

Trabajo de Introducción a la Ciencia de Datos

Lidia Sánchez Mérida

1. Apartado de Regresión

El dataset con el que se procede a trabajar para este primer apartado de aplicación de técnicas de Regresión se denomina **abalone**. En primer lugar, buscamos información sobre sus datos para descubrir sus características principales y composición. Según la descripción que proporciona Kaggle, se trata de un conjunto de **medidas físicas realizadas sobre un tipo de molusco para determinar su edad**. Por otro lado, en la página de UCI Machine Learning Repository se detallan los siguientes aspectos:

- Dispone de un total de **4177 registros y 9 variables** de diferente naturaleza:

Variable	Tipo	Descripción	Valores
Sex	Nominal	Género del ejemplar	M (masculino), F (femenino), I (infante)
Length	Real	Longitud de la concha	Milímetros
Diameter	Real	Ancho de la concha	Milímetros
Height	Real	Altura del ejemplar	Milímetros
Whole weight	Real	Peso total del ejemplar	Gramos
Shucked weight	Real	Peso del ejemplar sin la concha	Gramos
Viscera weight	Real	Peso de las vísceras del ejemplar	Gramos
Shell weight	Real	Peso de la concha sin el ejemplar	Gramos
Rings	Entero	Edad del animal (+1.5)	

- Se trata de un conjunto de datos asociado a un problema de clasificación.
- No contiene valores perdidos.

1.1. Análisis Exploratorio de Datos General

En este primer apartado procedemos a analizar las principales propiedades de este conjunto de datos. Como hemos observado en la descripción anterior, este dataset contiene 4.177 registros y 9 variables. Sin embargo, en el fichero **abalone.dat** la columna que representa el **género ya se encuentra codificada mediante etiquetas numéricas**, por lo que asumimos que la correspondencia con los valores nominales es la siguiente: 1-masculino, 2-femenino y 3-infante.

A continuación, aplicando la función **summary** obtenemos un resumen con las principales medidas estadísticas de cada variable.

```
## [1] 4177      9
##      Sex           Length         Diameter        Height
##  Min.   :1.000   Min.   :0.075   Min.   :0.0550   Min.   :0.0000
##  1st Qu.:1.000  1st Qu.:0.450   1st Qu.:0.3500  1st Qu.:0.1150
##  Median :2.000  Median :0.545   Median :0.4250  Median :0.1400
##  Mean   :1.955  Mean   :0.524   Mean   :0.4079  Mean   :0.1395
##  3rd Qu.:3.000  3rd Qu.:0.615   3rd Qu.:0.4800  3rd Qu.:0.1650
##  Max.   :3.000  Max.   :0.815   Max.   :0.6500  Max.   :1.1300
##      Whole_weight  Shucked_weight  Viscera_weight  Shell_weight
```

```

## Min.    :0.0020  Min.    :0.0010  Min.    :0.0005  Min.    :0.0015
## 1st Qu.:0.4415  1st Qu.:0.1860  1st Qu.:0.0935  1st Qu.:0.1300
## Median :0.7995  Median :0.3360  Median :0.1710  Median :0.2340
## Mean   :0.8287  Mean   :0.3594  Mean   :0.1806  Mean   :0.2388
## 3rd Qu.:1.1530  3rd Qu.:0.5020  3rd Qu.:0.2530  3rd Qu.:0.3290
## Max.   :2.8255  Max.   :1.4880  Max.   :0.7600  Max.   :1.0050
##          Ring
## Min.    : 1.000
## 1st Qu.: 8.000
## Median : 9.000
## Mean   : 9.934
## 3rd Qu.:11.000
## Max.   :29.000

```

El primer aspecto interesante ocurre en la variable **Sex** puesto que según sus cuartiles, en el primer intervalo se encuentran mayoritariamente ejemplares masculinos (etiqueta 1), mientras que en el último intervalo destacan los denominados infantes (etiqueta 3). Por otro lado, podemos apreciar que el ejemplar intermedio es femenino puesto que la mediana es 2. Esta distribución puede indicarnos que **esta variable de clase está balanceada** y existe, aproximadamente, el mismo número de ejemplares para cada categoría.

En relación a las dimensiones de la concha, podemos observar que no existe demasiada diferencia entre los respectivos cuartiles de las variables **Length** y **Diameter**. Además, sus medias y medianas son valores más cercanos al máximo, por lo que podemos intuir que existe una **concentración de ejemplares con una concha de mayor envergadura**. Sin embargo, este hecho contrasta con los valores de la variable **Height**, en los cuales destacan los siguientes puntos:

1. En primer lugar me resulta extraño que existan **ejemplares con una altura de 0 milímetros**, incluyendo tanto la concha como el animal en el interior. No se conoce cómo se han realizado las mediciones y por lo tanto si podrían ser datos erróneos, pero este tipo de valores parecen ser interesantes para futuros estudios de valores anómalos.
2. Por otro lado, la diferencia entre los cuartiles Q1 y Q3 es aún más reducida que en los casos anteriores, lo que significa que existe una **concentración de ejemplares de baja altura**. Si además observamos la media y la mediana, también son valores muy cercanos al mínimo.

En el caso de las medidas relativas a los diferentes tipos de pesos, parece haber una mayor dispersión de valores puesto que existen mayores diferencias entre sus respectivos cuartiles. No obstante, también podemos observar que sus medias y medianas son valores más cercanos al límite inferior de sus respectivos rangos de valores. Esto nos indica que **la mayoría de ejemplares no son pesados**, lo cual puede contrastar con las elevadas dimensiones de las conchas que hemos observado anteriormente.

Finalmente, cabe destacar que en la variable independiente **Ring** tampoco existe una diferencia considerable entre sus cuartiles, y su media y mediana son valores cercanos al mínimo, por lo que parece que en el conjunto de datos hay una **mayor representación de ejemplares jóvenes**.

1.1.1. Hipótesis Iniciales

En función del resumen estadístico anterior, procedemos a plantear un conjunto de hipótesis iniciales acerca de las muestras, los predictores y la variable independiente.

1. En primer lugar, parece que existe un **número similar de ejemplares para cada género**, por lo que las clases de la variable **Sex** estarían balanceadas. Creo que esta variable puede ayudar a determinar los ejemplares más jóvenes gracias a la categoría infante, puesto que es lógico pensar que los **ejemplares más jóvenes tendrán características físicas diferentes** a los adultos. Sin embargo, no tengo claro qué papel podría jugar para el resto de clases.
2. Existe una **concentración de ejemplares con conchas de gran tamaño** según los valores estadísticos de las variables **Length** y **Diameter**. Estos predictores pueden influir en la variable independiente

puesto que es lógico que un **ejemplar aumente su envergadura conforme aumente su edad**. Asimismo, también pienso que ambas variables van a estar relacionadas ya que también es lógico pensar que **a mayor longitud de la concha, mayor diámetro**, por lo que deberemos analizar sus coeficientes de correlación y seleccionar aquella que maximice la variabilidad de información para construir los modelos.

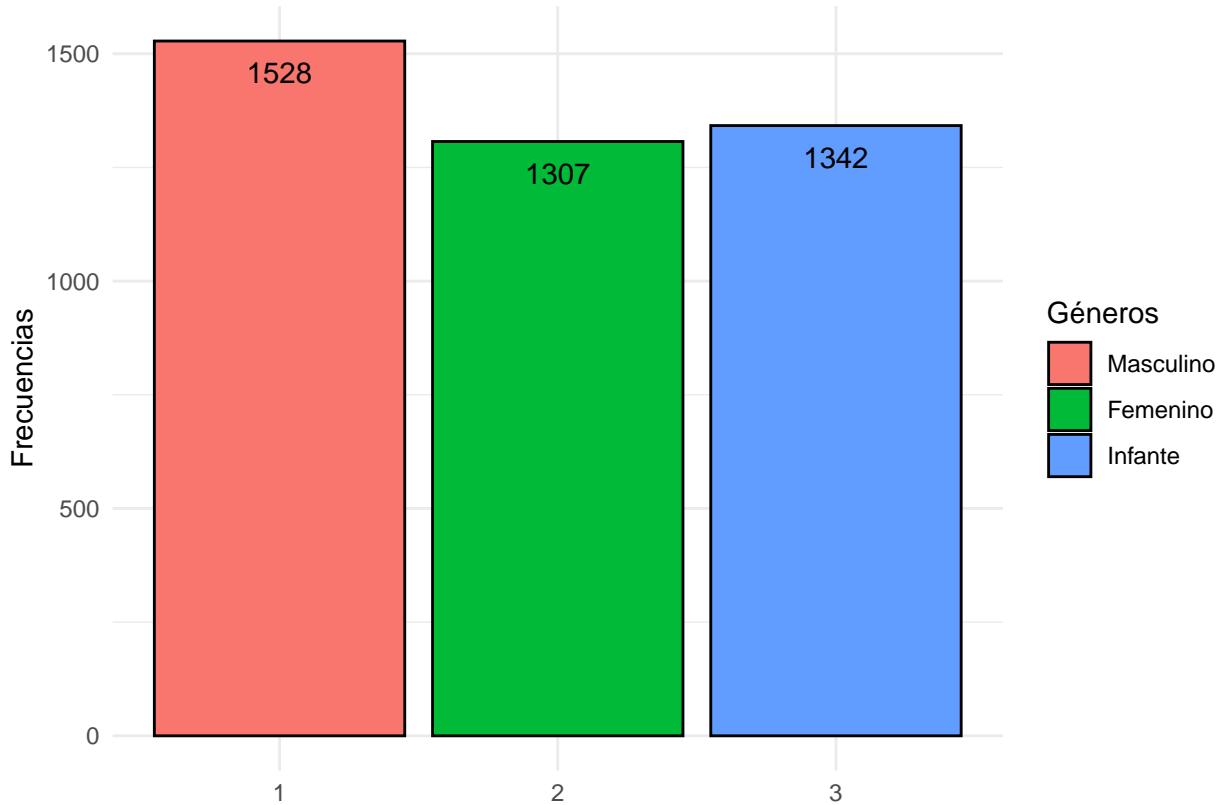
3. En la variable **Height** se observa una concentración de **ejemplares con alturas extremadamente bajas, incluso de 0 milímetros**. Por un lado, me parece extraño este último dato puesto que, independientemente de la edad, no me parece que tenga sentido un valor de 0 en una medida física. Por otro lado, también me extraña la cantidad de **ejemplares de poca altura, en comparación con los considerables tamaños de sus conchas**. A mi parecer, en el primer caso podemos encontrarnos con medidas erróneas, mientras que con respecto a la segunda observación, pienso que este tipo de especie puede estar caracterizada por tener un grosor muy fino. Sin embargo, no consigo visualizar la importancia que puede jugar esta variable en la predicción de la edad de esta especie.
4. Las diferentes medidas de peso disponen de un rango más amplio de valores, con una **concentración de ejemplares de poco peso**. Este hecho me resulta extraño puesto que, como se ha comentado anteriormente, el tamaño de sus conchas es considerablemente grande, por lo que **cabría esperar que sus pesos también fueran mayores**. Por otro lado, creo que pueden existir las siguientes relaciones entre este tipo de predictores.
 - Todos los pesos particulares pueden estar relacionados linealmente con la variable que representa el peso total **Whole_weight**, por lo que puede ser lógico pensar que conforme **aumenten los pesos específicos, también aumente el peso general**. Esta teoría se podrá confirmar visualizando sus coeficientes de correlación.
 - Las variables **Shucked_weight** y **Viscera_weight** pueden estar relacionadas linealmente, puesto que me parece lógico pensar que si **aumenta el peso de los órganos de un ejemplar, también aumenta su peso corporal**. De igual modo, esta hipótesis podrá comprobarse mediante un gráfico de correlaciones.
5. Finalmente, cabe destacar que parece haber una **mayor representación de ejemplares jóvenes**. Este hecho puede suponer un problema en caso de que las características físicas varíen en función de la edad, puesto que si existen pocos ejemplares de mayor edad, los modelos pueden no aprender suficientemente bien los rasgos característicos de cada generación como para identificar las futuras muestras.

1.2. Análisis Exploratorios Univariantes

El objetivo de esta sección consiste en realizar un estudio detallado de cada variable dependiendo de la naturaleza de sus valores. En el caso de las **variables nominales**, se pretende analizar el **balanceamiento de sus clases** a través de gráficos de barras que representen la frecuencia de cada categoría. Mientras que para **variables escalares** se realizarán **análisis estadísticos** sobre la varianza, desviación típica, grado de asimetría, tipo de curtosis y distribución.

1.2.1. Variables nominales

En el caso de este dataset, la **única variable nominal es Sex**, por lo tanto se pretende mostrar el número de ejemplares de cada uno de los géneros disponibles: masculino (1), femenino (2) e infantil (3). Nuestra **hipótesis inicial afirma que las clases están balanceadas**, por lo que el número de muestras de cada categoría no debería de variar demasiado. En el siguiente gráfico podemos observar que la **hipótesis es cierta**, puesto que existen 1528 muestras masculinas, 1307 femeninas y 1342 ejemplares infantiles, es decir, las diferentes categorías disponen de una representación razonablemente equivalente.



1.2.2. Variables numéricas

A continuación se calculan las medidas estadísticas mencionadas anteriormente para las variables numéricas de este dataset. Para la varianza se utiliza la función `var` aplicada a cada uno de los predictores, mientras que para la desviación típica, grado de asimetría y curtosis se hace uso de la **función `describe` del paquete psych**.

Tras analizar los resultados, podemos observar que los predictores disponen de una **varianza mínima** puesto que sus valores se encuentran muy cercanos a 0. Este hecho nos indica que existe **muy poca variabilidad** en sus datos. En combinación con esta medida estadística, podemos observar que la **desviación estándar también es bastante baja**, lo cual indica que los valores de los predictores se encuentran **muy cercanos a la media**. La única excepción es la variable independiente `Ring`, cuya varianza y desviación típica son más elevadas ya que existen ejemplares de diferentes edades.

En relación con la forma de la distribución, la mayoría de variables disponen de un **grado de asimetría superior a 0.5**. Este hecho nos indica que tanto los predictores como la variable independiente tienen desiguales concentraciones de valores en zonas específicas de sus intervalos. En particular, todas las **medidas relativas al peso se encuentran encoladas a la derecha** puesto que su grado de asimetría es positivo. Mientras que las **medidas relativas al tamaño de la concha, como Length y Diameter, están encoladas a la izquierda** por sus coeficientes negativos. Los casos de la **variable independiente y del predictor Height** destacan sobre los restantes puesto que sus coeficiente son mayores que 1, lo cual indica que están caracterizadas por una **fuerte asimetría**, especialmente la variable dependiente.

Situaciones similares ocurren con los **coeficientes de curtosis**, que indican la forma de las colas asociadas a las distribuciones. La mayoría de estos valores se encuentran en el intervalo [-0.05, 0.6], lo cual afirma que muchos predictores disponen de concentraciones menores en algunos extremos, pero al tratarse de **valores cercanos a 0 podemos intuir que siguen una distribución similar a la Gaussiana**. Sin embargo, esto no es aplicable ni a la variable `Height`, cuyo **coeficiente es muy elevado**, ni al término independiente `Ring` cuyo **valor también se aleja bastante de 0**. En el primer caso, la distribución del predictor parece estar caracterizada por una considerable concentración de valores en un intervalo muy reducido, mostrando

un gran pico, mientras que para la variable `Ring` dicha acumulación de datos no será tan masiva pero también seguirá una forma similar.

Finalizando este estudio, a continuación procedemos a comprobar si siguen una **distribución normal**. Para ello vamos a experimentar con dos metodologías diferentes, una gráfica y otra estadística. En el primer caso vamos a hacer uso de los **gráficos QQ únicamente sobre las variables Height y Ring**, puesto que son las más peculiares según los resultados anteriores. En siguiente gráfico podemos apreciar que en el caso de la variable `Height`, sus cuantiles teóricos y empíricos coinciden prácticamente al 100%, a excepción de dos casos. Como consecuencia, este predictor parece que **gráficamente puede seguir una distribución normal**. No obstante, el caso de la variable independiente es diferente, puesto que parece haber una menor coincidencia entre ambas medidas, lo que puede significar que la variable `Ring` no sigue una distribución normal.

Gráfico QQ de Height

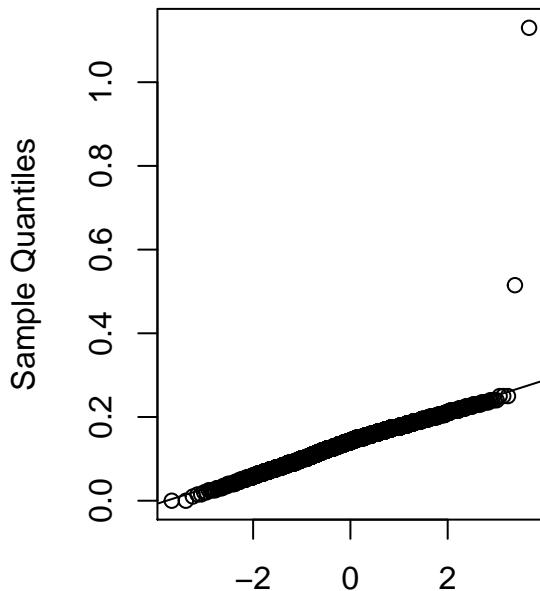
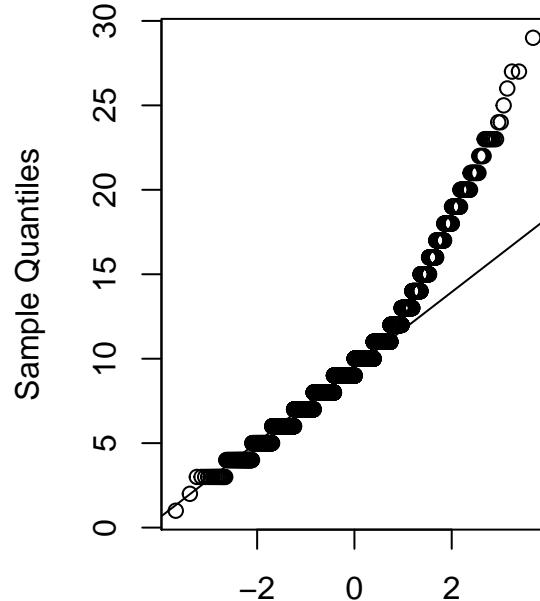


Gráfico QQ de Ring



Theoretical Quantiles

Theoretical Quantiles

Sin embargo, para confirmar la normalidad de cada variable se aplica un test estadístico específico para demostrar esta propiedad: el **test de Shapiro-Wilk's**. La hipótesis nula establece que la variable sigue una distribución normal, frente a la alternativa contraria. Por lo tanto, si el test rechaza, es decir, obtiene un p-valor menor que el umbral=0.05, podemos determinar estadísticamente que la variable no sigue una distribución normal. A continuación se presentan los resultados vinculando cada variable con el p-valor de su respectivo test. Tal y como podemos apreciar, **todos los tests han rechazado la hipótesis nula** puesto que en todos se obtienen p-valores menores que el umbral establecido en 0.05. Esto significa que **ninguna de las variables sigue una distribución normal**, lo cual aumenta la importancia de utilizar tests estadísticos en lugar de métodos gráficos que pueden inducirnos a errores.

```
##          Variables      PValores
## 1        Length 7.442090e-29
## 2       Diameter 1.648335e-28
## 3        Height 1.181265e-47
## 4 Whole_weight 1.013778e-27
## 5 Shucked_weight 9.340986e-32
## 6 Viscera_weight 1.777103e-29
## 7 Shell_weight 1.565014e-28
## 8        Ring 3.259150e-40
```

