UNIVERSIDAD
DE GRANADA

Introducción a la Ciencia de Datos
máster ciencia de datos e ingeniería de computadores

Trabajo teórico/práctico

Análisis de datos, Regresión y Clasificación

Autor
Ignacio Vellido Expósito
ignaciove@correo.ugr.es

Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

Curso 2020-2021

Índice

1. Regresión: Análisis Estadístico de Datos

1.1. Introducción

Para el problema de regresión hacemos uso del dataset **autoMPG6** [1], donde se codifica el consumo de gasolina de distintos coches (en millas por galón, Mpg) en base a las siguientes características:

1. **Displacement**: Indica la cilindrada del coche, la suma del volumen útil de los cilindros del motor, medido en pulgadas cúbicas.
2. **Horse_power**: Mide la potencia del coche.
3. **Weight**: Peso en libras.
4. **Acceleration**: Aceleración del coche de 0 a 60 millas por hora, medido en segundos.
5. **Model_year**: Indica las dos últimas cifras del año de producción.

El objetivo es poder predecir con estos cinco atributos el consumo de Mpg de un nuevo coche:

6. **Mpg**: Millas-por-galón, indica la cantidad de galones (1G $\approx$ 3,78L) de fuél que consume un vehículo al recorrer una milla (1m $\approx$ 1,6km).

El dataset contiene 392 instancias codificando esta información.

La descripción del problema nos da alguna información adicional sobre las variables:

1. **Displacement**: Variable numérica continua, contamos con valores reales en el rango [68.0,455.0].
2. **Horse_power**: Variable numérica continua, contamos con valores enteros en el rango [46,230].
3. **Weight**: Variable numérica continua, contamos con valores enteros en el rango [1613,5140].
4. **Acceleration**: Variable numérica continua, contamos con valores reales en el rango [8.0,24.8].
5. **Model_year**: Variable numérica discreta, contamos con valores enteros en el rango [70,82].
6. **Mpg**: Variable numérica continua, contamos con valores reales en el rango [9.0,46.6].

Hipótesis de partida

- **H.1**: Horse_power puede influir en Mpg: A más potencia, más consumo.
- **H.2**: Weight debe influir en Mpg: Un coche más pesado debería consumir más.
- **H.3**: Debería haber correlación entre displacement (cilindrada) con horse y acceleration

- **H.4**: Horse y acceleration podrían estar relacionadas
- **H.5**: Viendo que contamos con un rango pequeño de años, no debería haber un cambio significativo de prestaciones entre años
- **H.6**: Pero debería existir una tendencia de mejora de prestaciones con los años, incluyendo aumento de Displacement, Horse_power y Acceleration.
- **H.7**: Model_year podría no mostrar relación con Mpg: Pese al paso de los años si contamos con diferentes tipos de vehículos (todoterrenos, familiares, deportivos...) podría haber un consumo dispar. (Si existiera tendencia, viendo que los años son de las últimas décadas del siglo XX, podría ir el consumo hacia abajo)
- **H.8**: Esta última hipótesis se puede aplicar al resto de variables, indicándonos que Model_year no debería tener relevancia para este problema de regresión.
- **H.9**: Horse_power podría depender de las variables Displacement y Weight

1.2. Análisis Estadístico de Datos

Antes de comenzar a analizar las variables nos plantemos una cuestión: ¿Debemos considerar Model_year como una variable numérica o como un factor categórico? Aunque por la hipótesis H.7 podríamos acabar no eligiendo la variable para el problema, es necesario preguntarnos por esto antes de comenzar.

Sabemos que las observaciones para esta variable cuenta con valores entre 72 y 82, por lo que tenemos información exacta del año (en comparación, por ejemplo, con agrupaciones mayores como la década o el siglo). El hecho de tratarla como categórica o cuantitativa depende mucho del problema. En este caso, tenemos interés en cuestionarnos por valores entre años, por ejemplo, el consumo entre los años 75 y 76.

Por tanto, de cara al problema de regresión que nos atañe, tendríamos dos opciones:

- Mantenerlo como categórico y generar variables dummy (valores 0-1 para indicar si la instancia es de ese año). Suponiendo que tenemos al menos una instancia de cada año, esto nos generaría 12 variables nuevas.
- Mantenerlo como numérico, pero teniendo cuidado de cómo interpretar el año.

Proseguimos con tanto dejando Model_year como variable numérica.

1.2.1. Análisis univariable

La cabecera de nuestro dataset tiene esta forma:

Displacement	Horse_power	Weight	Acceleration	Model_year	Mpg
91	70	1955	20.5	71	26.0
232	100	2789	15.0	73	18.0
350	145	4055	12.0	76	13.0
318	140	4080	13.7	78	17.5
113	95	2372	15.0	70	24.0
97	60	1834	19.0	71	27.0

Y con la siguiente información estadística:

```
 Displacement    Horse_power       Weight      Acceleration     Model_year
Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00   Min.   :70.00
1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00
Median :151.0   Median : 93.5   Median :2804   Median :15.50   Median :76.00
Mean   :194.4   Mean   :104.5   Mean   :2978   Mean   :15.54   Mean   :75.98
3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00
Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80   Max.   :82.00
     Mpg
Min.   : 9.00
1st Qu.:17.00
Median :22.75
Mean   :23.45
3rd Qu.:29.00
Max.   :46.60
```

El dataset **no** cuenta con **valores repetidos** ni **missing values**.

Mostramos scatterplots univariables:

