10-", i, "tst.dat", sep="")
  x_tst <- read.csv(file, comment.char="@", header=FALSE)
  In <- length(names(x_tra)) - 1
  names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tra)[In+1] <- "Y"
  names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
  names(x_tst)[In+1] <- "Y"

  TrainData <- x_tra %>% select(-Y)
  TrainClasses <- x_tra %>% pull(Y)

  ldaModel <- train(TrainData, TrainClasses,
                    method = "lda", preProc = c("center", "scale"),
                    metric="Accuracy", tuneLength = 10)

  ldaPred <- predict(ldaModel, newdata = x_tst %>% select(-Y))
  cfm <- table(ldaPred, x_tst[,'Y'])
```

```
  accuracy.df[i,'train']<- ldaModel$results$Accuracy
  accuracy.df[i,'test']<- sum(diag(cfm))/length(x_tst[,'Y'])
}
accuracy.df
}
```

Ahora lo que hay que hacer es llamar a la función y visualizar los resultados. En este caso en el dataframe que devuelve la función viene la precisión por cada fold tanto para "train" como para "test".

```
# resultados
results.LDA <- run_lda_fold(10, nombre)
results.LDA
```

```
##    fold     train      test
## 1     1 0.7731198 0.7882353
## 2     2 0.7835894 0.7764706
## 3     3 0.7664598 0.8352941
## 4     4 0.7868014 0.7176471
## 5     5 0.7749245 0.7764706
## 6     6 0.7746155 0.7882353
## 7     7 0.7830136 0.7619048
## 8     8 0.7788020 0.7857143
## 9     9 0.7763899 0.8333333
## 10   10 0.7781581 0.7500000
```

Si queremos calcular la media de todos los folds:

```
# para calcular la media

LDA.train <- mean(results.LDA$train)
LDA.test <- mean(results.LDA$test)
LDA.train
```

```
## [1] 0.7775874
```

```
LDA.test
```

```
## [1] 0.7813305
```

Se han obtenido mejores resultados en test que en train, aunque comparando con las tablas proporcionadas, para este dataset se ha obtenido la misma precisión.

Utilizar el algoritmo QDA para clasificar

Puesto que las asunciones para QDA son las mismas que para LDA excepto por la asunción de varianzas homogéneas, de la misma forma para LDA aunque no se superen todas las suposiciones, aplicamos el algoritmo igualmente y analizamos los resultados.

Para implementarlo, se ha utilizado la misma función que para LDA solo que cambiando el modelo

```
# # función para QDA
run_qda_fold <- function(nfolds, x){
  set.seed(1)
  accuracy.df <- data.frame('fold'=1:nfolds, 'train'=1:nfolds, 'test'=1:nfolds)
  for (i in 1:10){
    file <- paste(x, "-10-", i, "tra.dat", sep="")
    x_tra <- read.csv(file, comment.char="@", header=FALSE)
    file <- paste(x, "-10-", i, "tst.dat", sep="")
    x_tst <- read.csv(file, comment.char="@", header=FALSE)
```

```
    In <- length(names(x_tra)) - 1
    names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
    names(x_tra)[In+1] <- "Y"
    names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
    names(x_tst)[In+1] <- "Y"

    TrainData <- x_tra %>% select(-Y)
    TrainClasses <- x_tra %>% pull(Y)

    qdaModel <- train(TrainData, TrainClasses,
                      method = "qda", preProc = c("center", "scale"),
                      metric="Accuracy", tuneLength = 10)

    qdaPred <- predict(qdaModel, newdata = x_tst %>% select(-Y))
    cfm <- table(qdaPred, x_tst[,'Y'])


    accuracy.df[i,'train']<- qdaModel$results$Accuracy
    accuracy.df[i,'test']<- sum(diag(cfm))/length(x_tst[,'Y'])
  }
  accuracy.df
}
```

LLamamos a la función y visualizamos la precisión por fold.