1.3. Análisis Exploratorio Multivariante

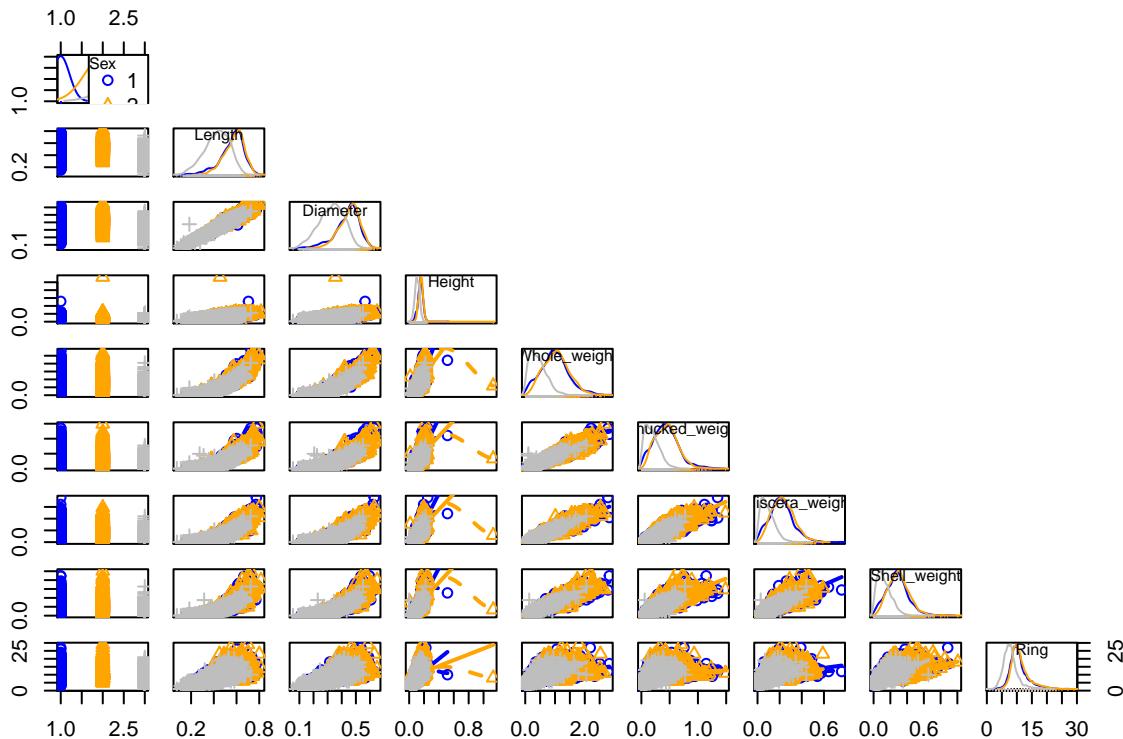
En esta sección se pretende realizar diversos análisis considerando la gran mayoría de variables con el objetivo de obtener una idea general de este conjunto de datos.

1.3.1. Matriz de Puntos

En primer lugar vamos a representar una matriz de puntos con todas las variables disponibles. En la diagonal se ha elegido representar las **curvas de densidad** para graficar sus distribuciones. Aunque su aplicación sobre variables nominales puede no tener sentido, incluimos el predictor **Sex** para distinguir las cualidades físicas de cada categoría. Para ello, aplicamos un **filtro por género** con el que representar cada clase con un color y forma diferente. Así, enriquecemos el gráfico con más información y podremos comenzar a contrastar las hipótesis iniciales.

Si observamos las distribuciones de todas las variables, excepto **Sex**, podemos apreciar que las relativas a los ejemplares masculinos y femeninos son prácticamente idénticas, puesto que se superponen en muchas de las gráficas. Este hecho puede indicarnos que **no existen diferencias físicas entre el género masculino y femenino**. No ocurre lo mismo con el caso de los infantes, ya que sus valores son inferiores en cada una de las medidas con respecto a los otros dos géneros, lo cual ya se había planteado inicialmente.

Por otro lado, si observamos las curvas de densidad de las variables **Length** y **Diameter**, podemos confirmar la teoría de que la mayoría de los ejemplares disponen de unas **conchas muy voluminosas en contraposición con la altura**, puesto que los valores de **Height** están mayormente concentrados al comienzo del intervalo. Esto nos puede indicar que este tipo de molusco dispone de una gran amplitud aunque tiene un grosor fino. Como se ha comentado anteriormente, en el caso de las medidas relativas a los diferentes pesos podemos observar que existe una mayor variabilidad puesto que su rango de valores es más amplio. De igual modo, si observamos las curvas de densidad de la variable independiente **Ring** podemos apreciar que la **mayoría de los ejemplares son bastante jóvenes** ya que el intervalo donde se concentran más es [5, 15].



La segunda parte de este estudio consiste en analizar las posibles asociaciones entre pares de variables. Tal y como podemos observar, parece ser que **la longitud está relacionada linealmente con el diámetro** de la concha, puesto que su gráfica asociada prácticamente representa una recta estrictamente creciente. Por lo que

nuestra teoría se confirma de que a mayor longitud de la concha, mayor diámetro. Una situación similar ocurre con el resto de variables, ya que en todas las gráficas relacionadas con la variable **Length** se puede visualizar una función creciente, siendo de menor intensidad en el caso de la variable **Height**, cuya representación se parece más a una raíz cuadrada. Todas estas conclusiones parecen ser aplicables también a la variable **Diameter** puesto que las gráficas con el resto de variables son muy similares. Continuando con los diferentes pesos también podemos observar que sus vínculos son montónicos, es decir, aumentan simultáneamente, por lo que podemos intuir que durante el desarrollo de los ejemplares sus diferentes **medidas de peso siguen una misma tendencia a la alza**.

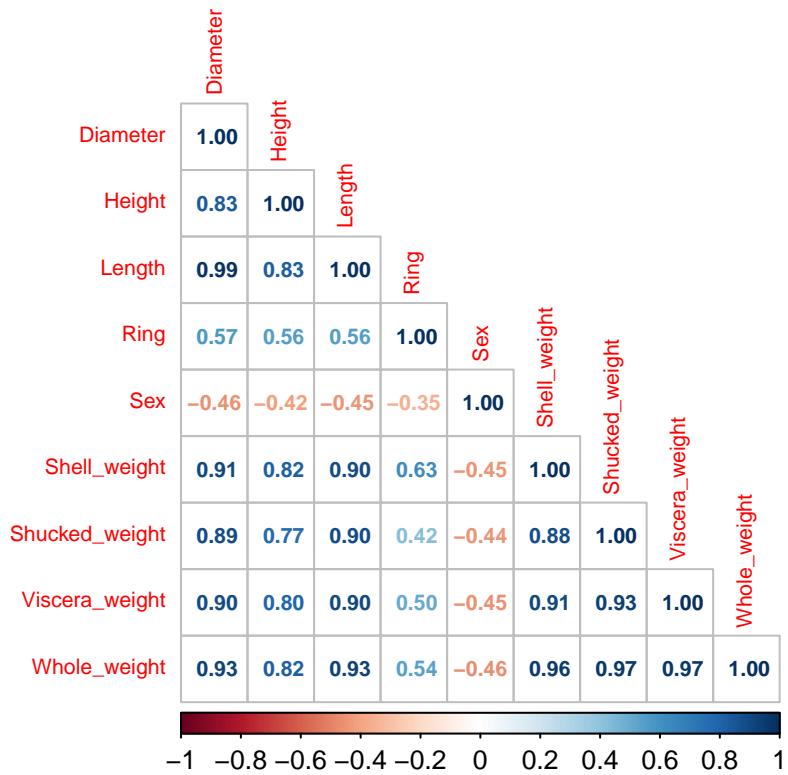
Un caso peculiar lo encontramos entre la variable **Height** y las variables con los diferentes pesos. Si observamos sus gráficas, podemos apreciar un crecimiento muy considerable y abrupto al comienzo del intervalo, es decir, existe una concentración de **ejemplares con una elevada altura y bajos pesos**. Sin embargo, a diferencia del resto de gráficas, en estos casos podemos observar que existen varias muestras que no siguen esta tendencia, ya que sus alturas disminuyen conforme aumentan sus pesos. Esto puede indicar que los ejemplares de mayor edad pierden cualidades físicas, como ocurre en la mayoría de las otras especies animales.

Finalmente, si observamos la variable independiente **Ring** podemos apreciar que es creciente con respecto a la longitud y diámetro, por lo que parece que los ejemplares de **mayor edad, tienen una concha de mayor longitud y diámetro**. Con respecto a los pesos hay una mayor variabilidad, aunque en el caso del peso asociado a la concha sí se aprecia un aumento de sus valores conforme mayor es la edad. Sin embargo, en las restantes variables podemos observar que existen diferencias entre los distintos géneros. Mientras que la tendencia de los infantes en todas las variables de pesos se mantiene creciente, existen ejemplares masculinos y femeninos cuyos tamaños se estabilizan e incluso se reducen conforme aumenta la edad.

1.3.2. Correlaciones

En esta sección se pretende estudiar los coeficientes de correlación de todas las combinaciones de variables. El objetivo es identificar aquellos vínculos relevantes que puedan permitirnos seleccionar los predictores más prometedores para los futuros modelos. Tal y como se aprecia en el siguiente gráfico, existen una gran cantidad de variables correladas.

- Como había comentado anteriormente, se confirma la teoría de que la **variable Length y Diameter están fuertemente correladas** puesto que ambas medidas son relativas al tamaño de la concha. Esta asociación nos permite seleccionar a una de las dos variables para construir los modelos. Del mismo, sendos predictores se encuentran también **fuertemente correladas con las diferentes medidas de peso**. Por lo tanto, parece que al ser medidas relativas al físico de los ejemplares, comparten cierto grado de información.
- Por otro lado, parece que la **contribución de la variable Sex al resto de variables no es demasiado relevante**, ya que sus coeficientes de correlación están por debajo de 0.5. Una razón explicativa podría ser que este predictor solo ayuda a identificar a ejemplares muy jóvenes, por lo que no cubre a la mayoría de muestras.

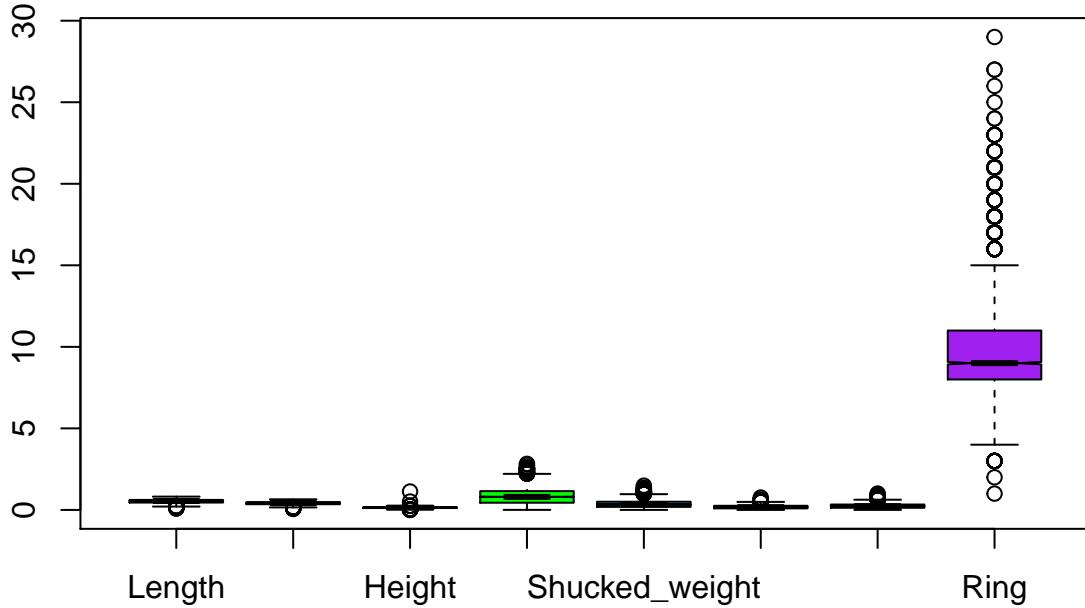


En función de las correlaciones resultantes entre los predictores y la variable independiente, podemos observar que la mayoría de sus coeficientes demuestran una relación moderada, a excepción de `Shell_weight`. Esta variable es la que presenta un valor más alto de 0.63 por lo que parece que es el predictor **más relacionado con el término independiente**. Como consecuencia, se postula como uno de los predictores que más probabilidad tiene de ayudar en la construcción de los modelos.

1.3.3. Diagramas de cajas

Finalizando el estudio multivariante, vamos a representar un diagrama de cajas con todas las variables numéricas para graficar sus dispersiones de datos y comprobar las hipótesis anteriores. Tal y como podemos apreciar, la mayoría de predictores tienen una longitud mínima, lo que significa que sus datos disponen de una **muy baja variabilidad**. Asimismo, también parece que sus **intervalos de valores son muy similares**, por lo que no existen variables con escalas predominantes que puedan afectar a los modelos. Otra conclusión que se puede extraer es que todos los predictores disponen de **outliers moderados**. Se trata de puntos exteriores muy cercanos a las cajas, por lo que parece que no son valores extremos.

Sin embargo, en el caso de la variable independiente `Ring` podemos observar que su **dispersión es mayor**, puesto que su correspondiente caja tiene una mayor longitud. Este hecho indica que dispone de una mayor variabilidad en sus datos, lo que confirma nuestra teoría inicial. Igualmente, también se confirma la hipótesis de que existe una **concentración de ejemplares jóvenes** de una edad entre 5 y 15, y puede ser la razón por la que aparece un mayor número de outliers en las otras franjas de edad. Observándolos, podemos contar que el segundo grupo más representado es el que tiene una edad a partir de 15, y los que menos aparecen son aquellos que se encuentran en el intervalo [0, 5].



1.4. Modelos de Regresión

En este apartado se pretende construir diferentes modelos aplicando diversas técnicas de regresión para intentar maximizar el número de aciertos en la tarea de predecir la variable `Ring` de este dataset.

1.4.1. Regresión Lineal Simple

Comenzamos utilizando la Regresión Lineal Simple entrenando un modelo distinto para cada uno de los cinco predictores más prometedores. Según los estudios anteriores, la variable `Shell_weight` es la que dispone de un coeficiente de correlación mayor con respecto `Ring`. Por lo tanto, el primer modelo estará compuesto por este predictor en solitario. Como podemos apreciar en los resultados, el p-valor del predictor es menor que el umbral 0.05 y dispone de un alto nivel de significación, por lo que la variable `Shell_weight` resulta útil para la predicción de `Ring`. Sin embargo, el valor de R^2 ajustado indica que este modelo solo es capaz de explicar un 39% aproximadamente de los datos, mientras que tiene asociado un RSE de 2.51. Como consecuencia, parece que será necesario añadir más variables con el fin de mejorar su capacidad de generalización.

```
##
## Call:
## lm(formula = Ring ~ Shell_weight, data = abalone.df)
##
## Residuals:
##    Min     1Q   Median     3Q    Max
## -5.9830 -1.6005 -0.5843  0.9390 15.6334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.46212   0.07715  83.76 <2e-16 ***
## Shell_weight 14.53568   0.27908  52.08 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.51 on 4175 degrees of freedom
## Multiple R-squared:  0.3938, Adjusted R-squared:  0.3937
## F-statistic: 2713 on 1 and 4175 DF,  p-value: < 2.2e-16
```

A continuación, procedemos a entrenar un **segundo modelo con la variable Diameter**, puesto que ha sido otro de los predictores que están más claramente asociados con la variable independiente. Sus resultados son muy similares a los del modelo anterior, puesto que el nivel de significación estadística del predictor **Diameter** es muy alto. Sin embargo, en este modelo el **RSE aumenta ligeramente** mientras que el valor de **R² ajustado disminuye** hasta un 33%. Este hecho nos puede indicar que la variable **Diameter** también proporciona información útil para predecir la variable independiente **Ring**, aunque en menor medida que el predictor **Shell_weight**.

```
## 
## Call:
## lm(formula = Ring ~ Diameter, data = abalone.df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -5.1868 -1.6932 -0.7200  0.9066 15.9999 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2.3186    0.1727  13.42   <2e-16 ***
## Diameter     18.6699   0.4115  45.37   <2e-16 ***  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.639 on 4175 degrees of freedom
## Multiple R-squared:  0.3302, Adjusted R-squared:  0.3301 
## F-statistic: 2059 on 1 and 4175 DF,  p-value: < 2.2e-16
```

Para el tercer modelo vamos a utilizar la variable **Length**, que como hemos observado anteriormente, está fuertemente correlada con **Diameter** y con la variable independiente. El objetivo consiste en entrenar un modelo diferente para cada predictor para comparar su resultados y conocer **cuál es el que proporciona más información** con la que predecir la edad de los ejemplares. En comparación con el modelo anterior, podemos observar que este tercer modelo entrenado con **Length** dispone de un **valor mayor de RSE y de R² ajustado**. Por tanto, este predictor aporta menor cantidad de información para la predicción de la variable independiente. Así, en caso de tener que seleccionar una de las dos, parece ser que la variable **Diameter** es la más prometedora para determinar la edad de las muestras de este dataset.

```
## 
## Call:
## lm(formula = Ring ~ Length, data = abalone.df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -5.9665 -1.6961 -0.7423  0.8733 16.6776 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2.1019    0.1855  11.33   <2e-16 ***
## Length      14.9464   0.3452  43.30   <2e-16 ***  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.679 on 4175 degrees of freedom
## Multiple R-squared:  0.3099, Adjusted R-squared:  0.3098 
## F-statistic: 1875 on 1 and 4175 DF,  p-value: < 2.2e-16
```

El cuarto modelo se encuentra compuesto de la variable **Height**, una de las más peculiares de este dataset,

según hemos podido comprobar. Como se ha observado en los diferentes análisis, este predictor dispone de una distribución más extraña, al compararla con el resto, y aporta información general al tamaño del ejemplar considerando tanto el animal como su concha. Tal y como se aprecia en el resumen del modelo, el predictor **Height** no mejora ni la explicabilidad de la variable independiente ni el RSE, incluso se puede observar un **ligero empeoramiento de las medidas** de calidad. Este hecho nos indica que aporta menos información útil para la predicción de la variable **Ring** que en los casos anteriores.

```
##
## Call:
## lm(formula = Ring ~ Height, data = abalone.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -44.496  -1.657  -0.607   0.839  17.112
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.9385    0.1443   27.30 <2e-16 ***
## Height      42.9714    0.9904   43.39 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.677 on 4175 degrees of freedom
## Multiple R-squared:  0.3108, Adjusted R-squared:  0.3106
## F-statistic:  1882 on 1 and 4175 DF,  p-value: < 2.2e-16
```

Para el quinto y último modelo el predictor participante es **Whole_weight**, una medida de peso que involucra al cuerpo y a la concha del ejemplar. Es otra de las variables que disponen de una correlación moderada con respecto a la variable independiente, aunque en menor medida que los predictores anteriores, por lo que no esperamos que produzca una considerable mejoría. Tal y como se comentaba anteriormente, **esta variable no está tan fuertemente asociada** a la variable independiente y por ello su RSE aumenta, mientras que el valor de R^2 ajustado disminuye hasta situarse por debajo del 30%. Por lo tanto, este predictor en solitario tampoco es suficiente como para explicar la mayor parte de la variabilidad de los datos.

```
##
## Call:
## lm(formula = Ring ~ Whole_weight, data = abalone.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.2693 -1.7518 -0.6874  1.0177 15.7029
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.98924    0.08244   84.78 <2e-16 ***
## Whole_weight 3.55291    0.08562   41.50 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.713 on 4175 degrees of freedom
## Multiple R-squared:  0.292, Adjusted R-squared:  0.2919
## F-statistic:  1722 on 1 and 4175 DF,  p-value: < 2.2e-16
```

Finalmente, las conclusiones que podemos extraer de los análisis de modelos entrenados con Regresión Lineal Simple se detallan a continuación:

1. El modelo con un **menor RSE** y un **mayor valor de R^2 ajustado** ha sido el primero en el que intervenía únicamente la **variable Shell_weight**, la cual estaba **más fuertemente correlada con la variable independiente Ring**. Por lo tanto, se trata de un predictor que aporta suficiente información como para alcanzar por sí solo casi un 40% de la explicabilidad de los datos.
2. Pese a la extraña distribución y a su menor grado de correlación con la variable independiente, el **predictor Height es el que ha proporcionado el segundo mejor modelo** con un 31% de explicabilidad y un RSE de 2.67. Este hecho nos indica que la altura puede identificar a un tercio de los ejemplares de manera independiente. Con una alta probabilidad, estas muestras pertenecerán a la categoría de infantes, puesto que parece lógico pensar que estarán caracterizados por una menor altura que los adultos.
3. Finalmente, en los modelos con múltiples predictores deberemos de **eliminar la variable Length** debido a su alta correlación con el predictor **Diameter**. Si bien proporcionan el mismo tipo de información, esta última variable ha conseguido obtener un modelo con una mayor explicabilidad y menor error, por lo que su información resulta más útil para la predicción de la edad de esta especie.