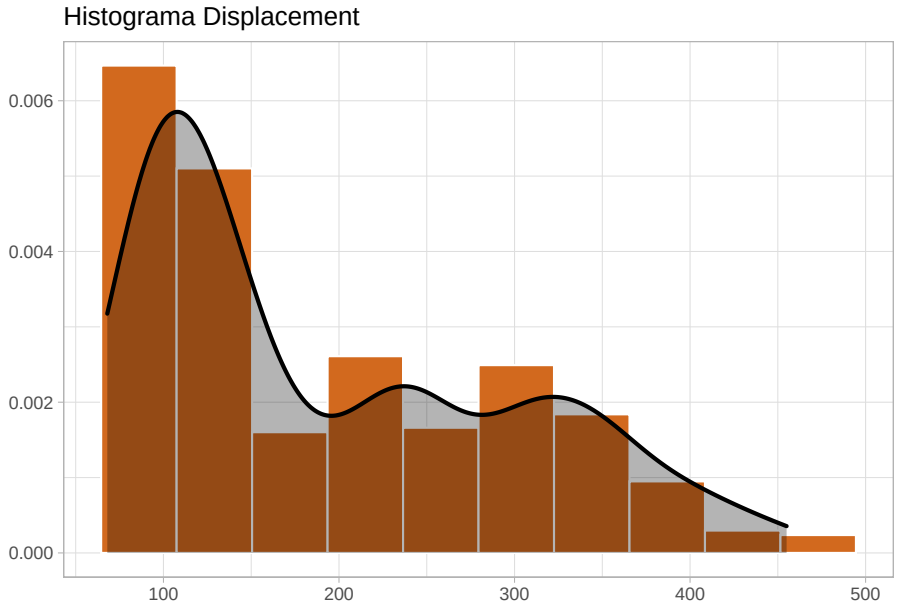


Figura 1

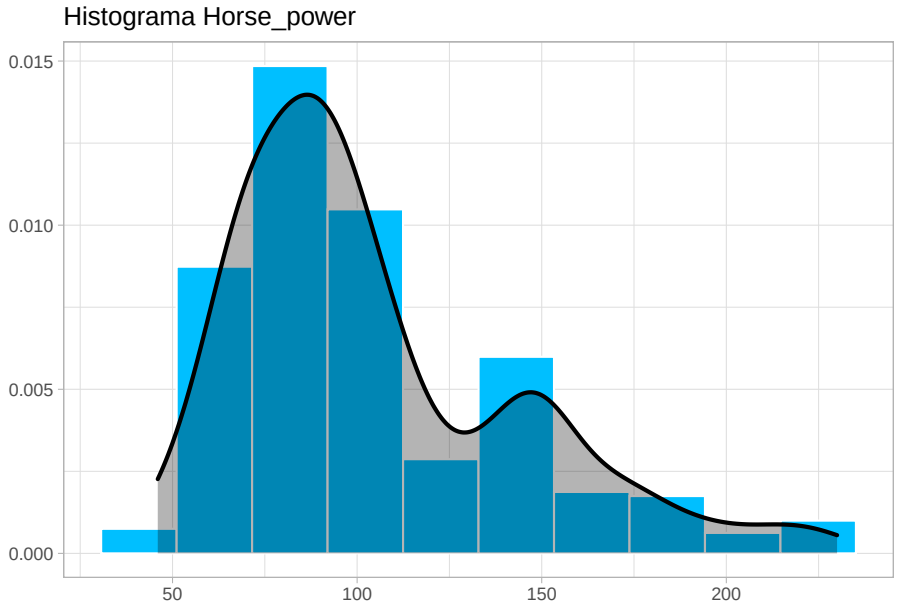


Figura 2

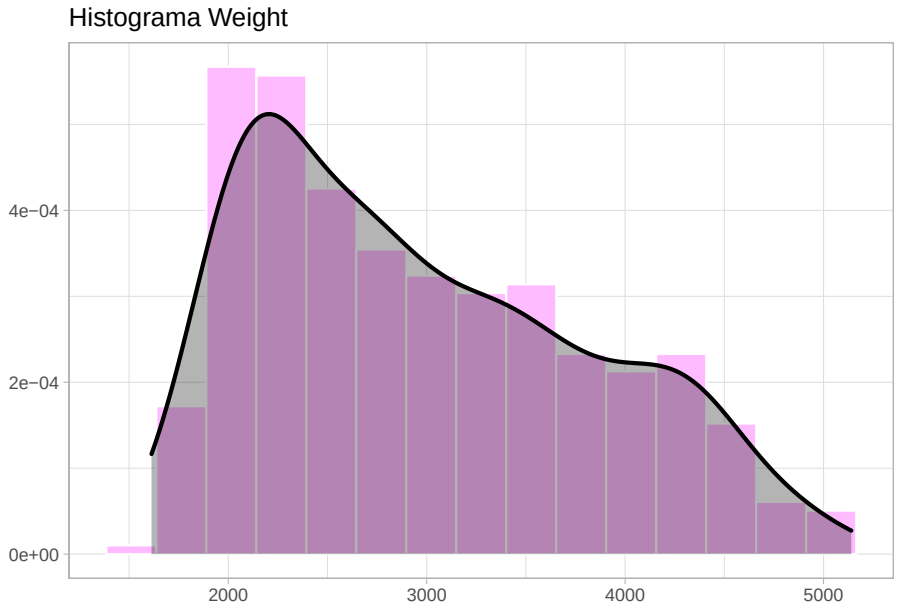


Figura 3

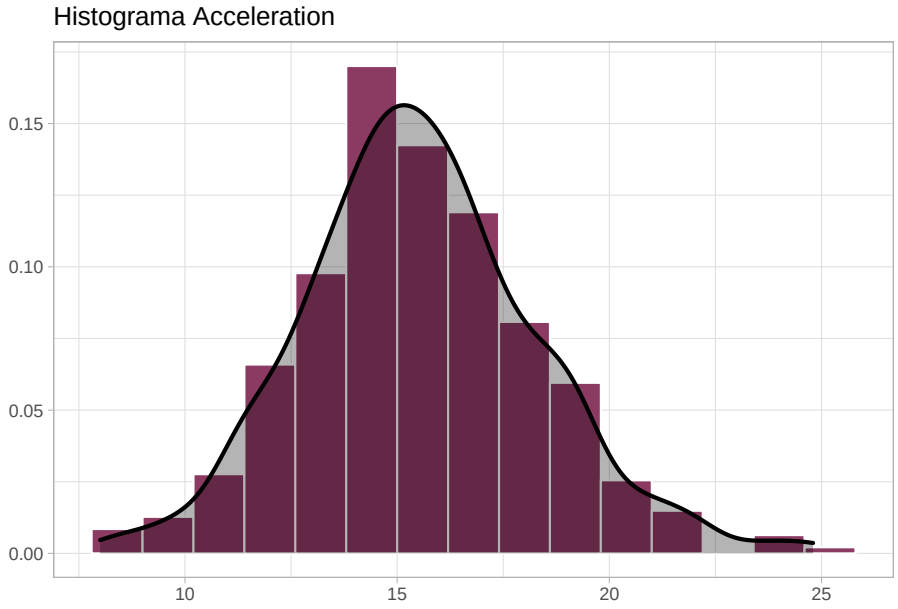


Figura 4

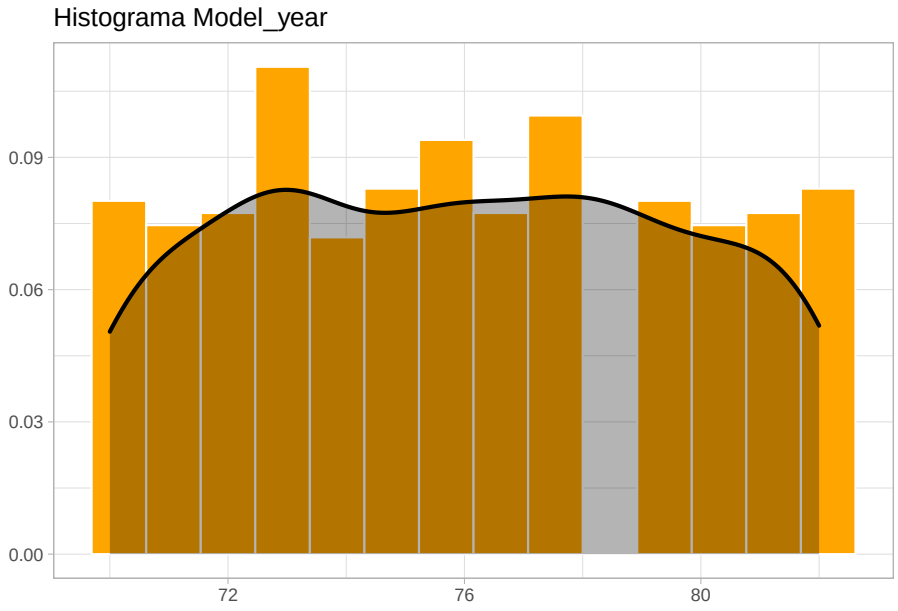


Figura 5

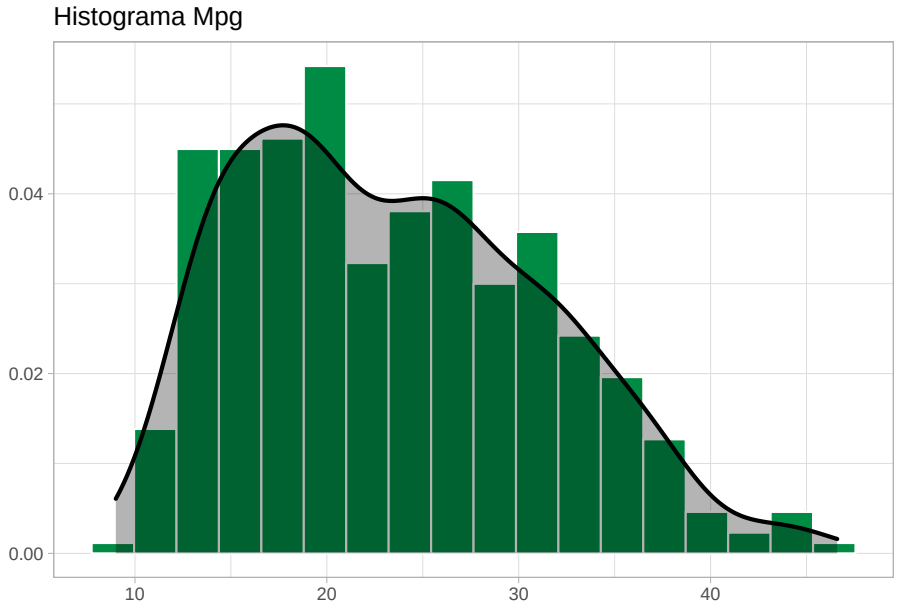


Figura 6

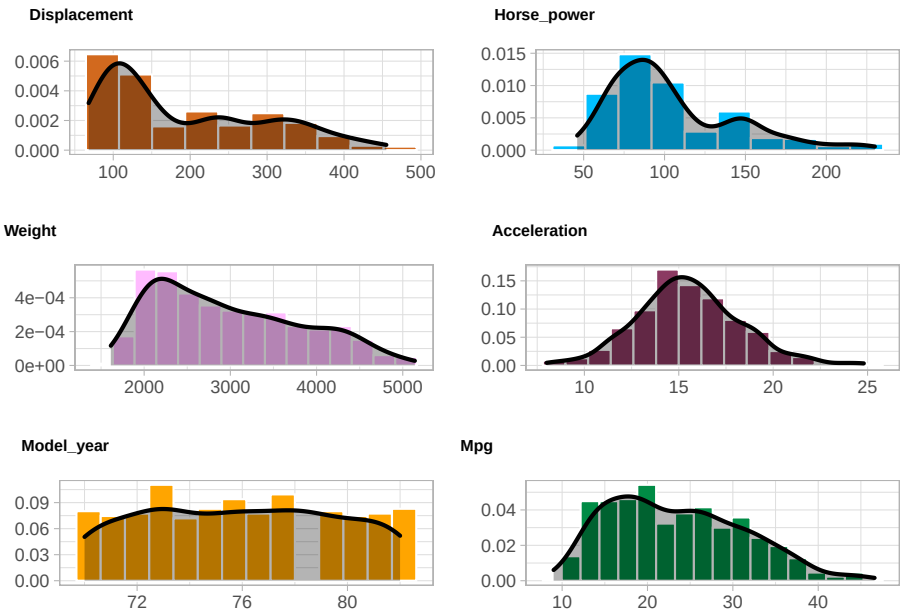


Figura 7

Y boxplots sobre las distribuciones de los datos:

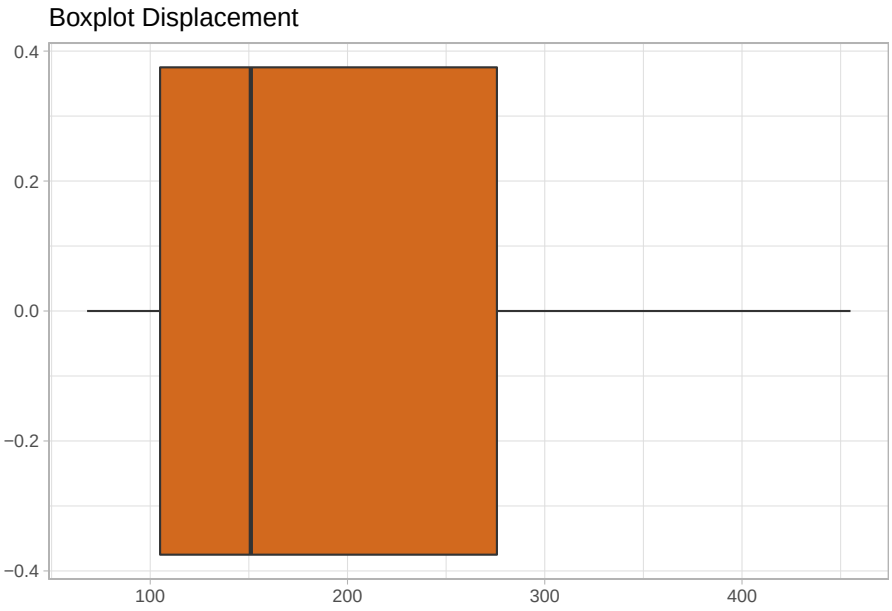


Figura 8

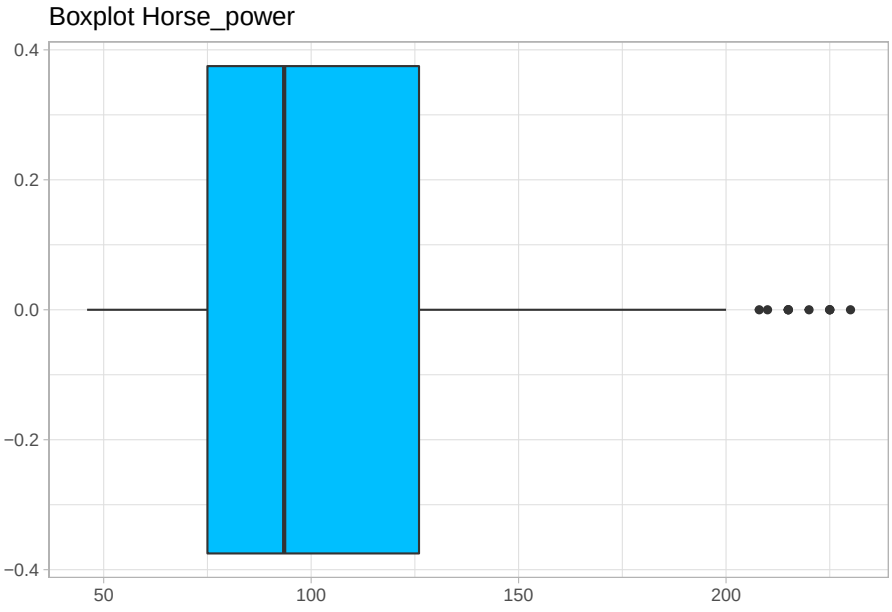


Figura 9

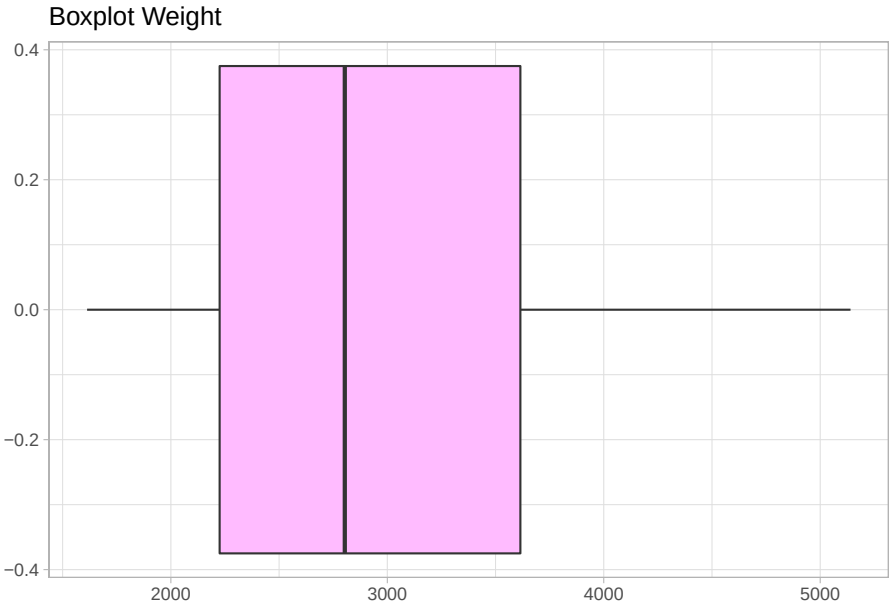


Figura 10

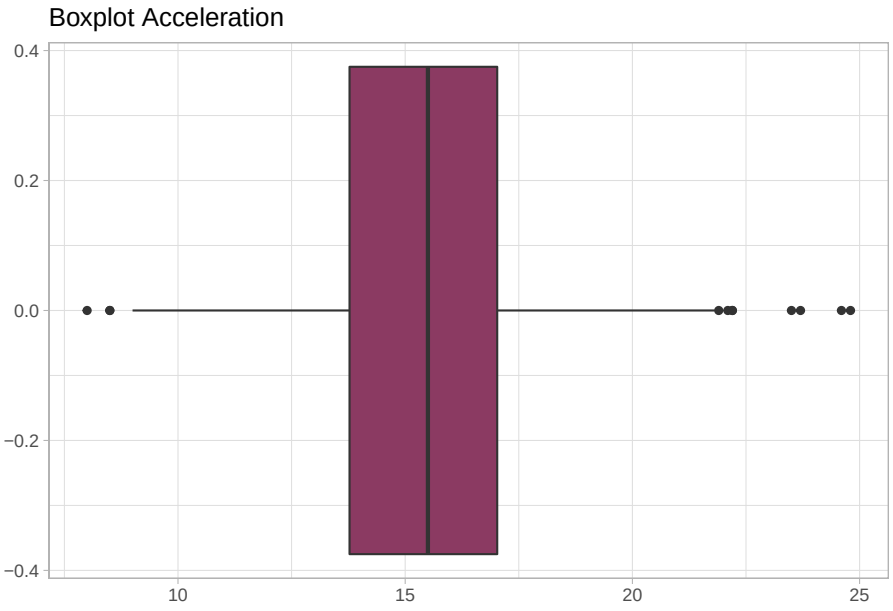


Figura 11

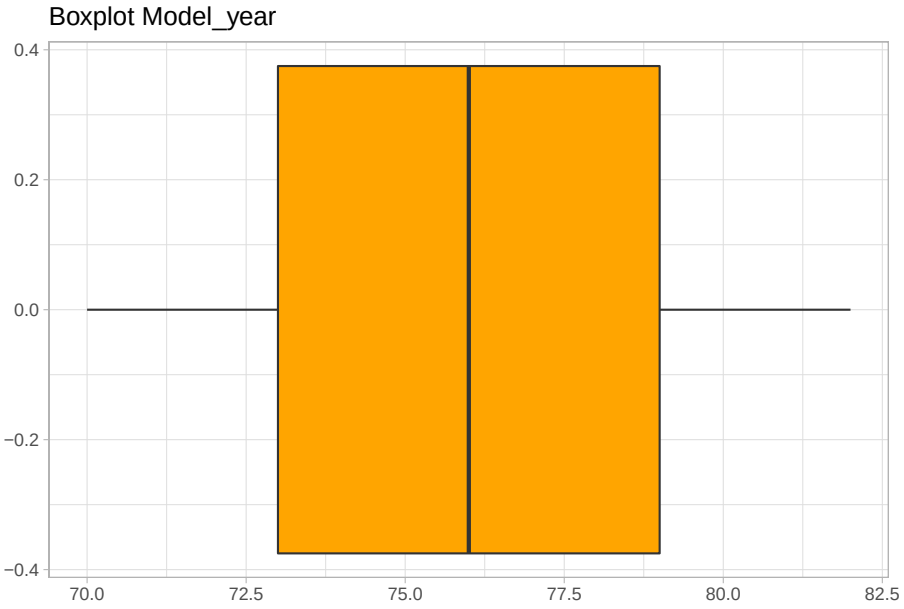


Figura 12

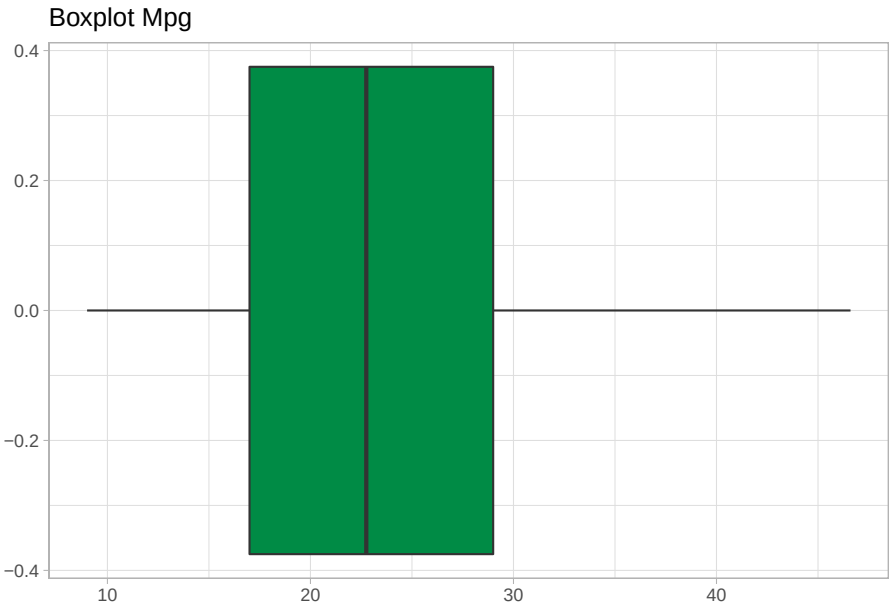


Figura 13

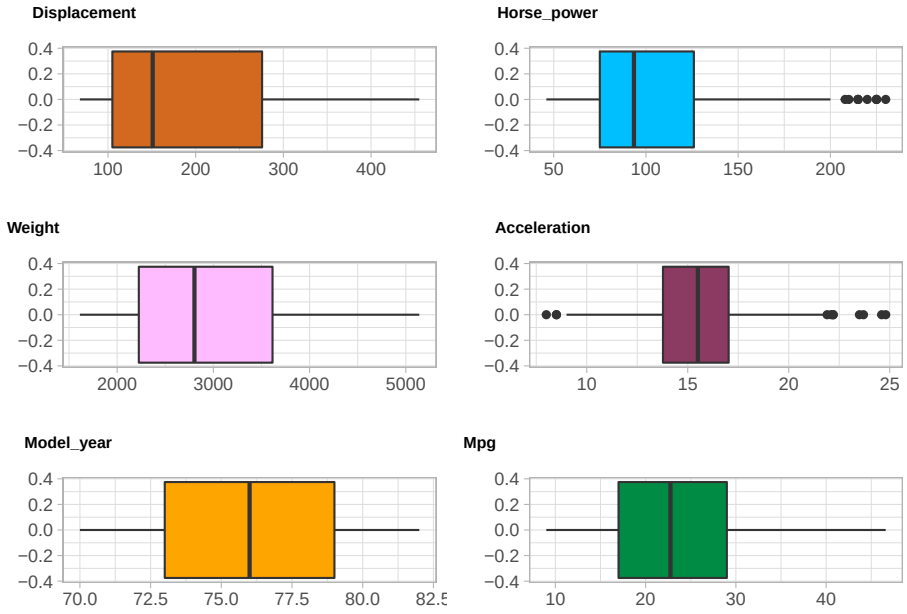


Figura 14

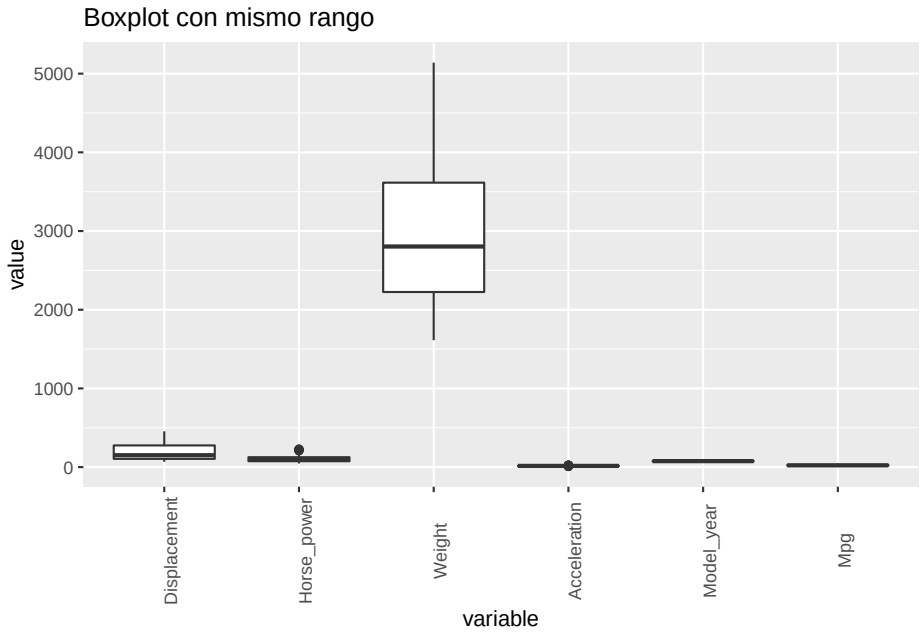


Figura 15

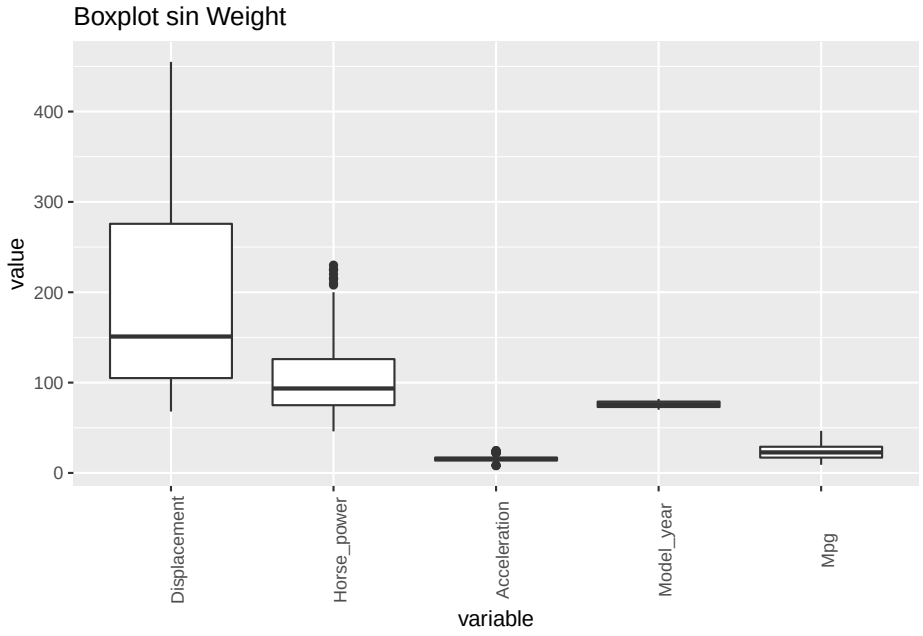


Figura 16

Ya la descripción del problema nos lo decía, los rangos en los que se distribuyen los datos son muy diferentes dependiendo de la variable. Se pueden estandarizar los datos para solucionar este problema, aunque para regresión lineal no es necesario (sí lo es para KNN) Podemos comparar los rangos intercuartiles si estandarizamos antes el dataset

```
Displacement  Horse_power       Weight Acceleration   Model_year          Mpg
    1.631723     1.324980     1.635856     1.178021     1.628781     1.537475
```

También podemos ver la distancia entre mínimos y máximos

```
Displacement  Horse_power       Weight Acceleration   Model_year          Mpg
    3.698253     4.780318     4.152330     6.089463     3.257562     4.817420
```

Displacement La cilindrada vemos que cuenta con una desviación grande y una gran concentración en los valores inferiores. Desviado a la izquierda, no parece seguir una distribución normal. Existe una alta concentración en torno al valor 125, muy por encima del recuento que alcanzan el resto de valores.

Horse_power Similar a Displacement pero tiene mayor dispersión y algunos valores muy altos. A día de hoy los coches suelen rondar los 120 en turismos y los 200 en SUVs. Aquí contamos con predominancia en el rango aproximado [70, 125] con algunas instancias por encima de los 200. Desviado a la izquierda, tampoco parece seguir una distribución normal.