```
results.QDA <- run_qda_fold(10, nombre)
results.QDA
```

```
##    fold     train      test
## 1     1 0.8240908 0.8823529
## 2     2 0.8294758 0.8705882
## 3     3 0.8359419 0.8705882
## 4     4 0.8375482 0.8352941
## 5     5 0.8412518 0.8235294
## 6     6 0.8325722 0.8352941
## 7     7 0.8482480 0.7857143
## 8     8 0.8319249 0.8571429
## 9     9 0.8327603 0.8690476
## 10   10 0.8361569 0.8928571
```

Calculamos la media de todos los folds.

```
# para calcular la media
QDA.train <- mean(results.QDA$train)
QDA.test <- mean(results.QDA$test)
QDA.train
```

```
## [1] 0.8349971
```

```
QDA.test
```

```
## [1] 0.8522409
```

Otra vez se ha obtenido una precisión mayor en los subconjuntos de “test” que en los de “train”. En cuanto a la comparativa entre LDA y QDA, QDA ha obtenido mejores resultados para este dataset, esto se puede deber a la suposición que se tiene en cuenta para LDA sobre la homogeneidad de las varianzas, ya que para QDA no es necesario y en este dataset tal y como hemos comprobado no teníamos varianzas homogéneas

para ninguna variable.

Comparativa de los tres algoritmos de clasificación

Para esta última parte, el objetivo es realizar un estudio comparativo de los distintos algoritmos de clasificación. Para ello se hará uso de las tablas proporcionadas con los resultados de los algoritmos para distintos datasets.

En primera instancia cargamos las tablas:

```
#leemos la tabla con la precisión media de test
resultados <- read.csv("input/Tablas/clasif_test_alumos.csv")
tablatst <- cbind(resultados[,2:dim(resultados)[2]])
colnames(tablatst) <- names(resultados)[2:dim(resultados)[2]]
rownames(tablatst) <- resultados[,1]

#leemos la tabla con la precisión media de entrenamiento
resultados <- read.csv("input/Tablas/clasif_train_alumnos.csv")
tablatra <- cbind(resultados[,2:dim(resultados)[2]])
colnames(tablatra) <- names(resultados)[2:dim(resultados)[2]]
rownames(tablatra) <- resultados[,1]
```

Lo siguiente que debemos hacer es normalizar ambas tablas para poder aplicar el test de Wilcoxon.

```
# Normalizamos las tablas con el código propuesto
##TABLA NORMALIZADA - para WILCOXON
# + 0.1 porque wilcox R falla para valores == 0 en la tabla
# train
difs <- (tablatra[,1] - tablatra[,2]) / tablatra[,1]
wilc_1_2.tra <- cbind(ifelse (difs<0, abs(difs)+0.1, 0+0.1), ifelse (difs>0,    abs(difs)+0.1, 0+0.1))
colnames(wilc_1_2.tra) <- c(colnames(tablatra)[1], colnames(tablatra)[2])
head(wilc_1_2.tra)
```

```
##      out_train_knn out_train_lda
## [1,]     0.1000000     0.1021667
## [2,]     0.2824899     0.1000000
## [3,]     0.1000000     0.1309740
## [4,]     0.1000000     0.1514882
## [5,]     0.1000000     0.2511537
## [6,]     0.1000000     0.1353018
```

```
# Test
difs.tst <- (tablatst[,1] - tablatst[,2]) / tablatst[,1]
wilc_1_2.tst <- cbind(ifelse (difs<0, abs(difs)+0.1, 0+0.1), ifelse (difs>0,    abs(difs)+0.1, 0+0.1))
colnames(wilc_1_2.tst) <- c(colnames(tablatst)[1], colnames(tablatst)[2])
head(wilc_1_2.tst)
```

```
##      out_test_knn out_test_lda
## [1,]    0.1000000    0.1021667
## [2,]    0.2824899    0.1000000
## [3,]    0.1000000    0.1309740
## [4,]    0.1000000    0.1514882
## [5,]    0.1000000    0.2511537
## [6,]    0.1000000    0.1353018
```

Aplicamos el test de Wilcoxon para comparar el modelo de LDA con k-nn. Primero lo haremos sobre el subconjunto de entrenamiento.