1.4.2. Regresión Lineal Múltiple

El objetivo de este apartado consiste en realizar diferentes combinaciones de predictores para entrenar modelos con Regresión Lineal Múltiple y mejorar los resultados del mejor modelo obtenido con Regresión Lineal Simple. Para ello aplicaremos la **técnica backward** en el que comenzamos con un primer modelo en el que se incluyen todos los predictores, para después eliminar aquellos que no sean relevantes.

```
##
## Call:
## lm(formula = Ring ~ ., data = abalone.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5991  -1.3120  -0.3549   0.8968  14.0582
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.1478    0.3013 13.765 < 2e-16 ***
## Sex          -0.3885    0.0467 -8.319 < 2e-16 ***
## Length       -0.8264    1.8122 -0.456   0.648
## Diameter     11.9640    2.2254  5.376 8.02e-08 ***
## Height        11.2045    1.5374  7.288 3.75e-13 ***
## Whole_weight   9.0702    0.7270 12.476 < 2e-16 ***
## Shucked_weight -20.1061   0.8168 -24.617 < 2e-16 ***
## Viscera_weight -10.1551   1.2941 -7.847 5.36e-15 ***
## Shell_weight    8.7011    1.1277  7.716 1.49e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.2 on 4168 degrees of freedom
## Multiple R-squared:  0.5353, Adjusted R-squared:  0.5345
## F-statistic: 600.3 on 8 and 4168 DF,  p-value: < 2.2e-16
```

En el resumen estadístico podemos observar que, considerando todos los predictores, hemos conseguido mejorar la explicabilidad de los datos aumentando el valor de R^2 ajustado hasta un 53%, además de haber reducido el RSE. Sin embargo, podemos apreciar que el **predictor Length tiene asociado un p-valor muy superior** al umbral de 0.05, lo que nos indica que puede ser eliminado sin influir en el modelo. Este hecho ya se había previsto anteriormente si recordamos la fuerte correlación existente entre **Diameter** y **Length**. Por lo tanto, eliminamos esta variable y procedemos a entrenar un segundo modelo.

```

## 
## Call:
## lm(formula = Ring ~ . - Length, data = abalone.df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -10.5714 -1.3106 -0.3511  0.8918 14.0850 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.10391   0.28555 14.372 < 2e-16 ***
## Sex          -0.38954   0.04664 -8.352 < 2e-16 ***  
## Diameter     11.05438   0.98630 11.208 < 2e-16 ***  
## Height       11.18384   1.53662  7.278 4.02e-13 ***  
## Whole_weight  9.07432   0.72691 12.483 < 2e-16 ***  
## Shucked_weight -20.13583  0.81408 -24.734 < 2e-16 *** 
## Viscera_weight -10.20929  1.28847 -7.924 2.94e-15 *** 
## Shell_weight    8.71714   1.12699  7.735 1.29e-14 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 2.2 on 4169 degrees of freedom
## Multiple R-squared:  0.5353, Adjusted R-squared:  0.5345 
## F-statistic: 686.1 on 7 and 4169 DF,  p-value: < 2.2e-16

```

Tal y como se puede observar, este segundo modelo dispone de la **misma calidad** que el anterior solo que es **más sencillo** puesto que contiene un predictor menos. Observando de nuevo las variables, podemos apreciar que todas tienen una alta significación estadística, pero algunas disponen de p-valores más altos. Como el modelo todavía contiene muchos de los predictores iniciales, vamos a tratar de simplificarlo hasta conseguir un error y porcentaje de explicabilidad aceptable. En el tercer modelo vamos a eliminar la variable **Height** puesto que es la que mayor p-valor tiene asociado.

```

## 
## Call:
## lm(formula = Ring ~ . - Length - Height, data = abalone.df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -8.2669 -1.3216 -0.3672  0.8892 13.8981 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.47808   0.28262 15.845 < 2e-16 ***  
## Sex          -0.40557   0.04688 -8.652 < 2e-16 ***  
## Diameter     13.29679   0.94277 14.104 < 2e-16 ***  
## Whole_weight  9.17316   0.73130 12.544 < 2e-16 ***  
## Shucked_weight -20.32816  0.81871 -24.830 < 2e-16 *** 
## Viscera_weight -9.73483  1.29481 -7.518 6.75e-14 *** 
## Shell_weight    9.57291   1.12781  8.488 < 2e-16 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 2.213 on 4170 degrees of freedom
## Multiple R-squared:  0.5294, Adjusted R-squared:  0.5287 
## F-statistic: 781.9 on 6 and 4170 DF,  p-value: < 2.2e-16

```

En este tercer modelo se ha reducido ligeramente el valor de R^2 ajustado y ha aumentado muy levemente el RSE. Sin embargo, al eliminar el predictor **Height** podemos observar cómo los **p-valores de algunas variables han aumentado**, en particular, las dos últimas: **Viscera_weight** y **Shell_weight**. Esto nos indica que la variable **Height** comparte cierta información común con estos dos predictores y al eliminarla, **se revalora su importancia** en el modelo. No obstante, el primero de ellos sigue distinguiéndose del resto por tener un p-valor más bajo. Vamos a entrenar un cuarto modelo eliminándola para comprobar cuál es el comportamiento.

```
## 
## Call:
## lm(formula = Ring ~ . - Length - Height - Viscera_weight, data = abalone.df)
## 
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -7.8091 -1.3446 -0.3757  0.9102 14.7898 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.62349   0.28383 16.289 <2e-16 ***
## Sex          -0.39560   0.04717 -8.387 <2e-16 ***
## Diameter     12.42828   0.94187 13.195 <2e-16 ***
## Whole_weight  5.75862   0.57698  9.981 <2e-16 ***
## Shucked_weight -18.69347  0.79455 -23.527 <2e-16 ***
## Shell_weight  12.39342   1.07064 11.576 <2e-16 ***  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.228 on 4171 degrees of freedom
## Multiple R-squared:  0.523, Adjusted R-squared:  0.5225 
## F-statistic: 914.8 on 5 and 4171 DF,  p-value: < 2.2e-16
```

Como podemos apreciar, mientras que el RSE ha aumentado muy ligeramente el porcentaje de explicabilidad también se ha elevado un poco más. Esto nos indica que la información proporcionada por la **variable eliminada no es lo suficientemente importante** como para mantenerla en el modelo. Así, hemos conseguido un modelo más simplificado y con unas métricas de calidad razonablemente aceptables.

A continuación procedemos a añadir **términos no lineales** para intentar ajustar más el modelo a los datos y mejorar su capacidad de generalización. En el quinto modelo se añade una **interacción entre las variables Shell_weight y Diameter**, con las que se han obtenido dos de los mejores modelos utilizando Regresión Lineal Simple.

```
## 
## Call:
## lm(formula = Ring ~ . - Length - Height - Viscera_weight + Shell_weight * 
##      Diameter, data = abalone.df)
## 
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -8.8245 -1.3210 -0.3353  0.8957 15.9905 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.2272    0.2807 15.059 < 2e-16 ***
## Sex          -0.2987   0.0470 -6.354 2.32e-10 ***
## Diameter     8.1954    0.9872  8.302 < 2e-16 ***  
## Whole_weight  7.1265    0.5776 12.337 < 2e-16 ***
```

```

## Shucked_weight      -18.0521    0.7823 -23.075 < 2e-16 ***
## Shell_weight        34.8044    2.1030 16.550 < 2e-16 ***
## Diameter:Shell_weight -43.5165   3.5360 -12.307 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.189 on 4170 degrees of freedom
## Multiple R-squared:  0.5398, Adjusted R-squared:  0.5391
## F-statistic: 815.1 on 6 and 4170 DF,  p-value: < 2.2e-16

```

Como podemos apreciar en el resumen estadístico, la interacción entre sendas variables provoca la mejora tanto del porcentaje de explicabilidad, que aumenta hasta un 53.9%, mientras que por otro lado produce un ligero decremento del RSE. Esto nos indica que la **combinación de las variables Shell_weight y Diameter** permite un mayor ajuste a los datos, puesto que **potencia la información útil** para predecir la variable Ring. Sin embargo, podemos observar que de forma paralela también **reduce la significación estadística de la variable Sex**. Una posible explicación puede asociarse a que el tamaño y diámetro de la concha también pueden ser características diferenciadoras para los mismos ejemplares a los que la variable Sex ayuda a identificar: los infantes. Resulta lógico pensar que los animales más jóvenes de esta especie poseen conchas de menor tamaño. De hecho, si **eliminamos este predictor** podemos observar cómo en el sexto modelo las **medidas de calidad no se ven demasiado afectadas**, y de esta forma, conseguimos reducir la complejidad.

```

##
## Call:
## lm(formula = Ring ~ . - Length - Height - Viscera_weight - Sex +
##     Shell_weight * Diameter, data = abalone.df)
##
## Residuals:
##    Min      1Q  Median      3Q      Max
## -8.6319 -1.3597 -0.3493  0.8923 15.8480
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  3.3014    0.2411 13.695 <2e-16 ***
## Diameter     8.3733    0.9914  8.446 <2e-16 ***
## Whole_weight  7.4644    0.5779 12.916 <2e-16 ***
## Shucked_weight -18.1437   0.7859 -23.087 <2e-16 ***
## Shell_weight   36.6291   2.0931 17.500 <2e-16 ***
## Diameter:Shell_weight -47.2810   3.5025 -13.499 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.199 on 4171 degrees of freedom
## Multiple R-squared:  0.5353, Adjusted R-squared:  0.5347
## F-statistic: 960.9 on 5 and 4171 DF,  p-value: < 2.2e-16

```

Realizando varios experimentos que no se incluyen en esta memoria para no extenderla demasiado, he podido comprobar que añadiendo términos cuadráticos también se consigue cierta mejora en la calidad del modelo, a costa de añadir complejidad. Un ejemplo representativo es el que se muestra en el séptimo modelo, en el que se añaden los **términos cuadráticos de dos medidas del peso: Shucked_weight y Whole_weight**. Como se puede apreciar, el porcentaje de R² ajustado aumenta a un 55%, mientras que el RSE disminuye cuatro centésimas. Se trata de otra forma con la que poder potenciar la información proporcionada individualmente por estos dos predictores.

```

##
## Call:

```

```

## lm(formula = Ring ~ . - Length - Viscera_weight - Sex + Shell_weight *
##     Diameter + I(Shucked_weight^2) + I(Whole_weight^2), data = abalone.df)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -9.4721 -1.3126 -0.3132  0.9077 16.3520 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2.6238    0.3308   7.931 2.78e-15 ***
## Diameter     10.3002   1.4957   6.887 6.57e-12 ***
## Height       6.3820   1.5371   4.152 3.36e-05 ***
## Whole_weight 13.5881   1.1399  11.920 < 2e-16 ***
## Shucked_weight -35.5435  1.7407 -20.419 < 2e-16 ***
## Shell_weight  34.8012   2.8771  12.096 < 2e-16 ***
## I(Shucked_weight^2) 14.9836   1.3417  11.167 < 2e-16 ***
## I(Whole_weight^2)   -2.5094   0.4019  -6.244 4.68e-10 ***
## Diameter:Shell_weight -44.8286   5.2969  -8.463 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.156 on 4168 degrees of freedom
## Multiple R-squared:  0.5536, Adjusted R-squared:  0.5527 
## F-statistic:  646 on 8 and 4168 DF,  p-value: < 2.2e-16

```

En comparación con el modelo obtenido mediante Regresión Lineal Simple, podemos observar que ha **aumentado la explicabilidad en más de un 16%, mientras que se reduce el error en 0.36 décimas**. Como principal desventaja se encuentra el aumento de la complejidad del modelo, puesto que en el múltiple intervienen cinco predictores, una interacción y dos términos cuadráticos, mientras que en el modelo simple solo dispone de una única variable dependiente.

1.4.3. K Nearest Neighbours (KNN)

En esta sección se pretende entrenar varios modelos utilizando el **algoritmo KNN en combinación con Validación Cruzada**. Para ello, utilizaremos las cinco particiones de entrenamiento y validación en cada una de las iteraciones. El objetivo consiste en analizar si el mejor modelo entrenado con estas técnicas dispone de una mejor capacidad predictora que el mejor modelo entrenado con Regresión Lineal Múltiple. Como podemos observar, el **error obtenido en el conjunto de entrenamiento es mayor que el del mejor modelo con Regresión Lineal Múltiple**. Esto demuestra que no siempre una misma formulación proporciona buenos resultados para algoritmos diferentes, como ha sido el caso de esta comparación entre KNN y Regresión Lineal Múltiple.

```

## [1] 2.188686
## [1] 5.479208

```

1.5. Comparación de algoritmos de Regresión

Para finalizar este apartado de regresión, se realiza una comparación entre el comportamiento de los siguientes algoritmos: **KNN, Regresión Lineal Múltiple y M5**. En primer lugar, aplicamos **validación cruzada** sobre los dos primeros con el objetivo de obtener el error medio de entrenamiento y validación para cada técnica al iterar cinco veces sobre las cinco particiones de este dataset.

```

## [1] 4.81969
## [1] 4.942255

```

```
## [1] 2.217759
## [1] 5.398169
```

A continuación sustituimos los valores anteriores en la línea correspondiente al dataset `abalone` dentro de los ficheros `regr_train_alumnos.csv` y `regr_test_alumnos.csv`. La primera comparación se realiza entre **Regresión Lineal MMúltiple y KNN utilizando el test de Wilcoxon**. El objetivo consiste en conocer si existen diferencias significativas entre sendas técnicas. La hipótesis nula afirma que ambos algoritmos se comportan de forma semejante, mientras que la alternativa determina que no son iguales. Como podemos apreciar, el p-valor resultante de aplicar el test de Wilcoxon es mayor que el umbral 0.05, por lo que nos revela que **no se puede rechazar la hipótesis nula** de que el algoritmo KNN tiene el mismo comportamiento que Regresión Lineal Múltiple sobre los datasets experimentados Así, se puede determinar que solo existe un 23.4% de confianza de que sean distintos.

```
## [1] 0.7987061
```

A continuación, **añadimos el algoritmo M5 a la comparación**. Para ello aplicamos el **test de Friedman**, que extiende el objetivo del test de Wilcoxon pero para más de dos participantes. Tal y como se puede observar en los resultados, el p-valor resultante no es menor que el umbral prefijado de 0.05, por lo que **tampoco se puede rechazar la hipótesis nula de que los tres algoritmos tienen el mismo comportamiento**. Aunque no hay evidencias estadísticas de las diferencias entre las tres técnicas, vamos a agrupar los algoritmos por pares siendo 1:Regresión Lineal, 2:KNN y 3:M5. El objetivo consiste en identificar cuáles de ellos disponen de un comportamiento diferente. Con este objetivo, se aplican **múltiples tests de Wilcoxon aplicando la penalización por pasos de Holm** con la que controlar el error acumulado por realizar varios tests sobre la misma población. Como podemos observar en la tabla comparativa, **ninguno de los tests ha rechazado la hipótesis nula** de que los pares de algoritmos comparados son similares. Por lo tanto, **no existe evidencia estadística** que confirme que existen diferencias entre la Regresión Lineal Múltiple, KNN y M5.

```
##
## Friedman rank sum test
##
## data: as.matrix(tablatst)
## Friedman chi-squared = 5.3333, df = 2, p-value = 0.06948
##
## Pairwise comparisons using Wilcoxon signed rank test
##
## data: as.matrix(tablatst) and groups
##
##    1     2
## 2 0.97 -
## 3 0.27 0.27
##
## P value adjustment method: holm
```

2. Apartado de Clasificación

En este segundo apartado el dataset con que se procede a trabajar se denomina **vehicle**. Como en el caso anterior, comenzamos buscando información sobre sus datos y características principales. Según la descripción del repositorio UCI Machine Learning, este conjunto de datos está compuesto por las **propiedades de imágenes relativas a las siluetas de diferentes tipos de vehículos**. Los aspectos más destacables de este dataset se listan a continuación:

- Dispone de un total de **946 muestras y 19 atributos**. A continuación se presenta la información que he podido encontrar sobre cada uno de ellos.

Variable	Tipo	Descripción
Compactness	Entero	Relación entre el área y el volumen. Una figura compacta suele ser pequeña y redonda.
Circularity, Distance_circularity	Enteros	Relación entre la forma del área y del perímetro.
Radius_ratio	Entero	Relación entre el mínimo y máximo radio de un objeto. Ambos radios se refieren a los puntos más y menos cercanos dentro de una elipse.
Praxis_aspect_ratio, Max_length_aspect_ratio	Enteros	Relación entre el radio mínimo y máximo de un objeto.
Scatter_ratio	Entero	Relación entre la inercia del radio mínimo y máximo.
Elongatedness	Entero	Relación entre el área y el ancho reducido de un objeto.
Praxis_rectangular, Length_rectangular	Enteros	Relación entre el área y el ancho por el alto de un objeto.
Major_variance, Minor_variance	Enteros	Segundo momento del radio mínimo y del radio máximo. Un momento es la captura de una imagen con unas propiedades o interpretaciones específicas.
Gyration_radius	Entero	Relación entre las dos propiedades anteriores.
Major_skewness, Minor_skewness	Enteros	Asimetría de una imagen en ambos ejes.
Minor_kurtosis, Major_kurtosis	Enteros	Forma de la imagen en ambos ejes
Hollows_ratio	Entero	Relación entre el área de los píxeles no activos y el perímetro de una imagen.
Class	Nominal	El tipo de vehículo asociado a la silueta: van, saab, bus, opel

- Suele ser utilizado como un **problema de clasificación** para determinar el tipo de un vehículo a partir de su contorno.
- No se afirma ni se niega la existencia de valores perdidos.