Weight Una distribución más achatada que las anteriores, también ladeada hacia la izquierda. Cuenta con un rango mayor.

Acceleration Valores altamente concentrados pero en general con un rango grande. Su forma se asemeja a una distribución normal.

Model_year Aunque no se vea bien en las gráficas, contamos con valores de todos los años, más o menos equitativamente:

```
Años:   70 71 72 73 74 75 76 77 78 79 80 81 82
Conteo: 29 27 28 40 26 30 34 28 36 29 27 28 30
```

1.2.2. Análisis sobre las distribuciones

Hemos comentado antes que no apreciamos semejanzas con una distribución normal en algunas de las variables, lo comprobamos con un test estadístico (Shapiro-Wilk test):

vars	statistic	p_value	sample
Displacement	0.8818359	0.0000000	392
Horse_power	0.9040975	0.0000000	392
Weight	0.9414661	0.0000000	392
Acceleration	0.9918671	0.0305289	392
Model_year	0.9469666	0.0000000	392
Mpg	0.9671696	0.0000001	392

El test de Shapiro nos asegura con bastante certeza que ninguna variable sigue una distribución normal, aunque en menor grado en Acceleration (sigue siendo al 97 % de confianza).

Para los modelos de regresión que vamos a usar aún así la normalidad de los datos no es necesaria.

Se muestra aquí como no hay que dejarse engañar por los gráficos, puesto que Acceleration parecía seguirla. El p-value de Acceleration está muy cerca del umbral (0.03 vs 0.05). Es bastante probable de que la parte central derecha de la distribución sea la causante de no asegurar la normalidad.

Mostramos con gráficos Q-Q cómo se separan las distribuciones de su supuesta normal:

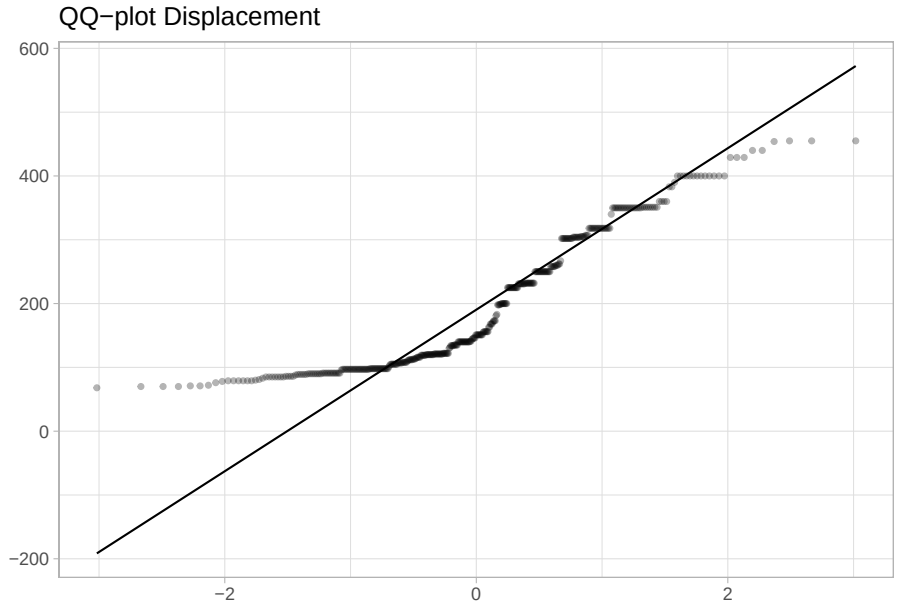


Figura 17

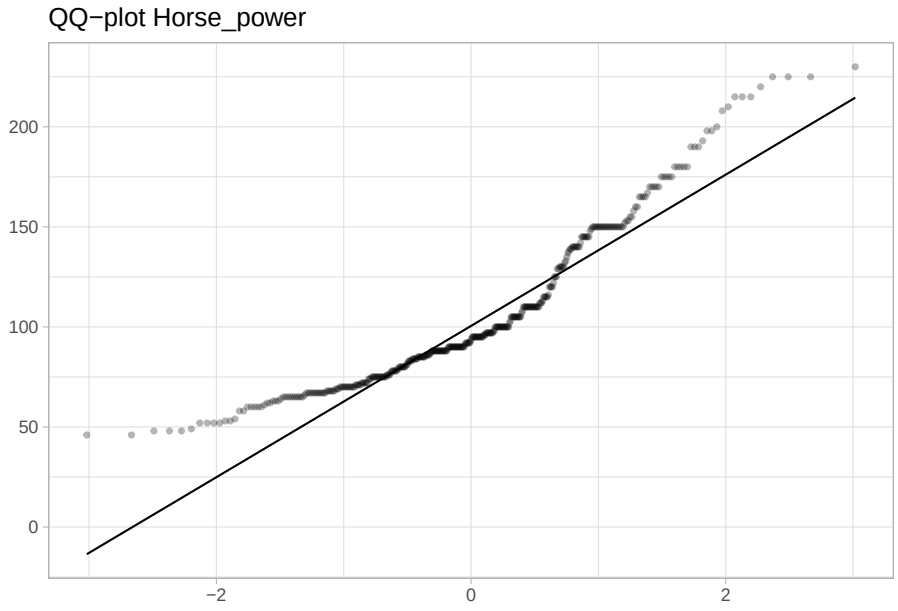


Figura 18

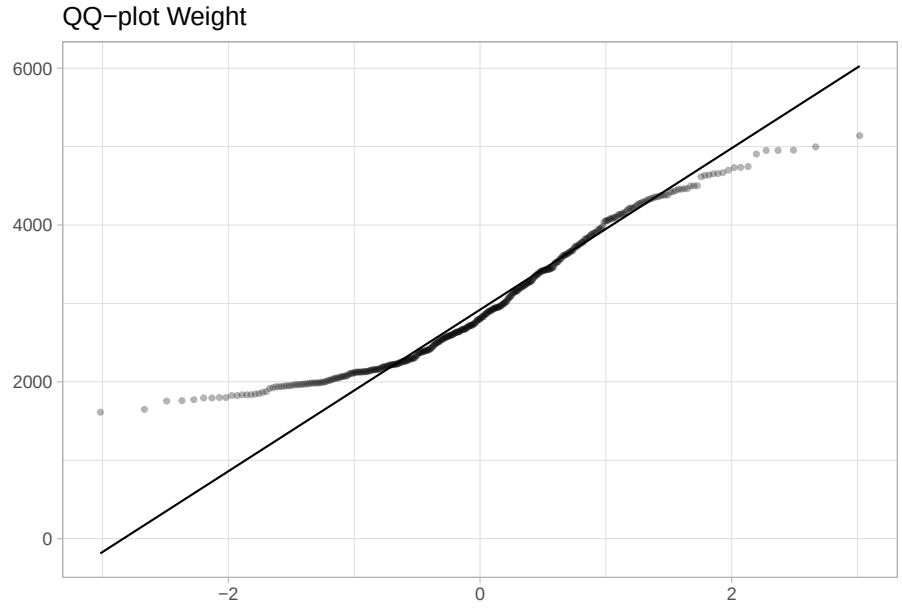


Figura 19

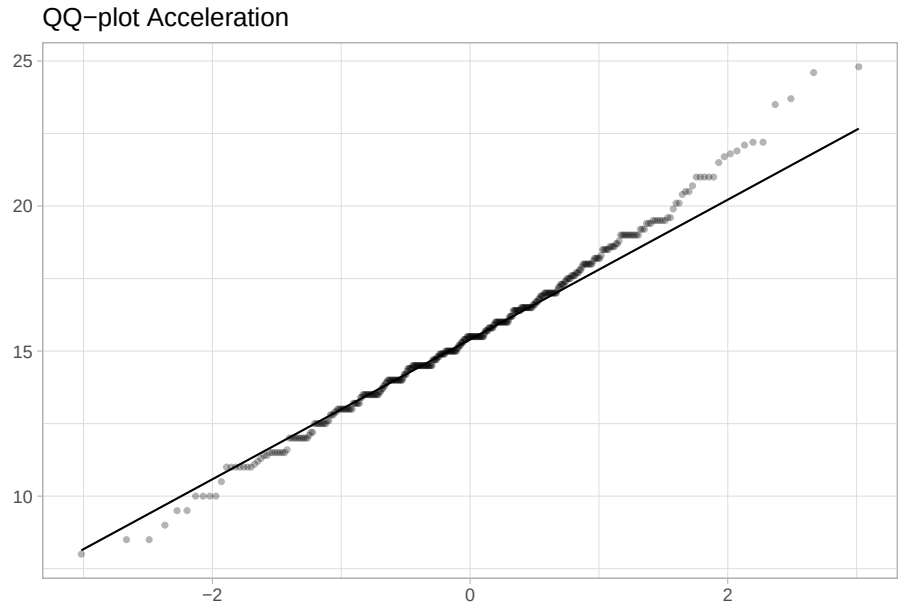


Figura 20

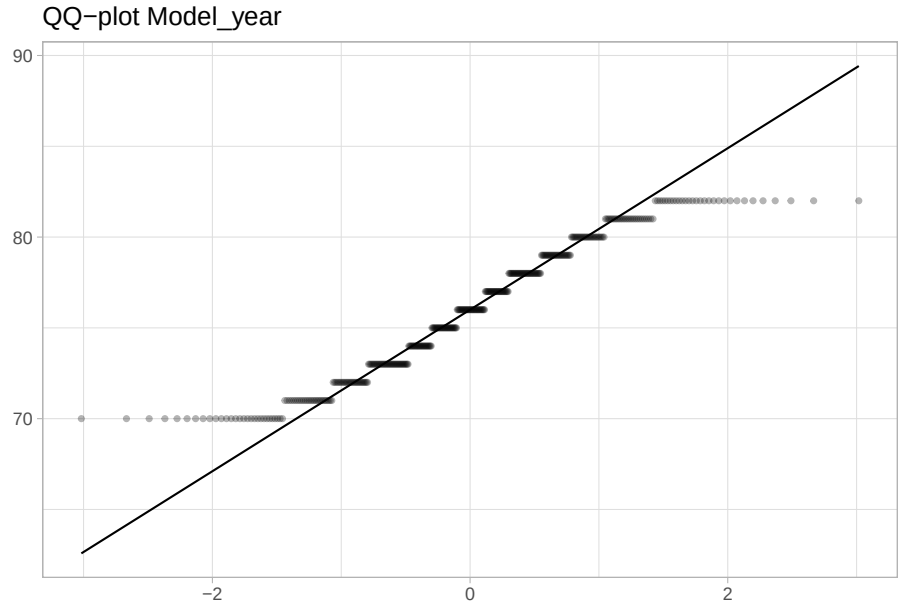


Figura 21

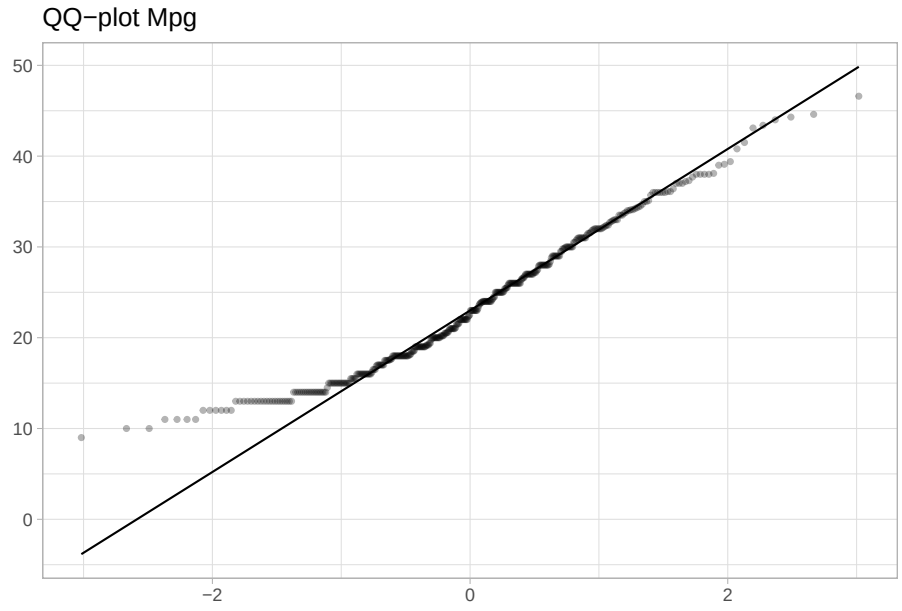


Figura 22

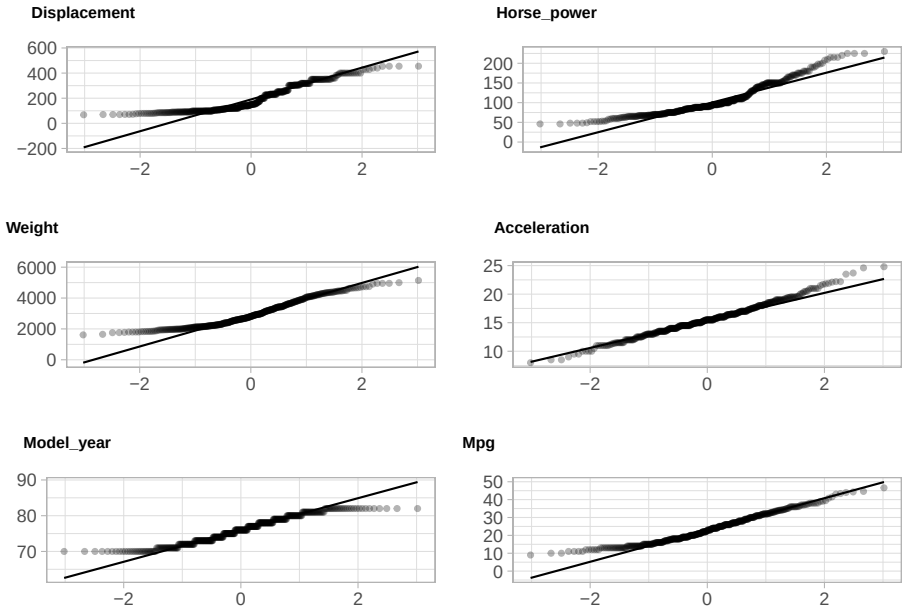


Figura 23

Estos gráficos Q-Q nos muestran más claramente que las variables no siguen distribuciones normales. La distribución de Acceleration es la que más se asemeja y eso lo vemos en el estadístico de Shapiro, pero en la cola superior existe una diferencia significativa que hace que el test rechace.

Skewness Contamos con 3 variables de skewness, todas positivas (hacia la izquierda):

```
Displacement:  0.6989813
Horse_power:  1.083161
Weight:  0.5175953

Adicionalmente, se muestra:
Mpg:  0.4553414
```

Los plots nos han dado idea de que Mpg tiene cierta skewness, pero cae por debajo del umbral de 0.5.

1.2.3. Transformaciones

En general no consideramos que sea necesaria ninguna transformación para el dataset con el que contamos. Tampoco vemos necesario crear variables nuevas a partir de las vistas, puesto que por el conocimiento que tenemos del problema parece que las variables son coherentes.

Las transformaciones necesarias para pasar a una distribución normal dependen de la variable en cuestión. Primero deberíamos averiguar que tipo de distribución siguen. Pese a ello, tal y como se ha comentado anteriormente, los métodos utilizados para regresión

(regresión lineal y KNN) no asumen ninguna forma para la distribución de los datos, por lo que no es necesario aplicar nada.

Adicionalmente, aunque para regresión lineal tampoco es absolutamente necesario, podemos estandarizar los datos a media 0 y desviación típica 1, facilitando un poco los cálculos. La inferencia estadística de la regresión no variaría, pero deberíamos tener cuidado a la hora de interpretar los resultados para no confundirnos.

1.2.4. Anomalías

Como hemos visto anteriormente en los boxplots, las únicas variables con valores muy alejados del centro de la distribución son Acceleration y Horse_power.

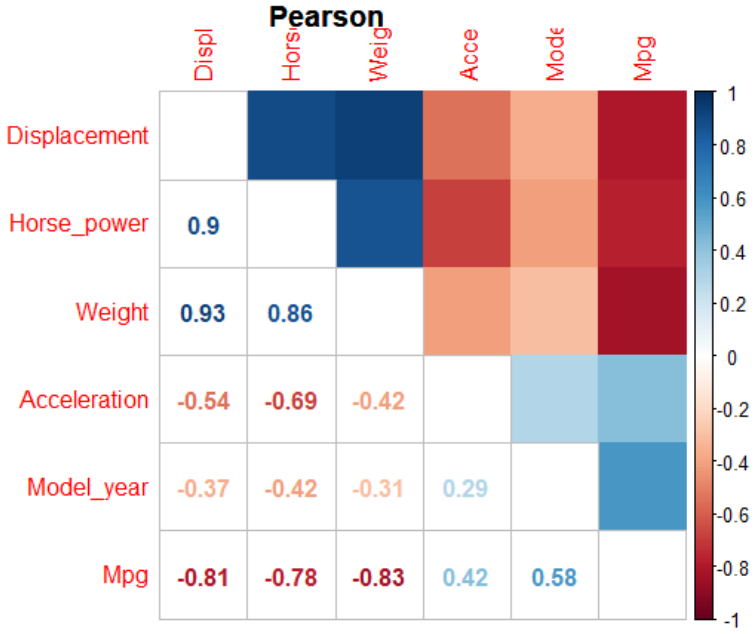
Por el significado del problema, probablemente estos posibles outliers correspondan a coches de alta gama o potentes en la época. Esto tampoco lo podemos asegurar puesto que no contamos con las características suficientes, pero se considera un razonamiento coherente. Además, puesto que los valores caen dentro de los rangos posibles para coches de la época, podemos descartar que sean errores de medida.