```
#Aplicación del test de WILCOXON
# subconjunto de train
LDAvsKNNtra <- wilcox.test(wilc_1_2.tra[,1], wilc_1_2.tra[,2], alternative = "two.sided", paired=TRUE)
Rmas <- LDAvsKNNtra$statistic
pvalue <- LDAvsKNNtra$p.value
LDAvsKNNtra <- wilcox.test(wilc_1_2.tra[,2], wilc_1_2.tra[,1], alternative = "two.sided", paired=TRUE)
Rmenos <- LDAvsKNNtra$statistic
Rmas
```

```
##  V
## 94
```

```
Rmenos
```

```
##   V
## 116
```

```
pvalue
```

```
## [1] 0.7011814
```

Con un p-valor de 0.7011814 no podemos afirmar de que haya diferencias estadísticamente significativas entre ambos algoritmos.

Para el subconjunto de "test":

```
# subconjunto de test
LDAvsKNNtst <- wilcox.test(wilc_1_2.tst[,1], wilc_1_2.tst[,2], alternative = "two.sided", paired=TRUE)
Rmas <- LDAvsKNNtst$statistic
pvalue <- LDAvsKNNtst$p.value
LMvsKNNtst <- wilcox.test(wilc_1_2.tst[,2], wilc_1_2.tst[,1], alternative = "two.sided", paired=TRUE)
Rmenos <- LDAvsKNNtst$statistic
Rmas
```

```
##  V
## 94
```

```
Rmenos
```

```
##  V
## 94
```

```
pvalue
```

```
## [1] 0.7011814
```

Se obtiene el mismo resultado que en el subconjunto de entrenamiento. Por lo que no podemos afirmar que haya diferencias estadísticamente significativas entre ambos algoritmos.

El último paso sería comparar estos dos algoritmos junto al algoritmo QDA (cuyos resultados tenemos en las tablas) aplicando el test de Friedman. Se aplicará a ambos subconjuntos.

```
# Aplicamos test de Friedman para comparar los tres algoritmos.
#Aplicación del test de Friedman
# Para train
test_friedman.tra <- friedman.test(as.matrix(tablatra))
test_friedman.tra
```

```
##
##  Friedman rank sum test
##
```

```
## data:  as.matrix(tablatra)
## Friedman chi-squared = 1.3, df = 2, p-value = 0.522
```

```
# para test
test_friedman.tst <- friedman.test(as.matrix(tablatst))
test_friedman.tst
```

```
##
##  Friedman rank sum test
##
## data:  as.matrix(tablatst)
## Friedman chi-squared = 0.7, df = 2, p-value = 0.7047
```

Obtenemos para ambas tablas un p-valor alto (0.522 para train y 0.7047 para test), que nos indica de que no tenemos suficiente evidencia estadística para considerar que al menos dos de los algoritmos son diferentes entre ellos. Para obtener las comparativas, aplicamos pst-hoc de Holm.

```
#Aplicación del test post-hoc de HOLM
# train
tam.tra <- dim(tablatra)
groups.tra <- rep(1:tam.tra[2], each=tam.tra[1])
pairwise.wilcox.test(as.matrix(tablatra), groups.tra, p.adjust = "holm", paired = TRUE)
```

```
##
##  Pairwise comparisons using Wilcoxon signed rank exact test
##
## data:  as.matrix(tablatra) and groups.tra
##
##   1    2
## 2 0.65 -
## 3 0.59 0.53
##
## P value adjustment method: holm
```

```
# test
tam.tst <- dim(tablatst)
groups.tst <- rep(1:tam.tst[2], each=tam.tst[1])
pairwise.wilcox.test(as.matrix(tablatst), groups.tst, p.adjust = "holm", paired = TRUE)
```

```
##
##  Pairwise comparisons using Wilcoxon signed rank exact test
##
## data:  as.matrix(tablatst) and groups.tst
##
##   1    2
## 2 1.00 -
## 3 0.53 1.00
##
## P value adjustment method: holm
```

Al obtener un p-valor alto en las comparativas tanto de “test” como de “train”, no podemos asegurar que haya diferencias estadísticamente significativas entre estos algoritmos de clasificación.