2.1. Análisis Exploratorio de Datos General

Tras cargar el dataset completo a partir del fichero `vehicle.dat`, replicamos el proceso analítico seguido para el apartado anterior. Como ya conocíamos, este conjunto de datos dispone de **845 registros y 19 atributos**, todos numéricos a excepción de la variable `Class` que contiene valores nominales. El primer aspecto destacable de este dataset es que sus predictores disponen de **escalas de valores muy diferentes**. Esta característica se deberá tener en cuenta al construir los modelos de clasificación para que no se vean influenciados por aquellas variables con escalas mayores. La segunda propiedad más relevante es que, como en el apartado anterior, la **variable nominal Class también se encuentra balanceada** puesto que dispone de un número razonablemente equivalente de muestras para cada categoría. Este hecho indica que existe una representación equitativa para cada uno de los vehículos que se pretenden identificar.

Debido a que se trata de un conjunto de datos con un número mayor de variables y orientado a una temática muy particular y desconocida para mí, pese a haber buscado información sobre cada atributo, no puedo

realizar un estudio tan exhaustivo como se ha efectuado con el dataset anterior. No obstante, si nos ceñimos a las medidas estadísticas, podemos observar que la **mayoría de variables disponen de intervalos de valores más amplios**, puesto sus mínimos y máximos valores se encuentran más alejados. Aunque existen algunas excepciones, como las variables **Circularity**, **Praxis_rectangular** y **Major_kurtosis** cuyos rangos de valores son de menor tamaño.

Observando los cuartiles de cada una de las variables, podemos apreciar que en **muchos de los predictores las diferencias entre Q1 y Q3 son mínimas**. Esto indica que existe una baja variabilidad en sus datos y, por ende, una concentración de individuos con valores cercanos en los tres primeros intervalos. Un ejemplo representativo es la variable **Major_skewness**, cuyo mínimo es de 59 y su valor máximo es 135. Sin embargo, su primer cuartil es 67 mientras que su tercer cuartil tiene un valor de 75. Este hecho nos indica que la mayoría de figuras se caracterizan por una asimetría situada en el intervalo [59, 75]. Son pocas las variables que disponen de una mayor dispersión en sus datos, como son los casos de **Radius_ratio**, **Scatter_ratio**, **Gyration_radius** y **Minor_variance**, puesto que la diferencia entre sus primeros y terceros cuartiles es más considerable.

Por otro lado, en la variable **Praxis_aspect_ratio** podemos apreciar que existe una concentración de vehículos cuyos radios se encuentran muy próximos, ya que la diferencia entre sus cuartiles es mínima. Considerando que estos radios parecen pertenecer a la forma de una elipse, podemos intuir que existe una **concentración de siluetas cuya altura y anchura son similares**. Como el dataset se basa en imágenes de vehículos, una posible razón para esta teoría podría hacer referencia a la existencia de una **mayor cantidad de datos sobre coches** que referentes a otros vehículos con medidas físicas más desiguales, como son las caravanas y los autobuses. Esta hipótesis puede apoyarse en dos relaciones más:

1. En la poca dispersión de dos medidas relacionadas con el área y el perímetro de un objeto: **Praxis_rectangular** y **Length_rectangular**.
2. Y también en la poca variabilidad de las tres primeras variables asociadas a la compactación y la forma circular de una silueta.

En el caso de las medidas más familiares, podemos apreciar que tanto el grado de asimetría como de curtosis son menores en el radio mínimo, que conecta los puntos más cercanos, y mayores en el radio máximo. Esta característica parece lógica puesto que es más probable que existan **más concentraciones de datos en un espacio mayor**.

```
## [1] 845 19

##   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.:146.0      1st Qu.:33.00
## Median  : 61.00      Median  : 8.000      Median :157.0      Median :43.00
## Mean    : 61.68      Mean    : 8.566      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
## 1st Qu.:19.00       1st Qu.:137        1st Qu.:167.0     1st Qu.: 318
## Median  :20.00       Median  :146        Median :179.0     Median : 364
## Mean    :20.58       Mean    :148        Mean   :188.6     Mean   : 440
## 3rd Qu.:23.00       3rd Qu.:159        3rd Qu.:217.0     3rd Qu.: 587
```

```

##   Max.    :29.00      Max.    :188      Max.    :320.0     Max.    :1018
##   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 : 72.00    Median : 6.000    Median :11.0
##   Mean   :174.7     Mean   : 72.47    Mean   : 6.378    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    bus  :218
##   1st Qu.:184.0     1st Qu.:190.0    opel:212
##   Median :188.0     Median :197.0    saab:217
##   Mean   :188.9     Mean   :195.6    van  :198
##   3rd Qu.:193.0     3rd Qu.:201.0
##   Max.   :206.0     Max.   :211.0

```

2.1.1. Hipótesis Iniciales

En función de las conclusiones estadísticas extraídas anteriormente, a continuación se plantea un conjunto de hipótesis iniciales sobre los datos, predictores y la variable independiente.

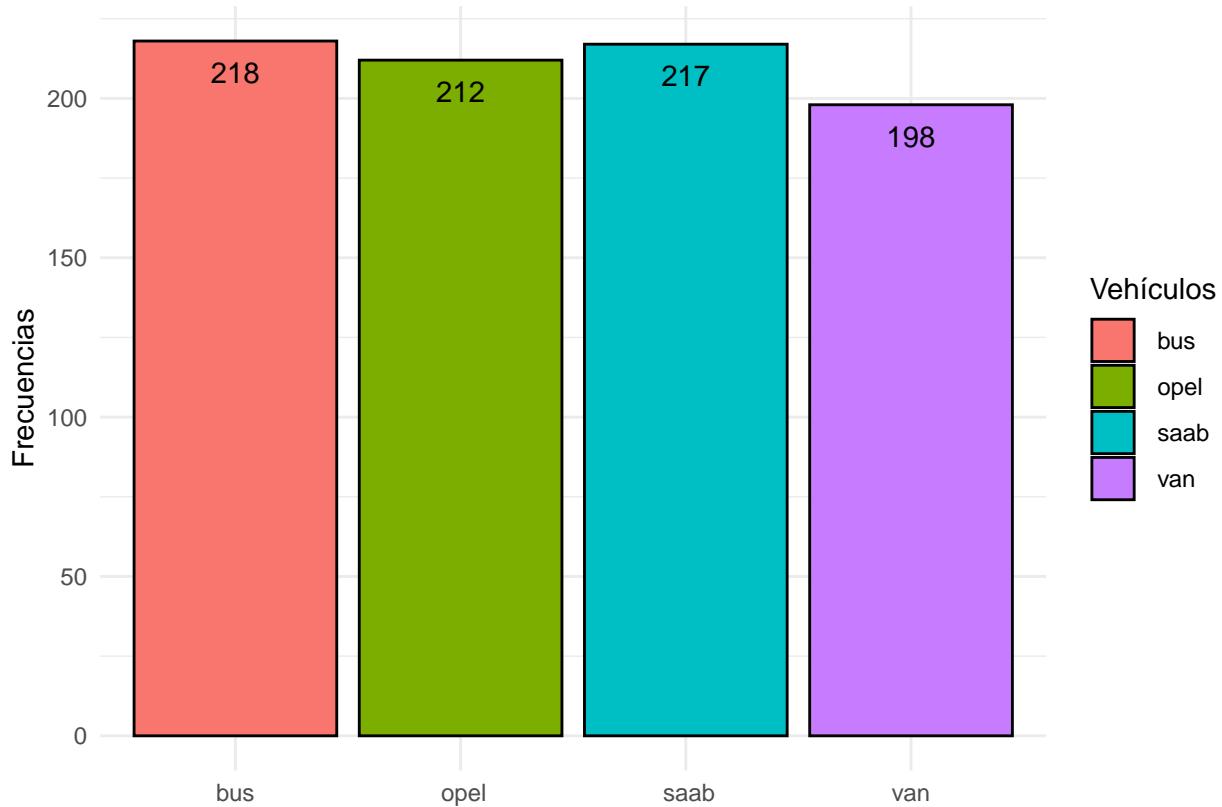
1. Como ocurría en el apartado anterior, en este dataset la variable **Class** dispone de un **número equivalente de muestras para cada tipo de vehículo**, por lo que sus categorías se encuentran balanceadas. Se trata de una gran ventaja puesto que favorece el entrenamiento de modelos **sin introducir el sesgo relativo a una clase predominante**.
2. Mientras que la mayoría de **predictores disponen de intervalos de valores más amplios**, no ocurre lo mismo con los **cuartiles, cuyos valores son más cercanos**. Esto significa que existe una concentración de valores similares en la mayor parte de los rangos de datos, a excepción de algunas muestras que disponen de valores más extremos. Una posible razón para este hecho es que **dos de los cuatro tipos de vehículos disponibles son marcas de automóviles**, por lo que las diferencias entre ambos pueden ser mínimas. Esta hipótesis también podría explicar la **concentración de muestras con pocas diferencias entre su radio mínimo y máximo**.
3. Finalmente, considerando que los atributos de este dataset hacen referencia a las siluetas de diversos vehículos, podemos intuir que aquellas muestras con un **mayor grado de asimetría pueden asociarse a vehículos de mayor tamaño**, como los autobuses y furgonetas. Por lo tanto, esta variable puede jugar un papel fundamental en el entrenamiento de modelos de clasificación.

2.2. Análisis Exploratorios Univariantes

Como en el apartado de regresión, a continuación se pretende realizar un análisis detallado para cada variable dependiendo de su tipo de datos. Para las **variables nominales representaremos la frecuencia de sus categorías** en un gráfico de barras, mientras que para los **predictores numéricos** realizaremos un estudio sobre su **varianza, desviación típica, grado de asimetría, tipo de curtosis y de distribución**.

2.2.1. Variables nominales

Para este dataset solo existe la única variable categórica coincide con la variable independiente que se pretende predecir. Según hemos podido observar en el resumen estadístico, la **variable Class está balanceada**, y tal y como podemos apreciar en el gráfico, dispone de un número muy similar de ejemplos para cada tipo de vehículo.



2.2.2. Variables numéricas

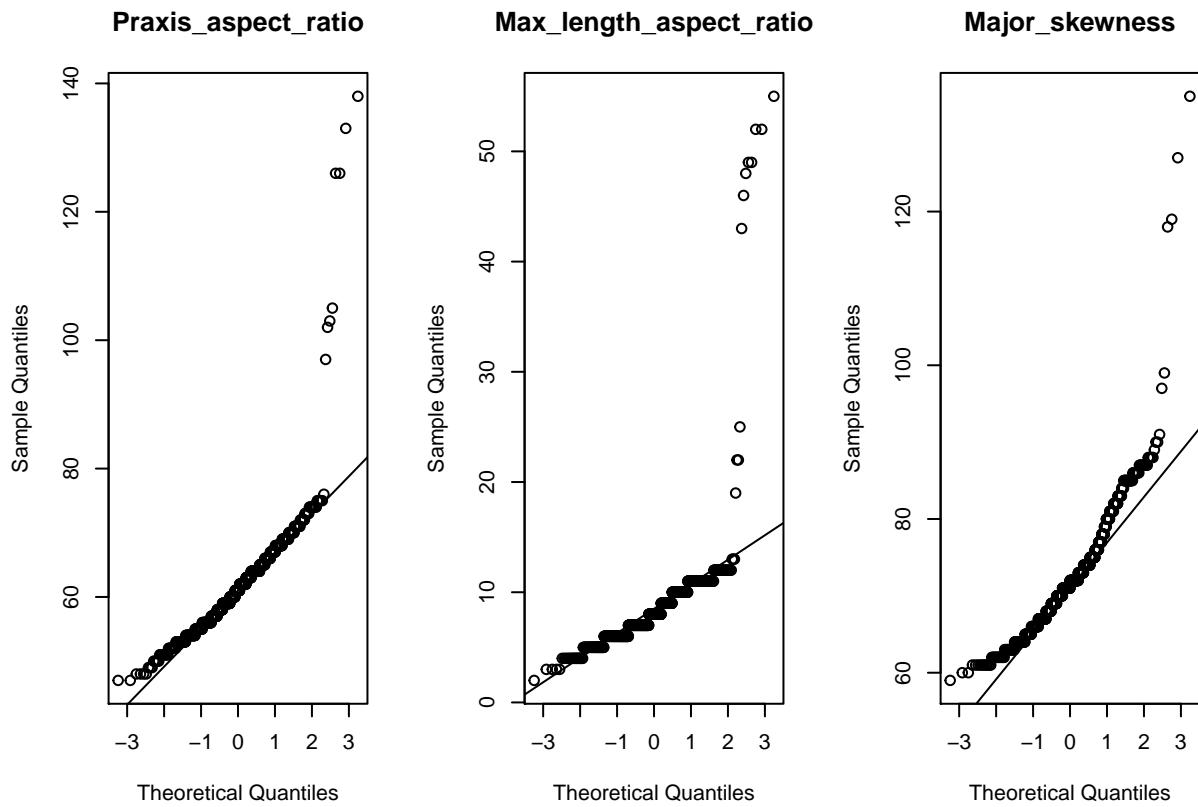
En la segunda parte de los análisis exploratorios individuales procedemos a calcular las medidas estadísticas comentadas anteriormente, repitiendo el proceso realizado para el dataset de regresión. Tal y como predijimos en las hipótesis iniciales, la mayoría de predictores se caracterizan por una **notoria dispersión en sus datos**, siendo especialmente acusada en el caso de `Radius_ratio`, `Scatter_ratio`, `Minor_variance`, `Major_variance` y `Gyration_radius`. Este hecho explica su gran variabilidad de valores y determina que existen muestras de muy diversas características. Apoyando esta teoría podemos observar los valores de las desviaciones típicas, los cuales también son más altos para aquellos predictores con una mayor variabilidad. Como consecuencia, podemos afirmar que muchas de las **muestras disponen de valores considerablemente lejanos** a las medias para cada predictor. Existen dos posibles explicaciones para esta situación, por un lado dependiendo de los ángulos en los que se hayan tomado las imágenes para cada vehículo, las propiedades pueden variar enormemente, mientras que por otro se encuentran las diferencias físicas entre los diferentes tipos de vehículos.

En referencia a la forma de la distribución, podemos observar que la mayoría de variables disponen de un coeficiente de asimetría por debajo de 0.5, por lo que la **mayoría de predictores parecen tener distribuciones simétricas**. Sin embargo, existen algunos casos excepcionales como `Scatter_ratio`, `Praxis_rectangular`, `Minor_variance`, `Minor_skewness` y `Minor_kurtosis` con una **moderada asimetría positiva**, por lo que estas variables se caracterizan por tener una concentración de datos a la derecha. Los casos más extremos son los predictores `Praxis_aspect_ratio`, `Max_length_aspect_ratio` y `Major_skewness` cuyos coeficientes de asimetría son superiores a 2 y por ello disponen de una **acusada asimetría positiva** con unas considerables concentraciones de muestras también a la derecha. Mediante esta medida estadística podemos intuir que existe una multitud de vehículos con valores sumamente extremos que producen distribuciones asimétricas en ciertas variables. La mayoría se encuentran asociadas con las relaciones entre las dimensiones de las imágenes y los radios de las figuras. Asimismo, el hecho de que **todas las asimetrías se encuentren a la derecha supone que se trata de valores cercanos al límite superior del intervalo**, por lo que pueden estar asociados a vehículos de mayor tamaño, como los autobuses y caravanas.

Una situación similar ocurre con el coeficiente de curtosis. La **mayoría de predictores disponen de un valor cercano a 0**, lo que indica que la forma de sus distribuciones pueden ser similares a la distribución normal. No es el caso de las variables `Praxis_aspect_ratio`, `Max_length_aspect_ratio` y `Major_skewness`, cuyos valores son considerablemente altos. Como consecuencia sus distribuciones presentan enormes concentraciones de datos en una zona específica del intervalo, representándose con un pico muy pronunciado.

	var	sd	skew	kurtosis
## Compactness	67.884834	8.239225	0.38027554	-0.54748625
## Circularity	38.100662	6.172573	0.26348002	-0.93214414
## Distance_circularity	249.034976	15.780842	0.10698248	-0.98689769
## Radius_ratio	1121.617157	33.490553	0.39007754	0.28235014
## Praxis_aspect_ratio	62.172234	7.884937	3.81932722	29.68954219
## Max_length_aspect_ratio	21.193844	4.603677	6.75259945	57.83152861
## Scatter_ratio	1106.482591	33.263833	0.60260098	-0.62810770
## Elongatedness	61.091416	7.816100	0.04812954	-0.87349340
## Praxis_rectangular	6.726739	2.593596	0.76678368	-0.40755701
## Length_rectangular	210.810225	14.519305	0.25773216	-0.77697995
## Major_variance	986.614493	31.410420	0.64795707	0.09938897
## Minor_variance	31252.869393	176.784811	0.83138350	-0.23210620
## Gyration_radius	1060.426540	32.564191	0.28009723	-0.50246883
## Major_skewness	56.114005	7.490928	2.06328731	11.24835551
## Minor_skewness	24.218688	4.921249	0.77032955	0.06958587
## Minor_kurtosis	79.847845	8.935762	0.68793667	-0.15513156
## Major_kurtosis	38.034858	6.167241	0.24648164	-0.60600307
## Hollows_ratio	55.399052	7.443054	-0.22477478	-0.82348470

A continuación procedemos a realizar un estudio sobre la normalidad de cada uno de los predictores. De nuevo, vamos a experimentar con un método visual para las variables más peculiares y un método estadístico para determinar con certeza la normalidad de cada predictor. Aplicaremos los gráficos QQ a las tres variables anteriores que han destacado tanto por su asimetría como por su coeficiente de curtosis. Como podemos apreciar, **los tres gráficos son muy similares**, sus cuantiles teóricos y empíricos coinciden razonablemente en el intervalo [-1, 1] y son diferentes al principio y especialmente al final, debido a las colas que les caracterizan a la derecha de la distribución. Debido a esta gigantesca concentración de altos valores, mi hipótesis inicial es que **ninguna de las tres variables siguen una distribución normal**.