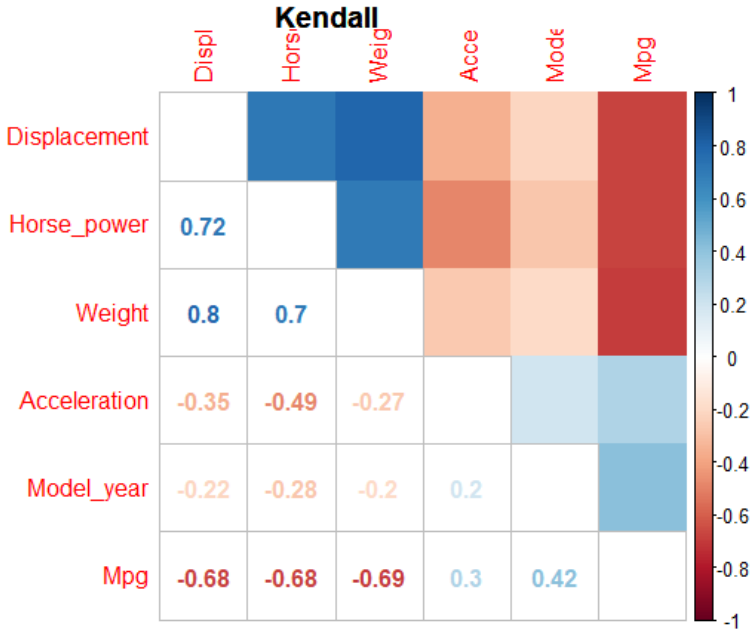
Deberíamos decidir si mantener o no estas instancias. Como en nuestro caso se nos ha pedido predecir el consumo Mpg, sin darnos consideraciones sobre los tipos/gamas de coches a los que se enfoca, proseguimos dejándo estas filas.

1.2.5. Análisis de correlación

Tenemos que tener en cuenta que las variables no siguen distribuciones normales. Aunque el coeficiente de Pearson no asume normalidad (si asume varianza y covarianza finitas), podemos adicionalmente usar el coeficiente de Kendall. Independientemente del método usado vamos a obtener las mismas correlaciones en este dataset, solo varía el valor de fuerza con la que se dan.

Para regresión la correlación en los datos no es preocupante. Al contrario, podría haber información (poca, pero alguna cantidad) que se aporte y nos ayude en el problema. En el peor de los casos, la propia metodología de selección de variables en el modelo multivariable nos ayudará a descartar aquellas variables que no sean necesarias como regresor.


Pearson
Displ
Horse
Weig
Acce
Mode
Mpg
Displacement
Horse_power
Weight
Acceleration
Model_year
Mpg
0.9
0.93 0.86
-0.54 -0.69 -0.42
-0.37 -0.42 -0.31 0.29
-0.81 -0.78 -0.83 0.42 0.58
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


Kendall
Displ
Horse
Weig
Acce
Mode
Mpg
Displacement
Horse_power
Weight
Acceleration
Model_year
Mpg
0.72
0.8 0.7
-0.35 -0.49 -0.27
-0.22 -0.28 -0.2 0.2
-0.68 -0.68 -0.69 0.3 0.42
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

Estas gráficas nos dicen que existe una alta correlación en el dataset, generalmente entre todas las variables (a excepción de Model_year), pero extremadamente fuerte en las parejas:

1. Horse_power & Displacement
2. Weight & Displacement
3. Weight & Horse_power
4. Acceleration & Horse_power
5. Mpg & Horse_power
6. Mpg & Displacement
7. Mpg & Weight

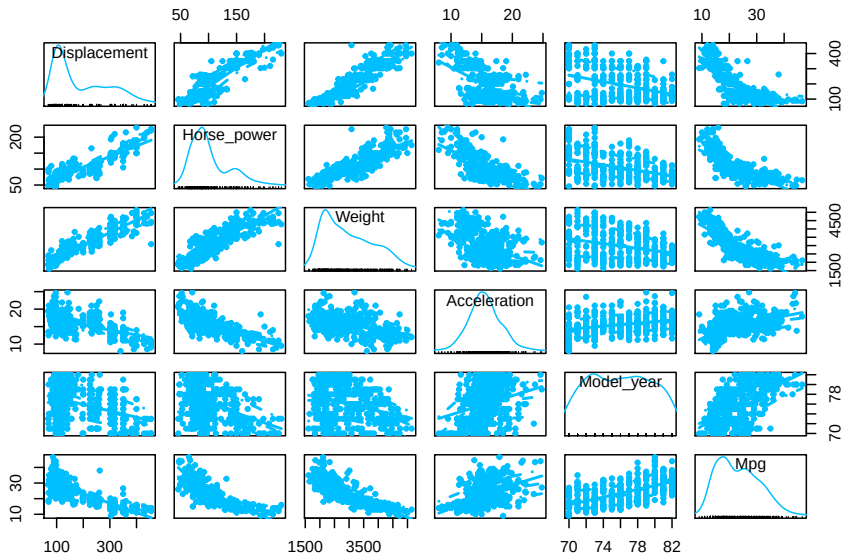


Figura 24

El scatterplot anterior nos muestra mejor la forma de estas correlaciones. Vemos que en todos los casos en los que se da una correlación positiva existe una tendencia lineal entre los datos de ambas variables, y en las negativas una tendencia logarítmica.

Vamos a mostrar algunas parejas con correlación positiva:

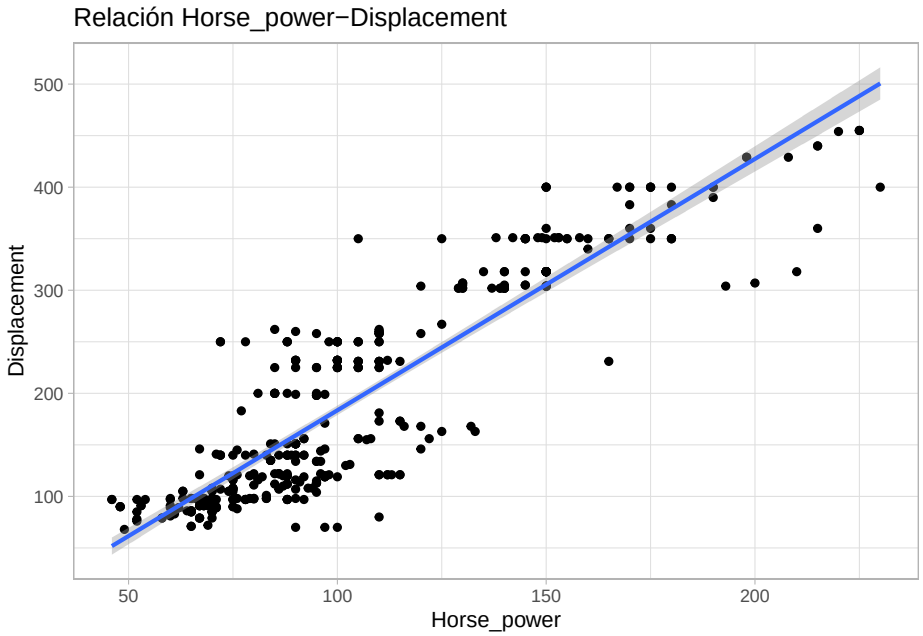


Figura 25

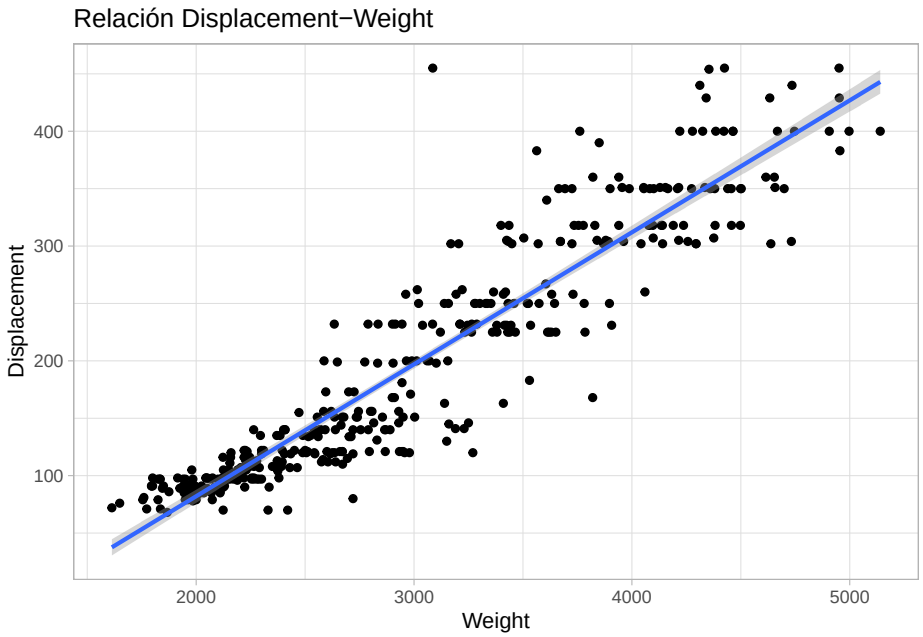


Figura 26

Y otras con correlación negativa:

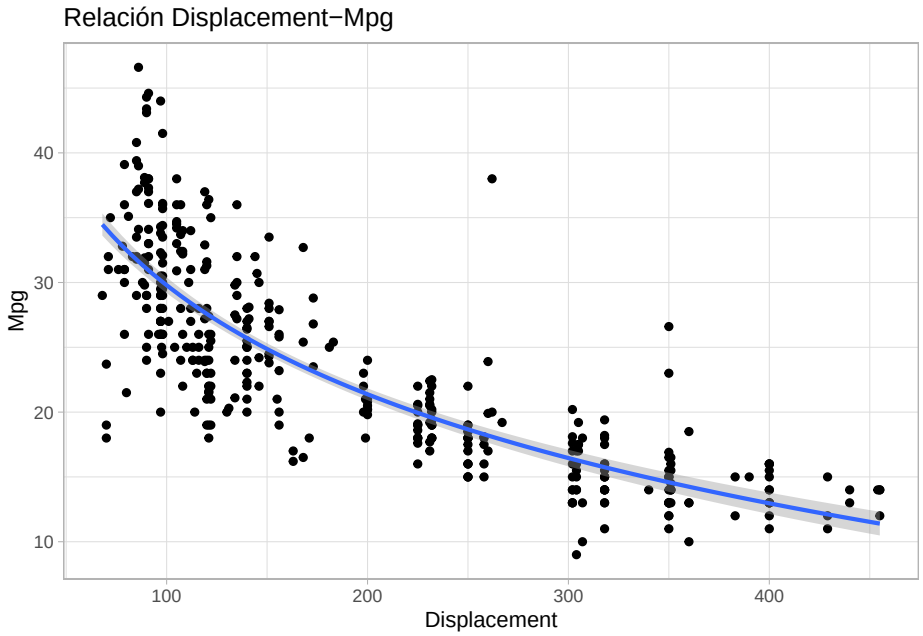


Figura 27

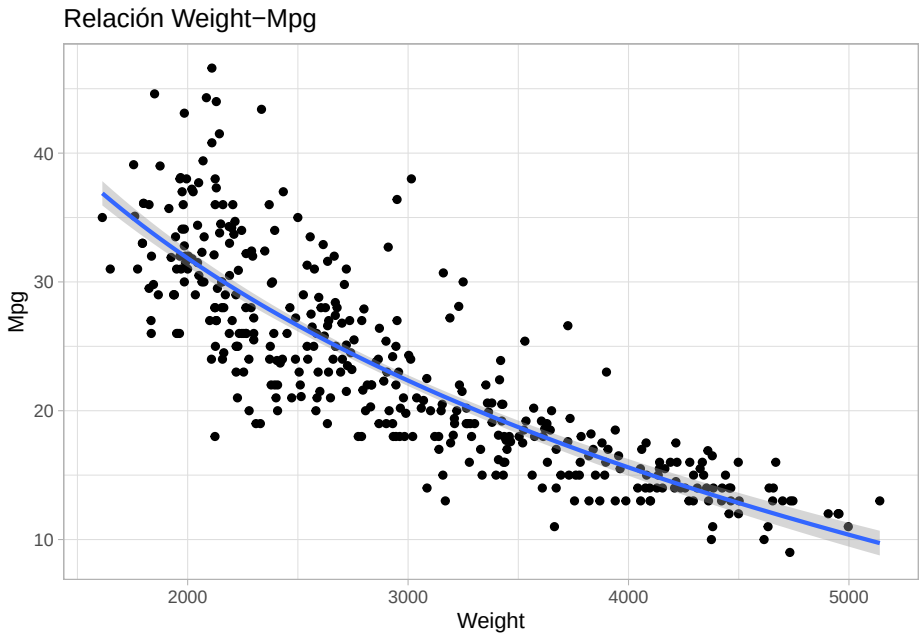


Figura 28

También podemos visualizar los regresores respecto a la salida:

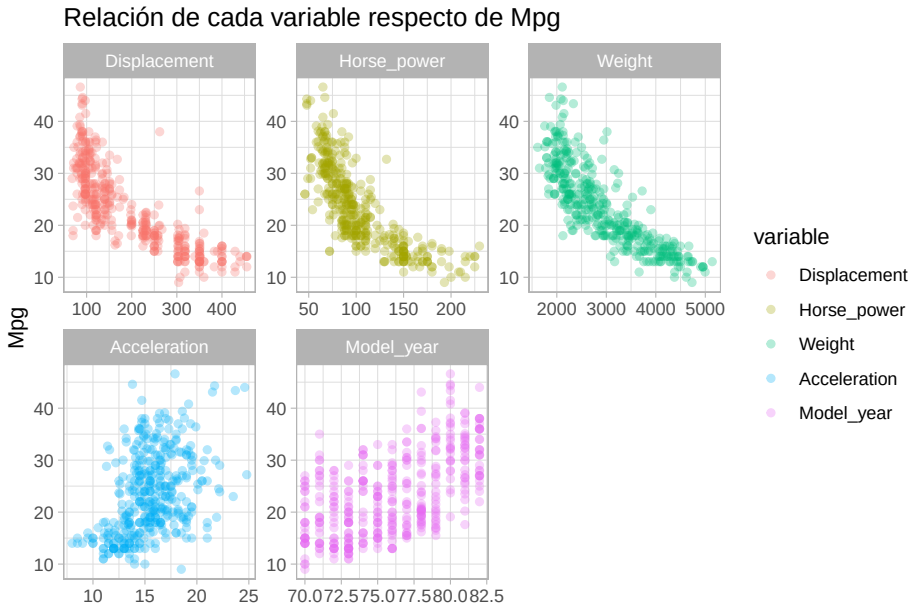


Figura 29

Como habíamos visto, existe alta correlación entre Displacement, Horse_power, Weight respecto de Mpg.

Haciendo referencia a la hipótesis H.9, Horse_power podría depender de Displacement y Weight. Parece bastante probable que la potencia de un motor dependa de la cilindrada y el peso que tenga.

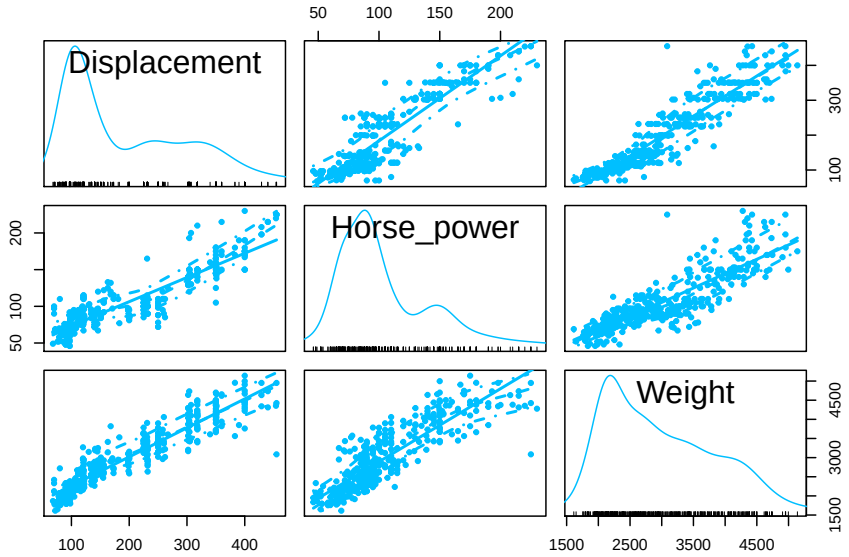


Figura 30

Viendo las funciones de densidad, buscamos ver si la media de las dos variables se asemeja con la distribución de Horse_power.

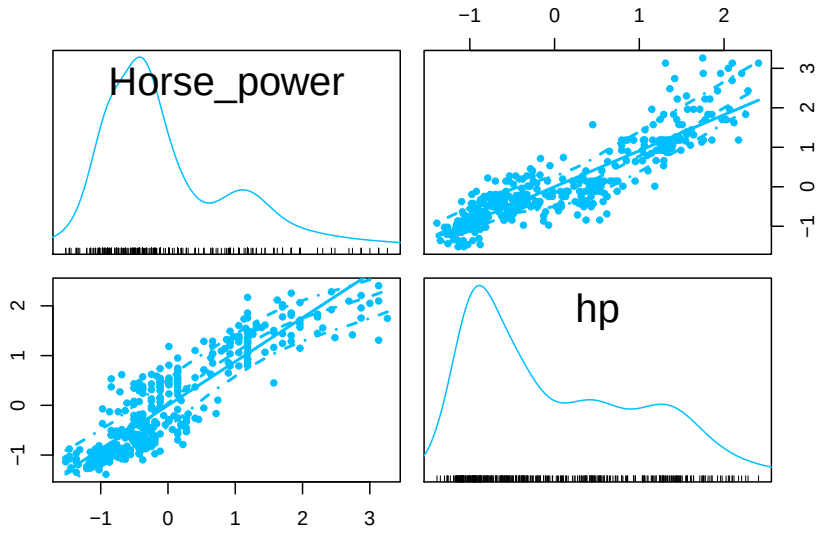


Figura 31

Viendo que no son tan similares como creíamos, buscamos diferentes fórmulas [2] para el cálculo de los caballos de vapor. Las fórmulas son un poco más complejas y no tenemos exactamente los datos necesarios para utilizarlas (no se descarta que no se puedan deducir, pero no sería un cálculo evidente).

1.2.6. Tratamiento de variables

Para este dataset, al ser casi todas las variables numéricas continuas, existen pocos tratamientos que aplicar.

Para añadir interpretabilidad, podríamos agrupar la variable Weight en intervalos, pero puesto que vamos a aplicar regresión sería más conveniente realizarlo con los resultados finales.

1.2.7. Ordenaciones

Volvemos a mostrar la cabecera de los datos:

Displacement	Horse_power	Weight	Acceleration	Model_year	Mpg
91	70	1955	20.5	71	26.0
232	100	2789	15.0	73	18.0
350	145	4055	12.0	76	13.0
318	140	4080	13.7	78	17.5
113	95	2372	15.0	70	24.0
97	60	1834	19.0	71	27.0

En este caso no es necesario aplicar ninguna reorganización. Cada variable ocupa su propia columna y contiene un único tipo de información, con unidades de observación diferentes. Tampoco existe ninguna relación entre variables sobre la información que codifican (en el sentido de que podrían agruparse).

1.2.8. Resolución de hipótesis

Nos habíamos planteado las siguientes hipótesis

- **H.1**: Horse_power puede influir en Mpg: A más potencia, más consumo.

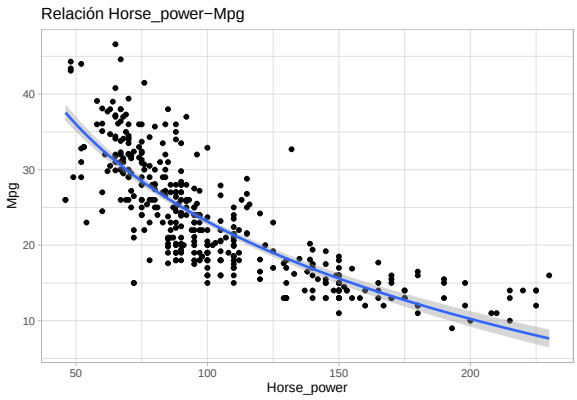


Figura 32

Con el plot y los resultados de la matriz de correlación queda claro que existe una correlación negativa entre estas dos variables. Por tanto, podemos considerar Horse_power como un buen candidato para la regresión

- **H.2**: Weight debe influir en Mpg: Un coche más pesado debería consumir más.

  Misma idea que en la hipótesis anterior, lo hemos visto anteriormente en la figura 30.

- **H.3**: Debería haber correlación entre displacement (cilindrada) con horse y acceleration.

  La hemos referenciado anteriormente.