Para comprobar esta teoría y la normalidad de las restantes variables numéricas, procedemos a aplicar el test de Shapiro-Wilk's sobre cada una. Recordemos que la hipótesis nula determina que sí sigue una distribución normal frente a la hipótesis alternativa contraria. Como podemos observar en los resultados, **todos los tests han vuelto a rechazar la hipótesis nula** puesto que sus p-valores son menores que el umbral establecido en 0.05. Así, con un 95% de evidencia estadística podemos determinar que **ninguno de los predictores de este dataset sigue una distribución normal**.

```
##          Variables      PValores
## 1      Compactness 2.774341e-10
## 2      Circularity 1.305447e-13
## 3 Distance_circularity 7.000994e-15
## 4      Radius_ratio 3.032547e-12
## 5 Praxis_aspect_ratio 7.706673e-34
## 6 Max_length_aspect_ratio 1.957048e-44
## 7      Scatter_ratio 2.299074e-19
## 8      Elongatedness 1.634349e-14
## 9 Praxis_rectangular 1.082978e-23
## 10 Length_rectangular 1.711942e-10
## 11 Major_variance 7.686128e-17
## 12 Minor_variance 4.163513e-23
## 13 Gyration_radius 6.892050e-08
## 14 Major_skewness 1.471981e-25
## 15 Minor_skewness 1.841791e-18
## 16 Minor_kurtosis 1.350752e-16
## 17 Major_kurtosis 7.235952e-09
## 18 Hollows_ratio 1.626074e-13
```

2.3. Análisis Exploratorio Multivariante

Como en el apartado de regresión, a continuación presentamos un conjunto de análisis globales en los que se involucran a todas las variables de este dataset. El principal objetivo consiste en visualizar y determinar las relaciones y patrones existentes entre ellos que ayuden a la predicción de la variable **Class**. Algunos de los estudios estadísticos solo admiten variables numéricas, como es el caso de las correlaciones, por lo que se añade **una columna más con los valores de Class codificados como etiquetas numéricas**.

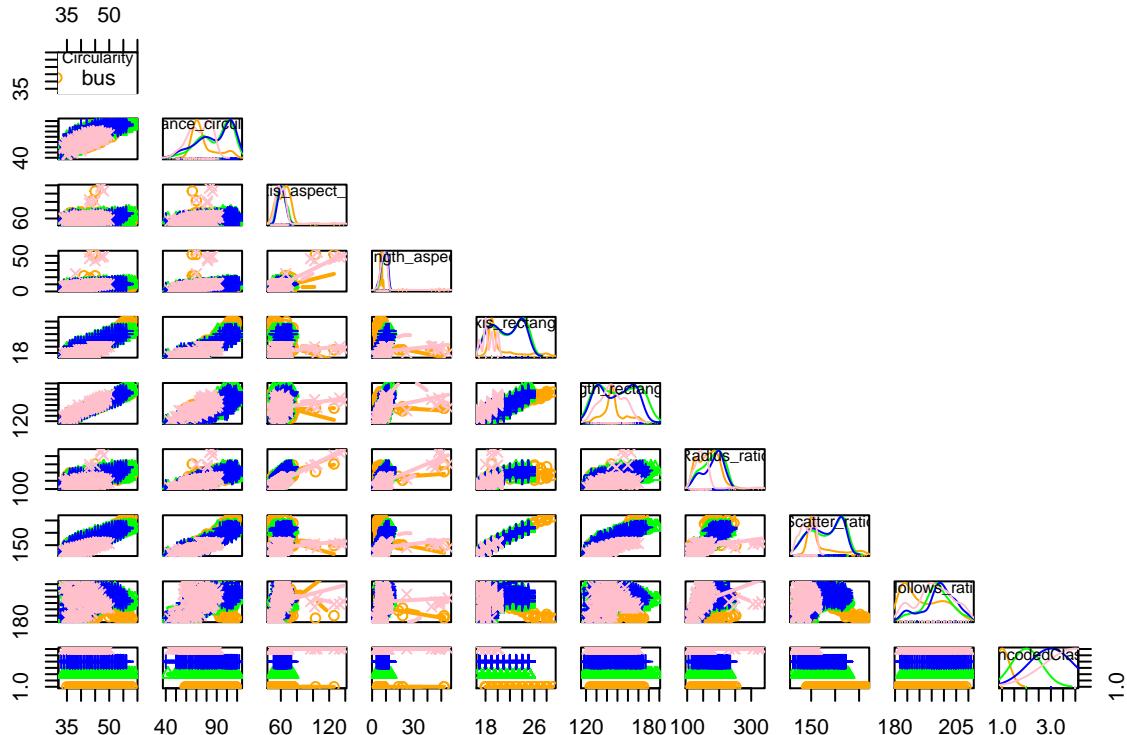
2.3.1. Matriz de Puntos

A diferencia del apartado de regresión, en este caso no se puede representar una única matriz de puntos con todas las variables disponibles, puesto que su mayor número provoca que el gráfico resultante sea imposible de interpretar. Para resolver este inconveniente vamos a hacer **dos grupos de diez variables para graficar dos matrices de puntos, incluyendo en ambas la variable independiente Class**. Como la información disponible sobre cada predictor es mínima, he tratado de componer los **grupos según los nombres en común y sus posibles significados**, de modo que el primero de ellos contiene las medidas relativas a los diferentes ratios, mientras que en el segundo se encuentran las métricas asociadas a los ejes de las siluetas. De nuevo, **diferenciamos las muestras según el tipo de vehículo** asignándoles un color y forma diferente. A continuación se presentan las distintas asociaciones: naranja-bus, verde-opel, azul-saab, rosa-van.

Si observamos las curvas de densidad del primer gráfico, podemos apreciar que la mayoría se caracteriza por tener varios máximos globales, lo que refuerza la teoría de que no siguen una distribución normal. Por otro lado, también se observa que en esta primera lista de predictores los valores de los **vehículos opel y saab son prácticamente similares**. Esta semejanza puede ser un problema en el entrenamiento de modelos puesto que, en este grupo, **no existen variables con los que poder diferenciarlos**. Esta situación no ocurre para los otros dos transportes de mayor tamaño, cuyos valores fluctúan más.

Por otro lado, observando las gráficas comparativas la característica que más destaca es que en algunas se puede apreciar una clara función creciente, mientras que otras son más confusas. Existen algunas medidas en las que las **diferencias entre los distintos tipos de vehículos son mínimas**, como es el caso de **Circularity, Distance_circularity, Length_rectangular y Hollows_ratio**. Este hecho puede indicar que **estas variables no aportan información suficiente** como para distinguir los vehículos disponibles en este dataset, por lo que podrían ser descartadas para el entrenamiento de los modelos. No ocurre la misma situación con las variables **Praxis_aspect_ratio, Max_length_aspect_ratio, Praxis_rectangular, Radius_ratio y Scatter_ratio**, en cuyas gráficas junto con la variable independiente podemos observar diferentes valores para los distintos tipos de transportes. En los dos primeros predictores, son los **vehículos de mayor tamaño los que disponen de mayores valores**. Una posible explicación consiste en que, al ser variables que miden la relación entre el radio mínimo y máximo, pueden **distinguir aquellos vehículos de mayor longitud**. Por el contrario, en las variables **Praxis_rectangular y Scatter_ratio** podemos observar que los dos tipos de automóviles disponen de los mismos rangos de valores, mientras que las furgonetas tienen los valores más bajos y los autobuses los más altos. Este hecho nos puede indicar que los **dos tipos de automóviles, al disponer de características parecidas, tienen la misma relación entre sus áreas y dimensiones**, mientras que en el caso de las furgonetas esta asociación es menor y para los autobuses es mayor. Quizás estos predictores puedan ser útiles para identificar a vehículos de mayor tamaño, como las furgonetas.

Finalmente analizamos las posibles relaciones lineales existentes entre las variables comparadas de este primer grupo. En base a los nombre de las variables, podemos intuir que **aquellas que se refieren a la misma propiedad**, como es el caso de **Circularity y Distance_circularity, pueden compartir una relación monotónica** y por lo tanto pueden aportar la misma información. En este caso, bastaría con seleccionar la que más información aporte para descartar las restantes en la construcción de los modelos de clasificación. Asimismo, a excepción de las gráficas comparativas con **Hollows_ratio**, en la mayoría se puede observar relaciones crecientes puesto que en sus representaciones los valores continúan aumentando a lo largo del intervalo. Este hecho puede indicarnos que la **mayoría de predictores se encuentran relacionados linealmente**, por lo que contienen una gran cantidad de información común.

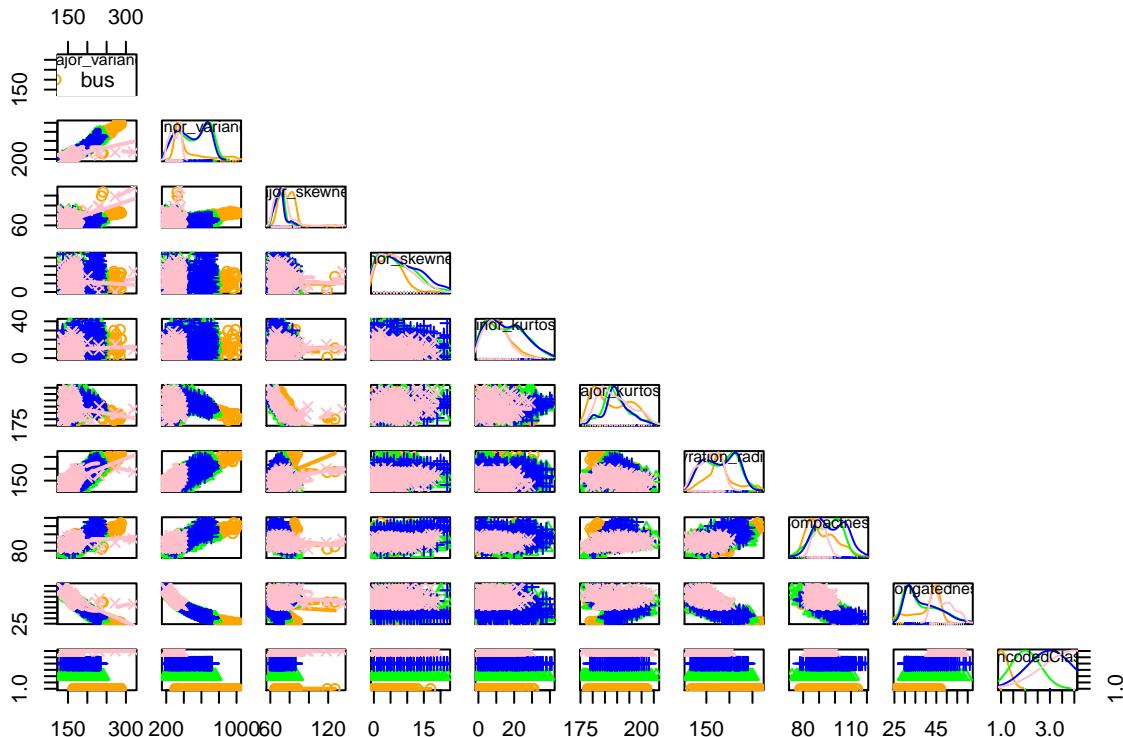


En esta segunda matriz de puntos se representan los restantes predictores incluyendo, también, la variable independiente **Class**. De nuevo, si observamos sus curvas de densidad podemos visualizar que la mayoría de ellas se caracterizan por dos propiedades: varios valores máximos globales y una **gran similitud entre los dos tipos de automóviles**. Adicionalmente, si examinamos las gráficas comparativas de los predictores con la variable a predecir, podemos apreciar que en **más de la mitad los valores son similares para los cuatro tipos de vehículos**. De nuevo, estos predictores parecen no ayudar en la identificación de cada uno de los transportes disponibles, por lo que quizás puedan ser eliminados de los modelos. Las cinco variables que pueden ser de mayor utilidad se listan a continuación:

- Los predictores **Minor_variance** y **Compactness** disponen de valores inferiores para el tipo de vehículo **van**, mientras que los restantes transportes disponen de rangos de valores más amplios, lo cual parece indicar que las **furgonetas disponen de siluetas más esféricas** que los demás vehículos. Además, en el caso de la última variable, podemos observar que junto con **Major_kurtosis** son las únicas en las que existen ciertas diferencias entre los dos tipos de automóviles. Por lo que estos dos predictores pueden aportar cierta información que permita su distinción.
- La variable **Major_skewness** contiene un intervalo de valores mayor para los vehículos de mayor tamaño que para los dos tipos de automóviles. Este predictor puede ayudar a **distinguir entre estos dos grupos de transportes**, ya que las siluetas de los autobuses y furgonetas suelen estar caracterizadas por una mayor asimetría.
- Por último, la variable **Elongatedness** dispone de valores inferiores para el tipo de vehículo **bus** en contraposición con los valores más altos asociados al tipo de transporte **van**. Esto indica que las **siluetas de las furgonetas son las más estiradas**, por lo que este predictor puede proporcionar información útil con la que diferenciar los dos tipos de vehículos de mayor tamaño.

Por último, procedemos a estudiar las posibles relaciones lineales entre los diferentes predictores disponibles. A diferencia de la matriz de puntos anterior, en la mayoría de las gráficas no se reconoce ninguna función particular que determine el aumento o disminución de los valores, por lo que **parece existir un menor número de asociaciones**. Las más destacables son las relaciones lineales entre las variables **Major_variance**, **Minor_variance** en combinación con **Gyration_radius**, **Compactness** y **Elongatedness**. Existen **cuatro relaciones monotónicas decrecientes** entre esta última variable y las cuatro restantes mencionadas

anteriormente. En primer lugar se encuentran las tres asociadas con los momentos de las imágenes en el radio mínimo y máximo, que parecen indicar que mayor valor, menor es la elasticidad de la silueta. Mientras que para la variable **Compactness** parece que conforme menos elástica es la silueta, menos estirada es, característica que parece lógica si consideramos la elasticidad de un círculo perfecto.



2.3.2. Correlaciones

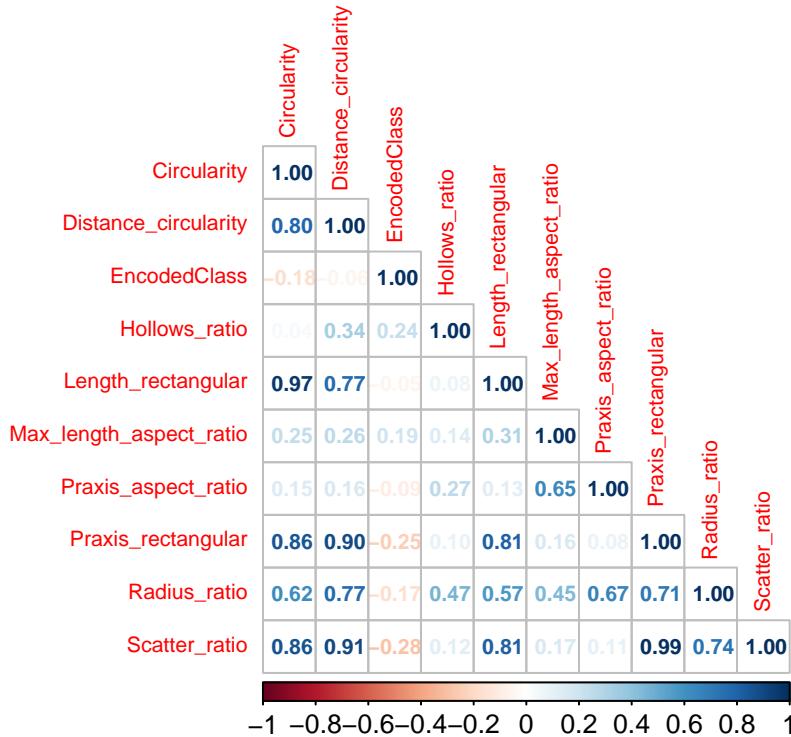
A continuación se representarán dos gráficas de correlaciones para cada uno de los diferentes grupos anteriores con el objetivo de identificar posibles asociaciones entre los predictores y con la variable independiente. Así, se pretende conocer cuáles son las variables más prometedoras para predecir el tipo de vehículo a partir de las propiedades de su silueta. Tal y como podemos observar en el siguiente gráfico, nuestra teoría inicial de que podría haber numerosas relaciones entre los predictores parece cumplirse. En particular, las siguientes asociaciones son las más relevantes:

1. En primer lugar se encuentran los predictores **Circularity** y **Distance_circularity** con un coeficiente de correlación de 0.8, lo que indica una **relación alta**. Sin embargo, esta asociación era de esperar puesto que ambas variables hacen referencia a la misma propiedad. Esto nos permitirá descartar a la que aporte menos información útil para la predicción de la variable independiente. **En esta misma situación se encuentran la pareja de predictores Length_rectangular y Praxis_rectangular.** Adicionalmente, podemos observar que **las cuatro variables están relacionadas entre sí** por unos coeficientes de correlación entre 0.77 y 0.97, por lo que comparten una considerable cantidad de información. Particularmente, la correlación entre **Circularity** y **Length_rectangular** afirma que **existe una asociación perfecta**. Una posible razón explicativa es que en las fórmulas de ambos predictores se consideran **el área y el perímetro de la silueta**, dos medidas matemáticas muy relacionadas entre sí.
2. Un segundo grupo de variables correladas lo forman aquellas que hacen referencia a una proporción o ratio entre varias medidas. Especialmente destacan los predictores **Scatter_ratio** y **Praxis_rectangular** con un **coeficiente de 0.99**, lo que indica que existe una **relación total** entre sendas variables. Mientras la primera hace referencia a la inercia entre el radio mínimo y máximo, la segunda vincula el área con las dimensiones de la silueta. Aunque parezcan medidas diferentes, parecen aportar la misma información por lo que se podrá descartar aquella que menos relevante sea para el entre-

namiento de modelos de clasificación. Otras correlaciones que se pueden categorizar como moderadas las protagonizan los pares de variables **Scatter_ratio** y **Radius_ratio**, **Praxis_aspect_ratio** y **Max_length_aspect_ratio**.

3. Adicionalmente, **algunas de las variables de los dos grupos anteriores también se encuentran correladas entre sí**. La relación más destacable por su fuerza es la de **Scatter_ratio** y **Distance_circularity con un coeficiente de 0.91**, lo que indica que sendos predictores también aportan el mismo tipo de información. LA explicación de este suceso puede ser similar a la anterior y es que la primera variable hace referencia a los radios de la silueta, mientras que la otra relaciona el área con su volumen, por lo que los componentes de cada medida matemática están fuertemente vinculados.

Finalmente, destacamos la **ausencia de correlaciones importantes con la variable independiente**, por lo que en este primer grupo no parece haber predictores particulares que expliquen la variabilidad de vehículos de este dataset.

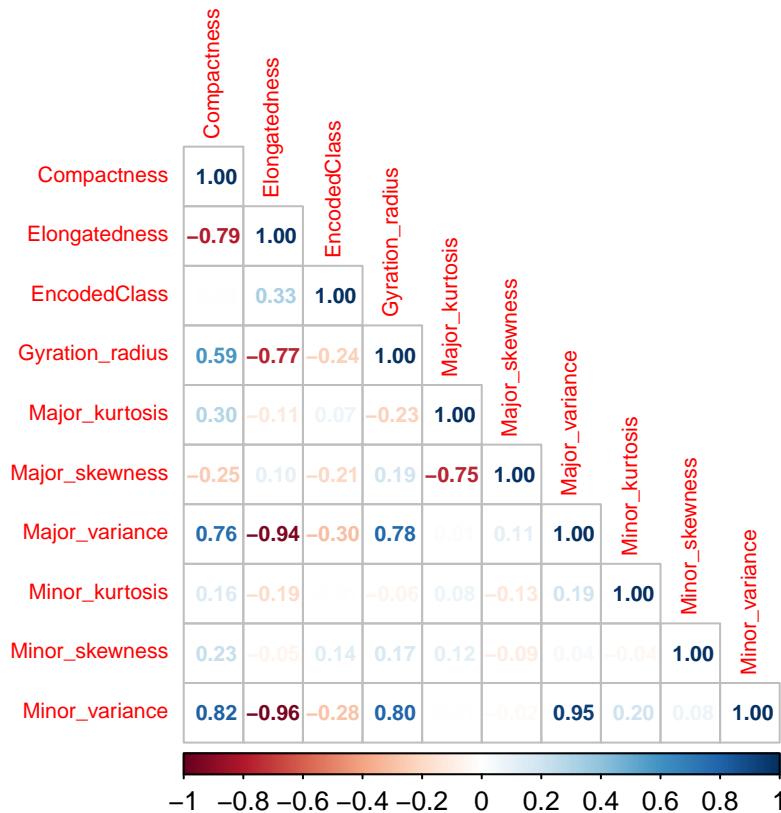


En este segundo gráfico se muestran las correlaciones existentes entre las variables del segundo grupo más la variable independiente. La primera cualidad destacable es que, a diferencia del gráfico anterior, la **mayoría de correlaciones importantes son negativas**, lo que indica que los predictores relacionados no comparten una asociación monotónica, si no un vínculo inverso, puesto que conforme aumenta uno disminuye el contrario, y viceversa. En este caso vamos a explicarlas según la obviedad de las mismas.

1. La primera asociación más trivial es la compuesta por **Compactness** y **Elongatedness** puesto que se trata de dos variables opuestas. Mientras la primera mide el grado de compactación de una silueta, entendiendo esta propiedad como el grado de similitud con un círculo perfecto, el segundo predictor representa cuán elástica es la silueta. Por lo tanto, su asociación contraria parece lógica.
2. Una segunda correlación positiva bastante obvia es la que forman las variables **Minor_variance** y **Major_variance**, puesto que ambas representan la **misma medida pero para diferentes longitudes de radio**. Esta relación se representa con una tercera variable denominada **Gyration_radius** y por ende, genera dos correlaciones adicionales con los dos primeros predictores. Esto nos indica que probablemente solo necesitemos el tercer predictor para representar toda la información que aportan estas propiedades.

3. Si existen varias correlaciones positivas moderadas y fuertes entre las variables **Compactness** y los tres predictores del punto anterior, es lógico encontrar **correlaciones negativas las tres variables anteriores y el predictor opuesto Elongatedness**. Sin embargo, estas relaciones contrarias parecen ser mucho más fuertes que las asociaciones positivas puesto que sus coeficientes son considerablemente mayores. Esto nos indica que comparten más información con el predictor **Elongatedness** que con la variable **Compactness**.
4. La última correlación a destacar la conforman **Major_skewness** y **Major_kurtosis**. Se trata de una **relación fuerte y en dirección opuesta** por su coeficiente negativo. A diferencia de los casos anteriores, esta relación no ha sido obvia a mi parecer debido a los conceptos estadísticos de asimetría y curtosis por los que me estaba guiando para intentar entender ambas variables. Sin embargo, en el caso de este dataset, parece que cuanto más asimétrica sea una silueta, menor coeficiente de curtosis, y viceversa.

Como en el gráfico anterior, podemos observar que en segundo **tampoco existen correlaciones relevantes con la variable independiente**, por lo que parece difícil explicar las siluetas de los vehículos con variables particulares. Deberemos de formar un conjunto de métricas que en conjunto aporten información suficiente para identificar los diferentes transportes disponibles.

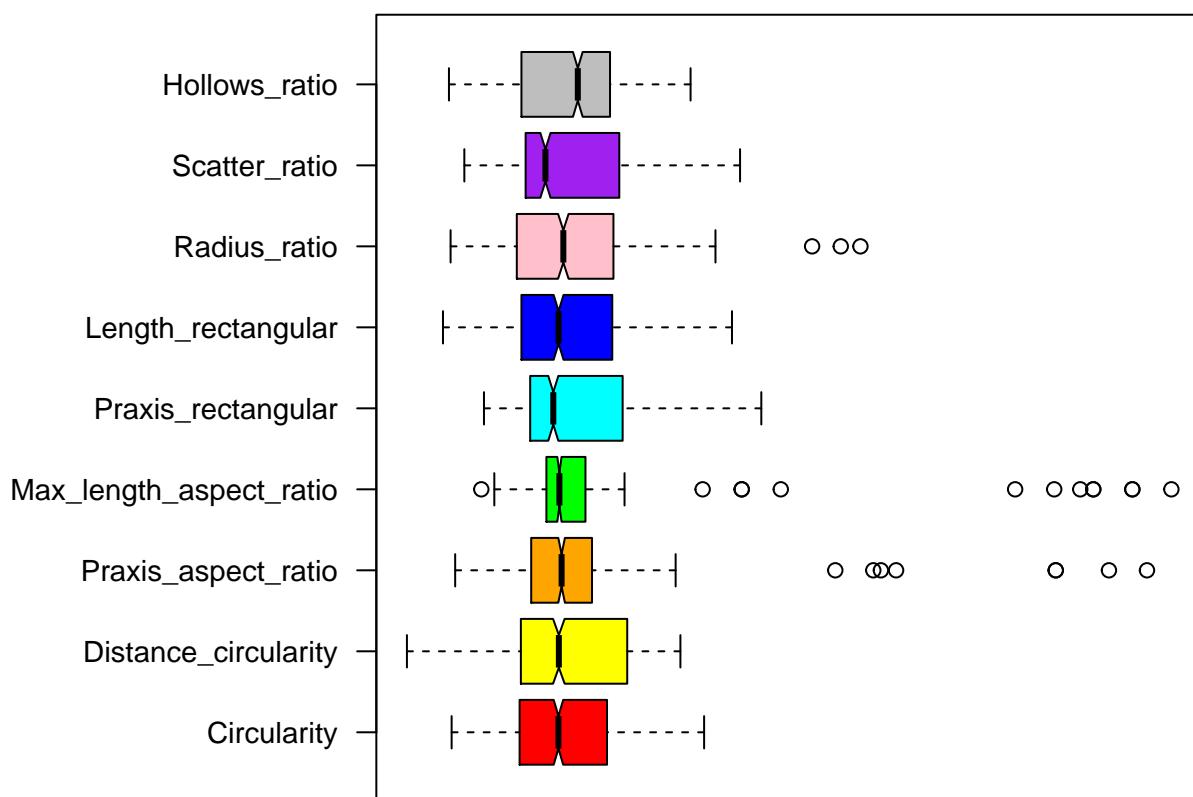


2.3.3. Diagramas de cajas

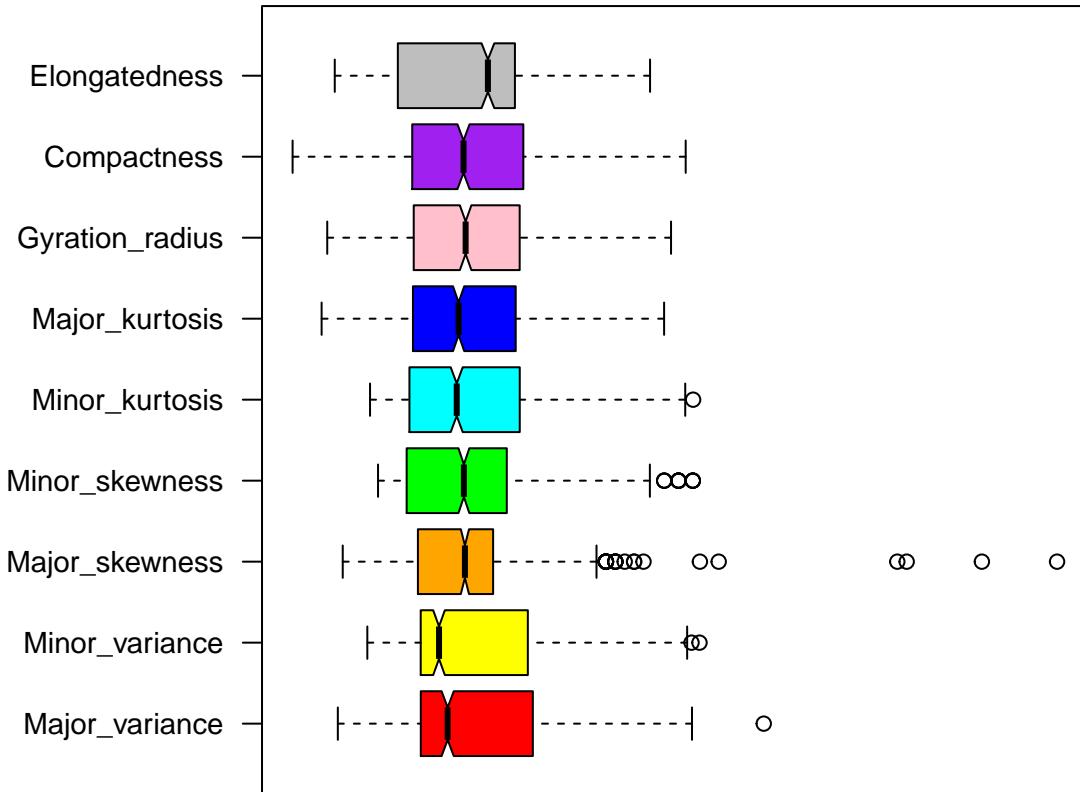
Para acabar el estudio multivariante de este dataset, a continuación procedemos a representar los diagramas de cajas de cada uno de los grupos anteriores para mostrar su dispersión y variabilidad de datos. Como hemos analizado anteriormente, a diferencia del dataset de regresión, las variables de **este conjunto de datos sí se caracterizan por escalas súmamente diferentes**, lo que puede influir tanto en este tipo de gráficos como en los clasificadores. Por ende, en primer lugar aplicamos la función **scale** para utilizar un **escalado y centrado mediante *Z score** con el que construir un segundo dataset cuyos datos dispongan de una media 0 y desviación típica de 1.

En este primer gráfico podemos observar que la **mayoría de predictores se caracterizan por una**

moderada o severa asimetría, según se ha estudiado en secciones anteriores, a excepción de las variables **Scatter_ratio**, **Max_length_aspect_ratio** y **Distance_circularity**, que disponen de niveles más bajos puesto que su mediana se sitúa aproximadamente en el centro de las cajas. Otra característica a destacar es que, como se predijo anteriormente, la **mayoría de predictores disponen de una gran dispersión en sus datos** en base a la longitud de sus correspondientes cajas. Esta hipótesis también ha sido estudiada anteriormente de forma generalizada y específica a las variables según la naturaleza de sus valores. No obstante, aún queda un tipo de análisis que no se ha efectuado aún: los valores outliers. Como podemos apreciar, **solo tres de las variables representadas disponen de muestras alejadas del conjunto de valores**. En el caso del predictor **Scatter_ratio** podemos observar que se trata de valores no demasiado alejados del valor máximo del conjunto. Sin embargo, los predictores **Praxis_rectangular** y **Max_length_aspect_ratio** sí que contienen datos extremadamente alejados del rango de valores habitual, por lo que estas variables pueden contar con un conjunto de outliers extremos. Las **tres variables hacen referencia a los radios mínimo y máximo** de una silueta, por lo que considerando que dos de los cuatro vehículos disponibles son automóviles, podemos intuir que estas muestras tan alejadas pueden referirse a **vehículos con siluetas de mayor tamaño**, como puede ser el caso de los autobuses.



A continuación se muestra un segundo gráfico con las restantes variables que no se mostraban en el anterior. Como ocurría en el primer diagrama de cajas, en esta segunda representación existen **ciertas variables con un mayor grado de asimetría**, mientras que otras disponen de una mediana bastante centrada, como es el caso de **Compactness**. De igual modo, podemos visualizar que la mayoría de predictores disponen tanto de un amplio rango de valores como de una considerable longitud de sus cajas, por lo que sus **datos también se encuentran considerablemente dispersos**. En el caso de los valores extremos, podemos observar que a diferencia de las variables anteriores, que los predictores de este gráfico **la mayoría solo dispone de outliers moderados**, es decir, puntos fuera del intervalo de valores pero bastante cercanos a él, a excepción de **Major_skewness**, que sí que dispone de un conjunto de muestras extremadamente alejadas de su respectiva caja. Como hemos analizado anteriormente, esta variable era capaz de diferenciar dos grupos de transportes: los dos tipos de automóviles y los dos vehículos de mayor tamaño. Por tanto, con gran probabilidad, los **outliers extremos que tiene harán referencia a enormes grados de asimetrías asociados a este último grupo**.



2.4. Modelos de Clasificación

En esta sección se pretende experimentar con diferentes algoritmos para entrenar diversos clasificadores y analizar sus métricas de bondad para seleccionar aquel que mejor comportamiento presente en la predicción de la variable independiente. Con el objetivo de establecer las mismas condiciones para todos los algoritmos y facilitar el entrenamiento de algunas técnicas basadas en distancias, como es el caso de KNN, **se utilizará la variable independiente codificada EncodedClass para la predicción del tipo de vehículos.**

2.4.1. K-Nearest Neighbors (KNN)

En esta subsección se entrenarán diferentes modelos utilizando el algoritmo de los K vecinos más cercanos o KNN, variando tanto el parámetro K como la métrica para calcular la matriz de distancias. Para ello vamos a reutilizar la función que se implementó para el primer ejercicio de evaluación continua de clasificación. En ella se **genera un conjunto de entrenamiento y test**, en caso de que este último no haya sido proporcionado, para calcular las **distancias entre cada muestra a clasificar y los datos de entrenamiento**. A continuación se realiza una **votación por mayoría entre los K vecinos más cercanos** para seleccionar la clase de la muestra de test. Con el objetivo de calcular la precisión para ambos conjuntos, se ha creado una segunda función para efectuar el cálculo de la matriz y la clasificación de las muestras a unos conjuntos proporcionados como argumentos.

Previo a su aplicación, recordemos los requisitos de datos asociados a este algoritmo.

1. Se deben **normalizar los datos** para que el cálculo de distancias no se vea influido por las diferentes escalas de los predictores. Este paso ya se ha realizado anteriormente al comienzo del análisis multivariante.
2. Los **valores nominales deben codificarse** como etiquetas numéricas o variables *dummy* para facilitar el cálculo de la matriz de distancias. De nuevo, este segundo proceso también ha sido realizado con anterioridad.

3. Los **valores perdidos o nulos deben ser eliminados o imputados**. En este caso el dataset no dispone de valores perdidos, tal y como se observó en el análisis general.

Una vez hemos comprobado que se cumplen las condiciones necesarias, podemos aplicar el algoritmo KNN. En mi caso particular he realizado cuatro experimentos diferentes, es decir, se han entrenado cuatro modelos distintos utilizando esta técnica para experimentar con dos métricas y dos valores de K diferentes. En los **dos primeros modelos se ha aplicado la distancia de Manhattan**, puesto que los cálculos que realiza para determinar la similitud entre dos muestras son más realistas que los que efectúa la distancia Euclídea al trazar una simple línea recta entre ambas. Mientras que para los **dos últimos modelos se usa el índice de Jaccard**, el cual realiza la intersección entre las muestras coincidentes y el total. Esta métrica suele aplicarse para el estudio de patrones, comparando sus diferencias y similitudes, y por ende me ha parecido adecuada para incluir en los experimentos puesto que en este problema de clasificación el objetivo es obtener las diferencias que permitan distinguir entre los cuatro tipo de vehículos. Adicionalmente, se han probado **dos valores del parámetro K, uno más pequeño y otro más alto** para comprobar cómo afecta en el entrenamiento de los modelos.

```
## [1] 0.8092105 0.2531646
## [1] 0.6236842 0.2533333
## [1] 0.8000000 0.2837838
## [1] 0.5947368 0.2692308
```

En los resultados que podemos observar, destacamos que la primera cifra se corresponde con la tasa de acierto sobre el conjunto de entrenamiento, mientras que la segunda se efectúa sobre el conjunto de test. Así, para el **primer modelo entrenado con K=3 y la distancia Manhattan**, podemos observar que su tasa de acierto en **entrenamiento es de un 81% aproximadamente**, mientras que en el conjunto de **test es de un 25%**. Como podemos apreciar, existe una gran diferencia entre el comportamiento que presenta con los datos de entrenamiento que con las muestras desconocidas. Una posible explicación reside en el valor del parámetro K, que al ser muy bajo puede favorecer la aparición del **sobreajuste**. Sin embargo, al observar las tasas de acierto del segundo modelo entrenado con la misma distancia pero con un valor de K más alto, podemos observar que si bien su **precisión sobre entrenamiento ha disminuido** considerablemente, la tasa de aciertos **en el conjunto de test es prácticamente similar**. A diferencia del primer modelo, en este caso la situación este segundo clasificador se caracteriza por un **subajuste**, lo que significa que no se ha fijado bien en los datos de entrenamiento para aprender los patrones de cada vehículo.

Observando los resultados de los **dos últimos modelos entrenados con el índice de Jaccard**, podemos apreciar que en el tercer modelo entrenado con K=3 dispone de una **precisión de entrenamiento similar** al primer clasificador que comparte el valor de este parámetro. Sin embargo, con esta nueva métrica se ha **mejorado la precisión sobre el conjunto de test hasta un 28%, aproximadamente**. Esto nos indica que el **índice de Jaccard es más efectivo** para calcular la similitud entre las siluetas de los diferentes vehículos. No obstante, igual que en el primer clasificador, este modelo también sufre un **importante sobreajuste**. Por otro lado, en el cuarto modelo entrenado con K=21 la tasa de acierto sobre el conjunto de entrenamiento es ligeramente menor que el segundo clasificador con el mismo parámetro, pero la **precisión del conjunto de test es ligeramente mejor**.

En función de los experimentos realizados podemos extraer las siguientes conclusiones:

1. Parece obvio que el comportamiento del algoritmo **KNN es altamente dependiente de la métrica utilizada para calcular la matriz de distancias** con la que clasificar las muestras. Por lo tanto, se deberán estudiar las distancias disponibles para intentar aplicar la que mejor se ajuste a cada problema en particular.
2. Si establecemos un **valor pequeño para el parámetro K**, existe una alta probabilidad de que el modelo sufra un **severo sobreajuste** como hemos podido comprobar. En cambio, con **valores más altos ocurre una situación opuesta de subajuste**, en la que el clasificador no es capaz de identificar completamente los patrones de cada clase y, por ende, pierde capacidad de generalización.

3. Finalmente, además de la influencia de los dos anteriores parámetros, también hay que considerar que el algoritmo KNN suele proporcionar **buenos resultados en caso de que las clases se encuentren bien separadas**. Como hemos observado en los diferentes análisis exploratorios, esta condición no se cumple puesto que las características de las siluetas de los distintos vehículos son muy similares, especialmente los dos tipos de automóviles.

2.4.2. Linear Discriminant Analysis (LDA)

A continuación se pretende utilizar el algoritmo LDA para intentar mejorar los resultados proporcionados de los clasificadores entrenados con KNN. Como en el caso anterior, vamos a utilizar el **conjunto de datos escalado y todos los predictores disponibles**. Previo a su aplicación, vamos a comprobar las asunciones asociadas:

1. **Las muestras de datos son independientes entre sí.** Como no conocemos el proceso de extracción de las siluetas de cada uno de los vehículos disponibles, asumiremos que esta condición se cumple.
2. **Los predictores siguen una distribución normal para cada clase.** En los análisis exploratorios hemos podido comprobar que ninguna de las variables sigue una distribución normal, por lo que es de esperar que tampoco se siga para cada una de las clases disponibles, por lo que esta condición no se cumple.
3. **Los predictores tienen una misma matriz de covarianza para cada clase.** Esta asunción se comprueba en el siguiente *chunk* aplicando el test de Levene a cada uno de los predictores que se van a utilizar. La hipótesis nula afirma que no disponen de la misma varianza, mientras que la hipótesis alternativa apuesta por la opción contraria. Como podemos observar en los resultados, todos los tests rechazan la hipótesis nula por lo que se confirma que **todos los predictores comparten la misma covarianza**.

```
##  
##  Compactness  
## Levene's Test for Homogeneity of Variance (center = median)  
##          Df F value    Pr(>F)  
## group     3  48.315 < 2.2e-16 ***  
##          841  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
##  Circularity  
## Levene's Test for Homogeneity of Variance (center = median)  
##          Df F value    Pr(>F)  
## group     3  54.641 < 2.2e-16 ***  
##          841  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
##  Distance_circularity  
## Levene's Test for Homogeneity of Variance (center = median)  
##          Df F value    Pr(>F)  
## group     3 22.588 4.521e-14 ***  
##          841  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
##  Radius_ratio  
## Levene's Test for Homogeneity of Variance (center = median)  
##          Df F value    Pr(>F)
```

```

## group 3 8.1635 2.319e-05 ***
##      841
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Praxis_aspect_ratio
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value   Pr(>F)
## group 3 12.052 9.919e-08 ***
##      841
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Max_length_aspect_ratio
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value   Pr(>F)
## group 3 5.0064 0.001908 **
##      841
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Scatter_ratio
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value   Pr(>F)
## group 3 31.553 < 2.2e-16 ***
##      841
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Elongatedness
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value   Pr(>F)
## group 3 11.163 3.451e-07 ***
##      841
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Praxis_rectangular
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value   Pr(>F)
## group 3 30.512 < 2.2e-16 ***
##      841
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Length_rectangular
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value   Pr(>F)
## group 3 63.885 < 2.2e-16 ***
##      841
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Major_variance

```

```

## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     3 16.699 1.518e-10 ***
##           841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Minor_variance
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     3 39.49 < 2.2e-16 ***
##           841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Gyration_radius
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     3 21.211 2.987e-13 ***
##           841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Major_skewness
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     3 11.298 2.856e-07 ***
##           841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Minor_skewness
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     3 25.561 7.869e-16 ***
##           841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Minor_kurtosis
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     3 34.687 < 2.2e-16 ***
##           841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Major_kurtosis
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     3 22.881 3.028e-14 ***
##           841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

## 
##  Hollows_ratio
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group     3 13.562 1.2e-08 ***
##          841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Tras conocer que solo dos de las tres condiciones necesarias se cumplen para aplicar el algoritmo LDA, a continuación entrenamos y validamos un primer modelo. En los siguientes resultados podemos observar que se ha conseguido una **tasa de acierto del 79.7% para entrenamiento y del 76.4% para test**, por lo que este clasificador dispone de una mayor capacidad de generalización con respecto a los anteriores y sin sufrir sobreajuste o subajuste. Una posible razón explicativa de este buen comportamiento reside en el cumplimiento de la tercera asunción, en el que **todos los predictores utilizados disponen de la misma covarianza**, pese a no seguir una distribución normal. Observando la matriz de confusión, podemos apreciar que la **mayoría de errores se concentran en los dos tipos de automóviles**. Este hecho confirma nuestra teoría inicial de que disponen de características muy similares y que no existe suficiente información como para identificar cada uno de los modelos de coche, con una mayor precisión. Por otro lado, parece ser que los transportes de mayor tamaño son más fácilmente reconocibles por este modelo.

```

## [1] 0.7960526
## [1] 0.7882353
##
##           bus   opel   saab   van
## bus      23      1      3      1
## opel      0     17      8      0
## saab      0      4     15      0
## van       1      0      0     12

```

2.4.3. Quadratic Discriminant Analysis (QDA)

Tras conocer los resultados proporcionados al entrenar un nuevo modelo con LDA, procedemos a aplicar la versión cuadrática de este algoritmo: QDA. Esta técnica dispone de las mismas condiciones anteriores, a excepción de la igualdad de covarianzas puesto que se encarga de calcular una matriz de covarianza particular a cada clase. A continuación se construye un modelo mediante QDA utilizando los **mismos conjuntos de entrenamiento y test** anteriores y **considerando todos los predictores** disponibles para predecir la variable **Class**. De este modo, las condiciones de entrenamiento son las mismas con el objetivo de comparar los resultados con el clasificador anterior.

Tal y como podemos observar en los siguientes resultados, el modelo entrenado con QDA ha conseguido mejorar la tasa de aciertos en **entrenamiento hasta un 91% y en test 87%**. Puesto que, como analizamos anteriormente todos los predictores comparten la misma covarianza, una posible explicación para esta considerable mejora puede fundamentarse en la flexibilidad que aporta este método a la hora de calcular **fronteras cuadráticas** en lugar de lineales. Esta técnica es de gran ayuda en aquellos casos en los que las clases no son fácilmente separables linealmente. Si analizamos la matriz de confusión podemos observar que aumenta su capacidad para **identificar mejor los dos tipos de automóviles**, puesto que el número de errores se reduce considerablemente comparado con el modelo entrenado con LDA.

```

## [1] 0.9105263
## [1] 0.8823529
##
##           bus   opel   saab   van
## bus      23      0      1      0

```

```

##    opel     0     19     5     0
##    saab     0      3    20     0
##    van      1      0     0    13

```

2.5. Comparación de algoritmos de Clasificación

Para terminar con este apartado de clasificación, se pretende realizar una comparación entre el comportamiento de los algoritmos anteriores utilizando varios problemas de clasificación. En primer lugar, obtenemos la precisión media de diez modelos entrenados con **KNN**, **LDA** y **QDA** aplicando validación cruzada durante diez iteraciones. Con el objetivo de comparar los resultados, en todos los clasificadores se consideran la totalidad de predictores disponibles y se utilizan las diez particiones específicas de entrenamiento y test para este dataset.

```

## [1] 0.9401087
## [1] 0.7163445
## [1] 0.7989229
## [1] 0.7813305
## [1] 0.9123989
## [1] 0.8522409

```

Una vez disponemos de la precisión en entrenamiento y validación para cada uno de los tres algoritmos, las sustituimos en los ficheros `clasif_train_alumnos.csv` y `clasif_test_alumnos.csv`, respectivamente, en la línea en la que aparece el dataset `vehicle`. Con el objetivo de comparar el comportamiento de las tres técnicas volvemos a aplicar el **test de Friedman**. Recordemos que la hipótesis nula apostaba por que los tres algoritmos son iguales, mientras que la hipótesis alternativa por la opción contraria. Como podemos observar en los resultados, el p-valor resultante no es menor que el umbral 0.05 por lo que **no podemos rechazar la hipótesis de que sean iguales**. Aunque no haya evidencias estadísticas de que los algoritmos se comporten de forma distinta, vamos a agruparlos por pares siendo 1:KNN, 2:LDA y 3:QDA. El segundo objetivo consiste en aplicar **varios tests de Wilcoxon con una penalización por pasos de Holm** para intentar identificar si existe algún algoritmo que destaque entre los tres. Sin embargo, de nuevo podemos observar como los p-valores de cada pareja de técnicas son mayores que el umbral establecido. Con esto concluimos que **no existe una certeza estadística** que respalde la teoría de que los algoritmos **KNN**, **LDA** y **QDA** son diferentes sobre los datasets experimentados.

```

##
## Friedman rank sum test
##
## data: as.matrix(tablatst)
## Friedman chi-squared = 1.2, df = 2, p-value = 0.5488
##
## Pairwise comparisons using Wilcoxon signed rank test
##
## data: as.matrix(tablatst) and groups
##
##    1     2
## 2 0.70 -
## 3 0.53 0.99
##
## P value adjustment method: holm

```

3. Apéndice de Regresión

```
# Librerías necesarias
require(car)
require(tidyverse)
require(psych)
require(corrplot)
require(kknn)
require(phulentropy)
require(MASS)
# Carga de las librerías
library(car)
library(tidyverse)
library(psych)
library(corrplot)
library(kknn)
library(phulentropy)
library(MASS)
# Semilla para que los resultados que dependen de la aleatoriedad
# sean reproducibles.
set.seed(0)
```

3.1. Análisis Exploratorio General

```
# Nombres de las variables según el fichero `abalone.dat`
abalone.colnames <- c("Sex", "Length", "Diameter", "Height", "Whole_weight",
                      "Shucked_weight", "Viscera_weight", "Shell_weight", "Ring")
# Leemos el dataset mediante el fichero `abalone.dat`
# Parámetros
# 1. Ruta hacia el fichero
# 2. Las columnas no se encuentran en la primera fila
# 3. Ignoramos las 13 primeras filas hasta alcanzar los datos
# 4. Separador de columnas
# 5. Nombres de las columnas
abalone.df <- read.table("./abalone/abalone.dat", header=FALSE, skip=13,
                           sep=",", col.names=abalone.colnames)
# Dimensión
dim(abalone.df)
# Tipos de las variables
str(abalone.df)
# Resumen del dataset
summary(abalone.df)
```

3.2. Análisis Exploratorios Univariantes

3.2.1. Variables nominales

```
# Definimos los géneros disponibles
genders <- c("Masculino", "Femenino", "Infante")
# Parámetros:
# 1. Convertimos las etiquetas a factores para obtener la frecuencia por clase
# 2. Contorno negro a las barras y la leyenda
# 3. Fondo blanco
```

```

# 4. Establecemos el nombre y las etiquetas de la leyenda para cada género
# 5. Ocultamos la etiqueta del eje X y establecemos la etiqueta del eje Y
# 6. Frecuencia de cada clase sobre sus respectivas barras
ggplot(abalone.df, aes(x=factor(Sex), fill=factor(Sex))) +
  geom_bar(color="black") + theme_minimal() +
  scale_fill_discrete(name='Géneros', labels=genders) +
  labs(x="", y="Frecuencias") +
  geom_text(stat='count', aes(label=..count..), vjust=2)

```

3.2.2. Variables numéricas

```

# Cálculo de la varianza para cada variable numérica
abalone.vars <- sapply(c(2:ncol(abalone.df)), function(x) var(abalone.df[,x]))
# Cálculo de la desviación típica, asimetría y curtosis
abalone.statistics <- data.frame(describe(abalone.df %>% dplyr::select(-Sex)))
# Dataframe con todas las medidas estadísticas por variable numérica
abalone.statistics <- abalone.statistics %>% dplyr::select(c('sd', 'skew', 'kurtosis'))
cbind(var=abalone.vars, abalone.statistics)

# Representar los dos siguientes gráficos en una única imagen
par(mfrow=c(1,2))
# Gráfico QQ para la variable dependiente `Height`
qqnorm(abalone.df$Height, main="Gráfico QQ de Height")
qqline(abalone.df$Height)
# Gráfico QQ para la variable independiente `Ring`
qqnorm(abalone.df$Ring, main="Gráfico QQ de Ring")
qqline(abalone.df$Ring)

# Aplicamos el test de Shapiro-Wilk's para cada variable extrayendo el p-valor
abalone.shapiros <- sapply(c(2:ncol(abalone.df)),
                           function(x) shapiro.test(abalone.df[,x])$p.value)
# Definimos un dataframe con los resultados para cada variable
abalone.shapiros.df <- data.frame(VARIABLES=colnames(abalone.df)%>%dplyr::select(-Sex)),
                           PValores=abalone.shapiros)
abalone.shapiros.df

```

3.3. Análisis Exploratorio Multivariante

3.3.1. Matriz de Puntos

```

# Parámetros
# 1. Representa solo la diagonal inferior porque la matriz es simétrica
# 2. Diferencia los ejemplares por género cambiando el color y su forma
# 2.1. Masculino: círculos azules.
# 2.2. Femenino: triángulos naranjas.
# 2.3. Infantes: cruces grises.
# 3. Representa curvas de densidad en la diagonal
scatterplotMatrix(abalone.df, diagonal=list(method="density"),
                  upper.panel=NULL, col=c("blue", "orange", "gray"),
                  group=as.factor(abalone.df$Sex))

```

3.3.2. Correlaciones

```
# Representa los coeficientes de correlación para todas las variables
# Parámetros
# 1. Calcula los coeficientes de correlación
# 2. Representa los coeficientes con números
# 3. Orden alfabético de las variables
# 4. Representa solo la matriz diagonal inferior
# 5. Tamaño de las etiquetas y de los coeficientes de correlación
corrplot(cor(abalone.df), method='number', order='alphabet',
         type='lower', tl.cex=0.7, number.cex=0.7)
```

3.3.3. Diagramas de cajas

```
# Definimos un color por cada variable
var_colors<-c("red", "yellow", "orange", "green", "cyan", "blue", "pink", "purple")
# Diagrama de cajas para cada variable
# Parámetros
# 1. Conjunto de datos a representar
# 2. Representa la asimetría de los datos
# 3. Colores para cada caja
boxplot(abalone.df %>% dplyr::select(-Sex), notch=TRUE, col=var_colors)
```

3.4. Modelos de Regresión

3.4.1. Regresión Lineal Simple

```
# Primer modelo: Ring~Shell_weight
abalone.ls1 <- lm(Ring~Shell_weight, data=abalone.df)
summary(abalone.ls1)

# Segundo modelo: Ring~Diameter
abalone.ls2 <- lm(Ring~Diameter, data=abalone.df)
summary(abalone.ls2)

# Tercer modelo: Ring~Length
abalone.ls3 <- lm(Ring~Length, data=abalone.df)
summary(abalone.ls3)

# Cuarto modelo: Ring~Height
abalone.ls4 <- lm(Ring~Height, data=abalone.df)
summary(abalone.ls4)

# Quinto: Ring~Whole_weight
abalone.ls5 <- lm(Ring~Whole_weight, data=abalone.df)
summary(abalone.ls5)
```

3.4.2. Regresión Lineal Múltiple

```
# Primer modelo: todos los predictores
abalone.lm1 <- lm(Ring~., data=abalone.df)
summary(abalone.lm1)
```

```

# Segundo modelo: todos los predictores menos Length
abalone.lm2 <- lm(Ring~.-Length, data=abalone.df)
summary(abalone.lm2)

# Tercer modelo: todos los predictores menos Length y Height
abalone.lm3 <- lm(Ring~.-Length-Height, data=abalone.df)
summary(abalone.lm3)

# Cuarto modelo: todos los predictores menos Length, Height y Viscera_weight
abalone.lm4 <- lm(Ring~.-Length -Height -Viscera_weight, data=abalone.df)
summary(abalone.lm4)

# Quinto modelo: interacción entre Shell_weight y Diameter
abalone.lm5 <- lm(Ring~.-Length -Height -Viscera_weight
                  + Shell_weight*Diameter, data=abalone.df)
summary(abalone.lm5)

# Sexto modelo: eliminando la variable Sex
abalone.lm6 <- lm(Ring~.-Length -Height -Viscera_weight -Sex
                  + Shell_weight*Diameter, data=abalone.df)
summary(abalone.lm6)

# Séptimo modelo: términos cuadráticos de Shucked_weight y Whole_weight
abalone.lm7 <- lm(Ring~.-Length -Viscera_weight -Sex +
                  Shell_weight*Diameter + I(Shucked_weight^2) +
                  I(Whole_weight^2), data=abalone.df)
summary(abalone.lm7)

```

3.4.3. K Nearest Neighbours (KNN)

```

# Función para aplicar validación cruzada sobre uno de los siguientes
# algoritmos de Regresión: KNN o Regresión Lineal Múltiple.
# Para KNN también se podrá elegir entre la formulación del
# mejor modelo de Regresión Lineal o la que incluye a todos los predictores
nombre <- "abalone"
run_regr_cv <- function(i, x, alg="knn", tt = "test", formula = "all") {
  # Lectura de las particiones de train y test
  file <- paste(x, "-5-", i, "tra.dat", sep="")
  x_tra <- read.csv(paste("./abalone/", file, sep=""),
                     comment.char="@", header=FALSE)
  file <- paste(x, "-5-", i, "tst.dat", sep="")
  x_tst <- read.csv(paste("./abalone/", file, sep=""),
                     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"
  if (tt == "train") {
    test <- x_tra
  }
  else {
    test <- x_tst
  }
}

```

```

}

# 1. Entrena un modelo con KNN
if (alg == "knn") {
  # Fórmula con todos los predictores
  if (formula == "all") {
    fitMulti <- kknn(Y~., x_tra, test)
  }
  # Fórmula del mejor modelo de Regresión Lineal
  else {
    fitMulti <- kknn(Y~.-X2 -X7 -X1 + X8*X3 + I(X6^2) + I(X5^2), x_tra, test)
  }
  # Obtenemos las predicciones para KNN
  yprime <- fitMulti$fitted.values
}
# 2. Entrena un modelo con Regresión Lineal Múltiple
else if (alg == "rlm") {
  fitMulti <- lm(Y~.,x_tra)
  # Obtenemos las predicciones
  yprime <- predict(fitMulti,test)
}
sum(abs(test$Y-yprime)^2)/length(yprime) # MSE
}

# Media del error para entrenamiento y validación de los modelos de KNN
# con la fórmula del mejor modelo con Regresión Lineal Múltiple
knnMSEtrain <- mean(sapply(1:5, run_regr_cv, nombre, "knn", "train", "RLM"))
knnMSEtrain
knnMSEtest <- mean(sapply(1:5, run_regr_cv, nombre, "knn", "test", "RLM"))
knnMSEtest

```

3.5. Comparación de algoritmos de Regresión

```

# Media del error de entrenamiento y validación para los modelos de
# Regresión Lineal Múltiple considerando todas las variables
lmMSEtrain<-mean(sapply(1:5, run_regr_cv, nombre, "rlm", "train"))
lmMSEtrain
lmMSEtest<-mean(sapply(1:5, run_regr_cv, nombre, "rlm", "test"))
lmMSEtest
# Media del error de entrenamiento y validación para los modelos con
# KNN considerando todas las variables
knnMSEtrain <- mean(sapply(1:5, run_regr_cv, nombre, "knn", "train"))
knnMSEtrain
knnMSEtest <- mean(sapply(1:5, run_regr_cv, nombre, "knn", "test"))
knnMSEtest

# Leemos los resultados sobre el conjunto de entrenamiento
resultados <- read.csv("regr_train_alumnos.csv")
tablatra <- cbind(resultados[,2:dim(resultados)[2]])
colnames(tablatra) <- names(resultados)[2:dim(resultados)[2]]
rownames(tablatra) <- resultados[,1]
# Leemos los resultados sobre el conjunto de test
resultados <- read.csv("regr_test_alumnos.csv")
tablatst <- cbind(resultados[,2:dim(resultados)[2]])
colnames(tablatst) <- names(resultados)[2:dim(resultados)[2]]

```

```

rownames(tablatst) <- resultados[,1]

# Normalizamos la tabla de resultados de tests para Wilcoxon
# ""+ 0.1 porque wilcox R falla para valores == 0 en la tabla"""
difs <- (tablatst[,1] - tablatst[,2]) / tablatst[,1]
wilc_1_2 <- 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) <- c(colnames(tablatst)[1], colnames(tablatst)[2])
# Aplicamos el test de Wilcoxon sobre Regresión Lineal Múltiple y KNN
LMvsKNNtst <- wilcox.test(wilc_1_2[,1], wilc_1_2[,2],
                            alternative = "two.sided", paired=TRUE)
Rmas <- LMvsKNNtst$statistic
pvalue <- LMvsKNNtst$p.value
LMvsKNNtst <- wilcox.test(wilc_1_2[,2], wilc_1_2[,1],
                            alternative = "two.sided", paired=TRUE)
Rmenos <- LMvsKNNtst$statistic
pvalue

# Aplicamos el test de Friedman para comparar los tres algoritmos simultáneamente
test_friedman <- friedman.test(as.matrix(tablatst))
test_friedman
# Aplicamos un Post Hoc con el test de Wilcoxon y una penalización por pasos de Holm
tam <- dim(tablatst)
groups <- rep(1:tam[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tablatst), groups, p.adjust = "holm", paired = TRUE)

```

4. Apéndice de Clasificación

4.1. Análisis Exploratorio General

```

# Nombres de las variables según el fichero `vehicle.dat`
vehicle.colnames <- 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")
# Leemos el dataset del fichero `vehicle.dat`
# Parámetros
# 1. Ruta hacia el fichero
# 2. Las columnas no se encuentran en la primera fila
# 3. Ignoramos las 24 primeras filas hasta alcanzar los datos
# 4. Separador de columnas
# 5. Nombres de las columnas
vehicle.df <- read.table("./vehicle/vehicle.dat", header=FALSE, skip=24,
                           sep=",", col.names=vehicle.colnames)
# Dimensión
dim(vehicle.df)
# Resumen del dataset
summary(vehicle.df)

```

4.2. Análisis Exploratorios Univariantes

4.2.1. Variables nominales

```
# Parámetros:  
# 1. Contorno negro a las barras y la leyenda  
# 2. Fondo blanco  
# 3. Ocultamos la etiqueta del eje X y establecemos la etiqueta del eje Y  
# 4. Frecuencia de cada clase sobre sus respectivas barras  
ggplot(vehicle.df, aes(x=Class, fill=Class)) +  
  geom_bar(color="black") + theme_minimal() +  
  scale_fill_discrete(name='Vehículos') +  
  labs(x="", y="Frecuencias") +  
  geom_text(stat='count', aes(label=..count..), vjust=2)
```

4.2.2. Variables numéricas

```
# Cálculo de la varianza para cada variable numérica  
vehicle.vars <- sapply(c(1:(ncol(vehicle.df)-1)), function(x) var(vehicle.df[,x]))  
# Cálculo de la desviación típica, asimetría y curtosis  
vehicle.statistics <- data.frame(describe(vehicle.df %>% dplyr::select(-Class)))  
# Dataframe con todas las medidas estadísticas por variable numérica  
vehicle.statistics <- vehicle.statistics %>% dplyr::select(c('sd', 'skew', 'kurtosis'))  
cbind(var=vehicle.vars, vehicle.statistics)  
  
# Representar los dos siguientes gráficos en una única imagen  
par(mfrow=c(1,3))  
# Gráfico QQ para la variable dependiente 'Praxis_aspect_ratio'  
qqnorm(vehicle.df$Praxis_aspect_ratio, main="Praxis_aspect_ratio")  
qqline(vehicle.df$Praxis_aspect_ratio)  
# Gráfico QQ para la variable dependiente 'Max_length_aspect_ratio'  
qqnorm(vehicle.df$Max_length_aspect_ratio, main="Max_length_aspect_ratio")  
qqline(vehicle.df$Max_length_aspect_ratio)  
# Gráfico QQ para la variable dependiente 'Praxis_aspect_ratio'  
qqnorm(vehicle.df$Major_skewness, main="Major_skewness")  
qqline(vehicle.df$Major_skewness)  
  
# Aplicamos el test de Shapiro-Wilk's para cada variable extrayendo el p-valor  
vehicle.shapiros <- sapply(c(1:(ncol(vehicle.df)-1)),  
  function(x) shapiro.test(vehicle.df[,x])$p.value)  
# Definimos un dataframe con los resultados para cada variable  
vehicle.shapiros.df <- data.frame(Variables=colnames(  
  vehicle.df %>% dplyr::select(-Class))), PValores=vehicle.shapiros)  
vehicle.shapiros.df
```

4.3. Análisis Exploratorios Multivariantes

```
# Codificamos la variable independiente a etiquetas numéricas  
# 1:bus, 2:opel, 3:saab, 4:van  
vehicle.df$EncodedClass <- as.numeric(vehicle.df$Class)
```

4.3.1. Matriz de Puntos

```
# Primer grupo de variables para representar una matriz de puntos
vehicle.first_group <- c("Circularity", "Distance_circularity", "Praxis_aspect_ratio",
                         "Max_length_aspect_ratio", "Praxis_rectangular",
                         "Length_rectangular", "Radius_ratio", "Scatter_ratio",
                         "Hollows_ratio", "EncodedClass")

# Parámetros
# 1. Representa solo la diagonal inferior porque la matriz es simétrica
# 2. Diferencia los ejemplares por género cambiando el color y su forma
# 3. Representa curvas de densidad en la diagonal
scatterplotMatrix(vehicle.df[vehicle.first_group], diagonal=list(method="density"),
                  upper.panel=NULL, col=c("orange", "green", "blue", "pink"),
                  group=as.factor(vehicle.df$Class))

# Segundo grupo de variables para representar una segunda matriz de puntos
vehicle.second_group <- c("Major_variance", "Minor_variance", "Major_skewness",
                           "Minor_skewness", "Minor_kurtosis", "Major_kurtosis",
                           "Gyration_radius", "Compactness", "Elongatedness",
                           "EncodedClass")

# Parámetros
# 1. Representa solo la diagonal inferior porque la matriz es simétrica
# 2. Diferencia los ejemplares por género cambiando el color y su forma
# 3. Representa curvas de densidad en la diagonal
scatterplotMatrix(vehicle.df[vehicle.second_group], diagonal=list(method="density"),
                  upper.panel=NULL, col=c("orange", "green", "blue", "pink"),
                  group=as.factor(vehicle.df$Class))
```

4.3.2. Correlaciones

```
# Representa los coeficientes de correlación para el
# primer grupo de predictores y la variable independiente
# Parámetros
# 1. Calcula los coeficientes de correlación
# 2. Representa los coeficientes con números
# 3. Orden alfabético de las variables
# 4. Representa solo la matriz diagonal inferior
# 5. Tamaño de las etiquetas y de los coeficientes de correlación
corrplot(cor(vehicle.df %>% dplyr::select(vehicle.first_group)),
         method='number', order='alphabet',
         type='lower', tl.cex=0.7, number.cex=0.7)

# Representa los coeficientes de correlación para el
# segundo grupo de predictores y la variable independiente
# Parámetros
# 1. Calcula los coeficientes de correlación
# 2. Representa los coeficientes con números
# 3. Orden alfabético de las variables
# 4. Representa solo la matriz diagonal inferior
# 5. Tamaño de las etiquetas y de los coeficientes de correlación
corrplot(cor(vehicle.df %>% dplyr::select(vehicle.second_group)),
         method='number', order='alphabet',
         type='lower', tl.cex=0.7, number.cex=0.7)
```

4.3.3. Diagramas de cajas

```
# Escalamos y centramos todas las variables numéricas
vehicle.scaled.df <- data.frame(scale(vehicle.df
    %>% dplyr::select(-Class, -EncodedClass)))
# Añadimos las variables de clase y la codificada al dataset escalado
vehicle.scaled.df$Class <- vehicle.df$Class
vehicle.scaled.df$EncodedClass <- vehicle.df$EncodedClass

# Definimos un color por cada variable
var_colors<-c("red", "yellow", "orange", "green", "cyan",
              "blue", "pink", "purple", "gray")
# Diagrama de cajas para el primer grupo de predictores,
# sin la variable independiente
# Parámetros
# 1. Escala al 90% del tamaño del nombre de las variables y del gráfico
# 2. Conjunto de datos a representar
# 3. Representa la asimetría de los datos
# 4. Colores para cada caja
# 5. Esconde el eje X y personaliza el eje Y para que el gráfico sea horizontal
# y muestre los nombres de las variables para cada caja
par(cex.axis=0.9, mar=c(1, 10, 1, 1))
boxplot(vehicle.scaled.df %>% dplyr::select(vehicle.first_group, -EncodedClass),
        notch=TRUE, col=var_colors,
        xaxt="n", yaxt="n", horizontal=TRUE)
axis(2, labels=vehicle.first_group[1:(length(vehicle.first_group)-1)],
     at=1:(length(vehicle.first_group)-1), las=2)

# Diagrama de cajas para el segundo grupo de predictores,
# sin la variable independiente
# Parámetros
# 1. Escala al 90% del tamaño del nombre de las variables y del gráfico
# 2. Conjunto de datos a representar
# 3. Representa la asimetría de los datos
# 4. Colores para cada caja
# 5. Esconde el eje X y personaliza el eje Y para que el gráfico sea horizontal
# y muestre los nombres de las variables para cada caja
par(cex.axis=0.9, mar=c(1, 10, 1, 1))
boxplot(vehicle.scaled.df %>% dplyr::select(vehicle.second_group, -EncodedClass),
        notch=TRUE, col=var_colors,
        xaxt="n", yaxt="n", horizontal=TRUE)
axis(2, labels=vehicle.second_group[1:(length(vehicle.second_group)-1)],
     at=1:(length(vehicle.second_group)-1), las=2)
```

4.4. Modelos de Clasificación

4.4.1. K-Nearest Neighbors (KNN)

```
# Función que calcula la matriz de distancias entre dos conjuntos proporcionados
# y clasifica las muestras de test según la distancia a los K vecinos más
# cercanos de train.
my_knn_train_test <- function(train, train_labels, test=NA, test_labels=NA,
                                k=1, metric="euclidean") {
    # Vector de predicciones
```

```

preds <- c()
# Matriz de distancias entre cada muestra de test y todas de train
for(i in 1:dim(test)[1]) {
  dist_mat <- c()
  for(j in 1:dim(train)[1]) {
    xmat <- rbind(test[i,], train[j,])
    # Silencia la salida de la función `distance`
    # para luego imprimir la tasa de aciertos
    dist_mat <- append(dist_mat, invisible(suppressMessages(
      distance(as.data.frame(xmat), metric))))
  }
  # Obtenemos las K muestras con menor distancia
  neigs <- sort(dist_mat)[1:k]
  # Realizamos una votación para decidir la clase a la que pertenece
  # la muestra de test.
  # Para ello se realiza una media de las clases de las muestras
  # más cercanas y se redondea el valor para obtener un número entero.
  sample_class <- 0
  for(s in 1:length(neigs)) {
    pos <- which(dist_mat == neigs[s])
    sample_class <- sample_class + train_labels[pos]
  }
  preds <- append(preds, round(sample_class/k))
}
# Calculamos la tasa de aciertos comparando las predicciones con
# las etiquetas reales del conjunto de test
return (mean(preds == test_labels, na.rm=TRUE))
}

# Función que aplica el algoritmo KNN sobre un conjunto de entrenamiento y
# validación.
my_knn <- function(train, train_labels, test=NA, test_labels=NA,
                     k=1, metric="euclidean") {
  new_train<-train
  new_test<-test
  # 90%-10% en caso de que no se haya especificado un conjunto de test
  if (is.na(test)) {
    shuffle_train <- sample(dim(train)[1])
    pct90 <- (dim(train)[1] * 90) %/% 100
    new_train <- train[shuffle_train[1:pct90], ]
    train_labels <- train_labels[shuffle_train[1:pct90]]
    new_test <- train[shuffle_train[(pct90+1):dim(train)[1]], ]
    test_labels <- train_labels[shuffle_train[(pct90+1):dim(train)[1]]]
  }
  # 1. Entrenamiento y validación sobre el mismo conjunto de train
  train_prec <- my_knn_train_test(new_train, train_labels,
                                    new_train, train_labels, k, metric)
  # 2. Entrenamiento con train y validación con test
  test_prec <- my_knn_train_test(new_train, train_labels,
                                 new_test, test_labels, k, metric)
  return (c(train_prec, test_prec))
}

```

```

# Primer modelo KNN con distancia Manhattan y K=3
vehicle.knn1 <- my_knn(vehicle.scaled.df %>% dplyr::select(-Class, -EncodedClass), vehicle.scaled.df$EncodedClass)
vehicle.knn1
# Segundo modelo KNN con distancia Manhattan y K=21
vehicle.knn2 <- my_knn(vehicle.scaled.df %>% dplyr::select(-Class, -EncodedClass), vehicle.scaled.df$EncodedClass)
vehicle.knn2
# Tercer modelo KNN con distancia Jaccard y K=3
vehicle.knn3 <- my_knn(vehicle.scaled.df %>% dplyr::select(-Class, -EncodedClass), vehicle.scaled.df$EncodedClass)
vehicle.knn3
# Cuarto modelo KNN con distancia Jaccard y K=21
vehicle.knn4 <- my_knn(vehicle.scaled.df %>% dplyr::select(-Class, -EncodedClass), vehicle.scaled.df$EncodedClass)
vehicle.knn4

```

4.4.2. Linear Discriminant Analysis (LDA)

```

# Aplicamos el test de Levene para verificar la covarianza
for (predictor in colnames(vehicle.scaled.df) %>% dplyr::select(-EncodedClass, -Class)) {
  cat("\n", predictor, "\n")
  print(leveneTest(vehicle.scaled.df[[predictor]] ~ Class, vehicle.scaled.df))
}

# Dividimos el conjunto de datos en train y test (90%-10%)
vehicle.shuffle.train <- sample(nrow(vehicle.scaled.df))
vehicle.pct90 <- (nrow(vehicle.scaled.df) * 90) %% 100
vehicle.scaled.train <- vehicle.scaled.df[vehicle.shuffle.train[1:vehicle.pct90], ]
vehicle.scaled.test <- vehicle.scaled.df[vehicle.shuffle.train[(vehicle.pct90+1):nrow(vehicle.scaled.df)], ]

# Primer modelo con LDA
vehicle.lda1 <- lda(Class ~ .-EncodedClass, data=vehicle.scaled.train)
# Obtenemos las predicciones sobre train y test
vehicle.lda1.train.preds <- predict(vehicle.lda1, vehicle.scaled.train)
vehicle.lda1.test.preds <- predict(vehicle.lda1, vehicle.scaled.test)
# Calculamos la precisión para train y test
mean(vehicle.lda1.train.preds$class == vehicle.scaled.train$Class)
mean(vehicle.lda1.test.preds$class == vehicle.scaled.test$Class)
table(vehicle.lda1.test.preds$class, vehicle.scaled.test$Class)

```

4.4.3. Quadratic Discriminant Analysis (QDA)

```

# Primer modelo con QDA
vehicle.qda1 <- qda(Class ~ .-EncodedClass, data=vehicle.scaled.train)
# Obtenemos las predicciones sobre train y test
vehicle.qda1.train.preds <- predict(vehicle.qda1, vehicle.scaled.train)
vehicle.qda1.test.preds <- predict(vehicle.qda1, vehicle.scaled.test)
# Calculamos la precisión para train y test
mean(vehicle.qda1.train.preds$class == vehicle.scaled.train$Class)
mean(vehicle.qda1.test.preds$class == vehicle.scaled.test$Class)
table(vehicle.qda1.test.preds$class, vehicle.scaled.test$Class)

```

4.5. Comparación de algoritmos de Clasificación

```

# Función para aplicar validación cruzada sobre uno de los siguientes
# algoritmos de Clasificación: KNN, LDA o QDA.
nombre <- "vehicle"
run_clasif_cv <- function(i, x, alg="knn", tt = "test") {
  # Lectura de las particiones de train y test
  file <- paste(x, "-10-", i, "tra.dat", sep="")
  x_tra <- read.csv(paste("./vehicle/", file, sep=""),
                     comment.char="@", header=FALSE)
  file <- paste(x, "-10-", i, "tst.dat", sep="")
  x_tst <- read.csv(paste("./vehicle/", file, sep=""),
                     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"
  if (tt == "train") {
    test <- x_tra
  }
  else {
    test <- x_tst
  }
  # 1. Entrena un modelo con KNN
  if (alg == "knn") {
    fitMulti <- kknn(Y~., x_tra, test)
    # Obtenemos las predicciones para KNN y su tasa de aciertos
    yprime <- fitMulti$fitted.values
    return (mean(yprime == test$Y, na.rm=TRUE))
  }
  # 2. Entrena un modelo con LDA
  else if (alg == "lda") {
    fitMulti <- lda(Y~.,x_tra)
    # Obtenemos las predicciones
    yprime <- predict(fitMulti,test)
  }
  # 3. Entrena un modelo con QDA
  else if (alg == "qda") {
    fitMulti <- qda(Y~.,x_tra)
    # Obtenemos las predicciones
    yprime <- predict(fitMulti,test)
  }
  return (mean(yprime$class == test$Y, na.rm=TRUE)) # Tasa de aciertos
}
# Media del error de entrenamiento y validación para los modelos de
# KNN considerando todas las variables
knnPrecTrain<-mean(sapply(1:10, run_clasif_cv, nombre, "knn", "train"))
knnPrecTrain
knnPrecTest<-mean(sapply(1:10, run_clasif_cv, nombre, "knn", "test"))
knnPrecTest
# Media del error de entrenamiento y validación para los modelos de
# LDA considerando todas las variables
ldaPrecTrain<-mean(sapply(1:10, run_clasif_cv, nombre, "lda", "train"))
ldaPrecTrain

```

```

ldaPrecTest<-mean(sapply(1:10, run_clasif_cv, nombre, "lda", "test"))
ldaPrecTest
# Media del error de entrenamiento y validación para los modelos de
# QDA considerando todas las variables
qdaPrecTrain<-mean(sapply(1:10, run_clasif_cv, nombre, "qda", "train"))
qdaPrecTrain
qdaPrecTest<-mean(sapply(1:10, run_clasif_cv, nombre, "qda", "test"))
qdaPrecTest

# Leemos los resultados sobre el conjunto de entrenamiento
resultados <- read.csv("clasif_train_alumnos.csv")
tablatra <- cbind(resultados[,2:dim(resultados)[2]])
colnames(tablatra) <- names(resultados)[2:dim(resultados)[2]]
rownames(tablatra) <- resultados[,1]
# Leemos los resultados sobre el conjunto de test
resultados <- read.csv("clasif_test_alumnos.csv")
tablatst <- cbind(resultados[,2:dim(resultados)[2]])
colnames(tablatst) <- names(resultados)[2:dim(resultados)[2]]
rownames(tablatst) <- resultados[,1]
# Aplicamos el test de Friedman para comparar los tres algoritmos simultáneamente
test_friedman <- friedman.test(as.matrix(tablatst))
test_friedman
# Aplicamos un Post Hoc con el test de Wilcoxon y una penalización por pasos de Holm
tam <- dim(tablatst)
groups <- rep(1:dim(tablatst)[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tablatst), groups, p.adjust = "holm", paired = TRUE)

```