- **H.4**: Horse y acceleration podrían estar relacionadas

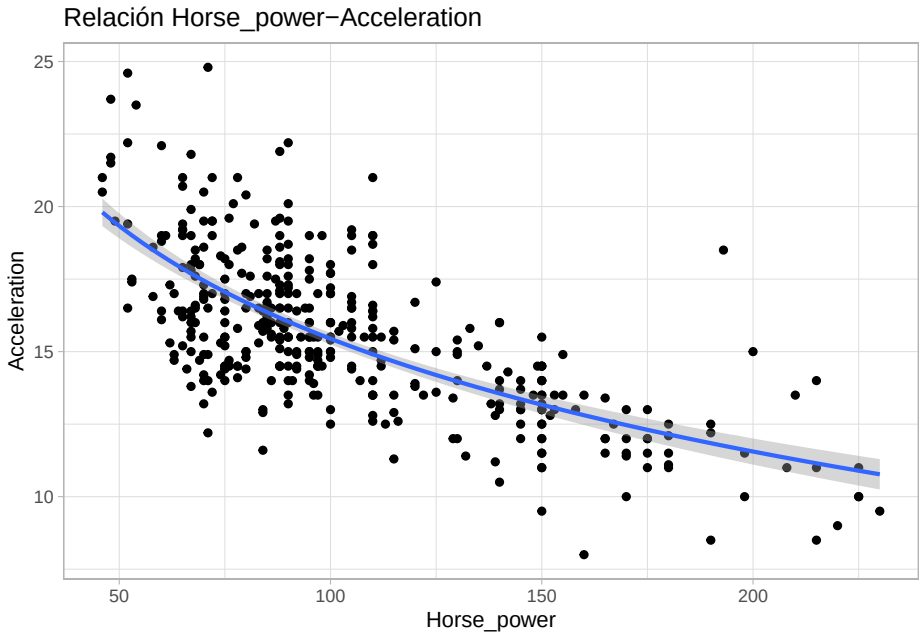


Figura 33

Se aprecia una correlación con forma logarítmica entre las dos variables.

- **H.5**: Viendo que contamos con un rango pequeño de años, no debería haber un cambio significativo de prestaciones entre años.

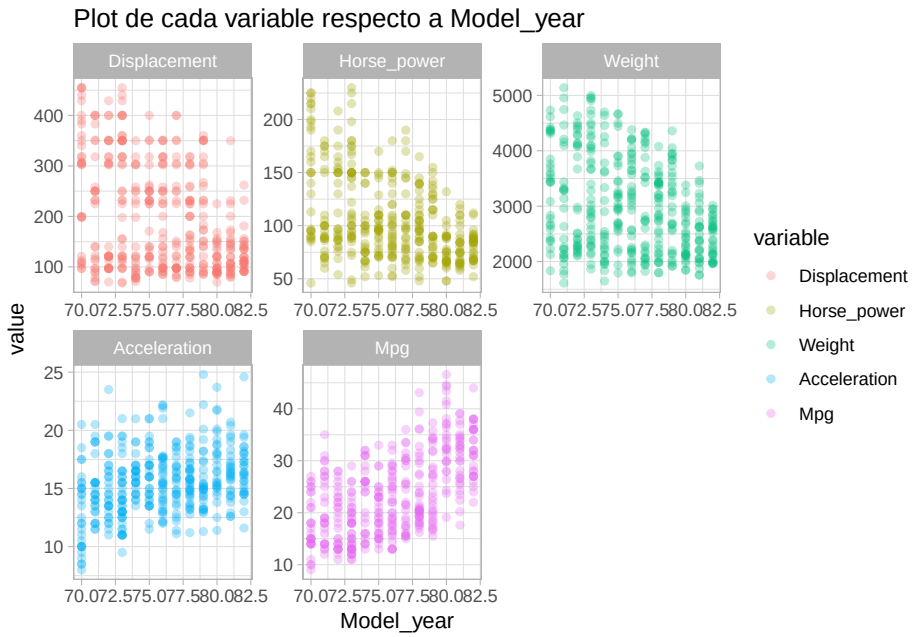


Figura 34

Existe una alta dispersión de los datos en cada una de las variables, pero aún así se aprecia tendencias en las variables. Acceleration y Mpg tienden a aumentar, y Displacement, Horse_power y Weight tienden a disminuir. También vemos que la dispersión en las prestaciones de los coches disminuyen ligeramente.

Podríamos creer en principio que puede deberse a un decremento del número de instancias con el paso de los años, pero recordamos que en general los datos están repartidos equitativamente

```
Años:   70 71 72 73 74 75 76 77 78 79 80 81 82
Conteo: 29 27 28 40 26 30 34 28 36 29 27 28 30
```

Podemos ver cómo varían los rangos para cada año

```
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
70     97           46          1835    8.0          70          9
       455          225         4732    20.5         70         27
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
71     71           60          1613    11.5         71         12
       400          180         5140    20.5         71         35
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
72     70           54          2100    11.0         72         11
       429          208         4633    23.5         72         28
```

```
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
73     68          46          1867    9.5          73         11
       455         230         4997    21.0         73         29
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
74     71          52          1649    13.5         74         13
       350         150         4699    21.0         74         32
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
75     90          53          1795    11.5         75         13
       400         170         4668    21.0         75         33
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
76     85          52          1795    12.0         76         13
       351         180         4380    22.2         76         33
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
77     79          58          1825    11.1         77         15
       400         190         4335    19.0         77         36
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
78     78          48          1800    1.2          78         16.2
       318         165         4080    21.5         78         43.1
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
79     85          65          1915    11.3         79         15.5
       360         155         4360    24.8         79         37.3
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
80     70          48          1845    11.4         80         19.1
       225         132         3381    23.7         80         46.6
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
81     79          58          1755    12.6         81         17.6
       350         120         3725    20.7         81         39.1
Year  Displacement Horse_power Weight Acceleration Model_year  Mpg
82     91          52          1965    11.6         82         22
       262         112         3015    24.6         82         44
```

- **H.6**: Pero debería existir una tendencia de mejora de prestaciones con los años, incluyendo aumento de Displacement, Horse_power y Acceleration.

Ciertamente. Se ha comprobado en la hipótesis anterior.

- **H.7**: Model_year podría no mostrar relación con Mpg: Pese al paso de los años si contamos con diferentes tipos de vehículos (todoterrenos, familiares, deportivos...) podría haber un consumo dispar. (Si existiera tendencia, viendo que los años son de las últimas décadas del siglo XX, podría ir el consumo hacia abajo)

Hemos visto que existe tendencia lineal con gran dispersión, y positiva.

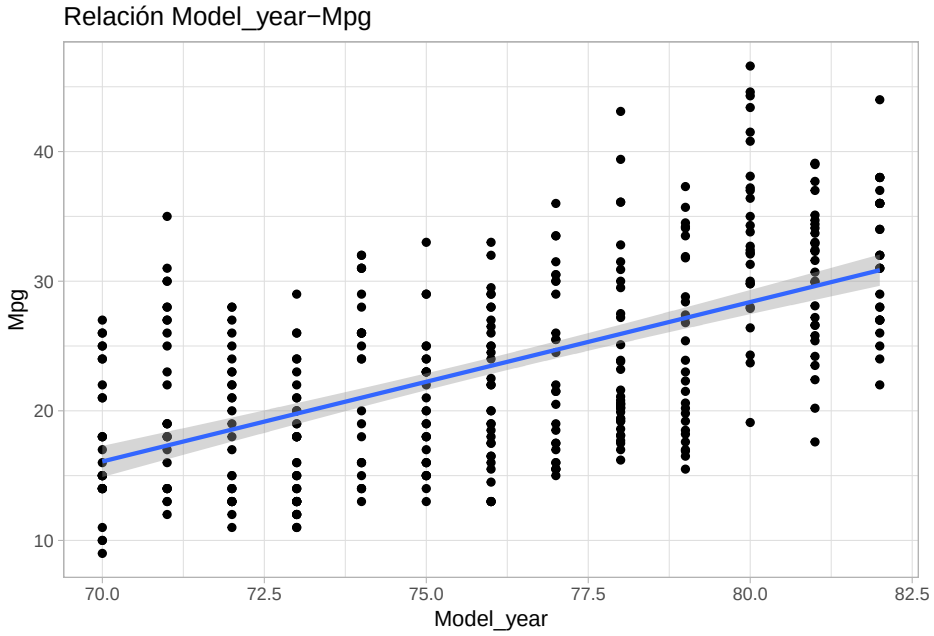


Figura 35

Por desgracia no contamos información sobre los modelos de los coches.

Podemos ver como se ubican los diferentes años en un plot Horse_power vs Mpg:

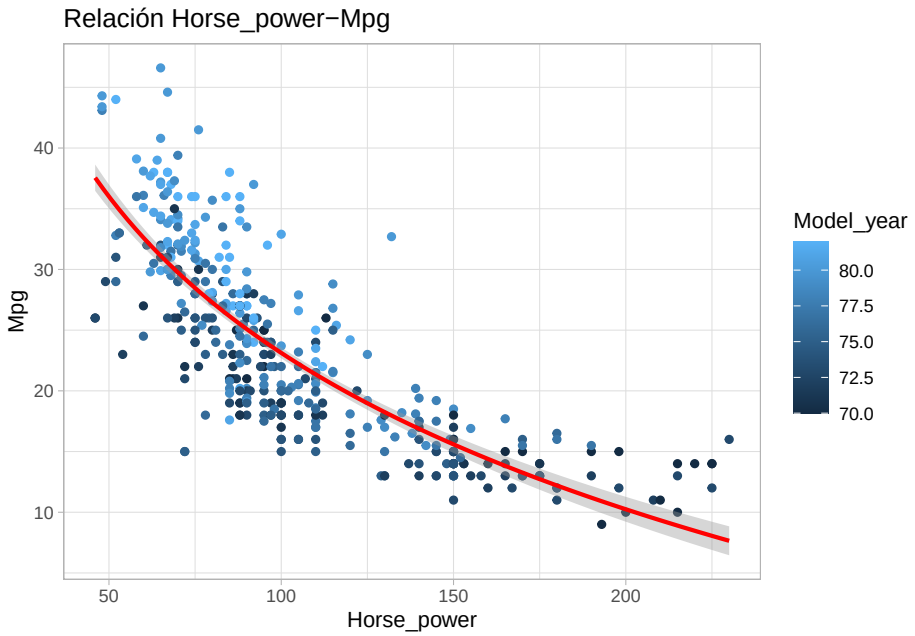


Figura 36

Y no se puede afirmar la hipótesis, los coches están entremezclados por diferentes años.

- **H.8**: Esta última hipótesis se puede aplicar al resto de variables, indicándonos que Model_year no debería tener relevancia para este problema de regresión.

No podemos afirmar la hipótesis anterior y por consiguiente esta tampoco.

- **H.9**: Horse_power podría depender de las variables Displacement y Weight

Se ha comentado anteriormente.

1.3. Conclusiones

Como conclusiones podemos decir que tenemos un dataset altamente correlacionado, distribuído de forma no normal pero con la información bien representada. Existen relaciones fuertes entre las variables de entrada y de las de salida para la regresión que probablemente nos ayuden a solucionar con facilidad el problema.

Aunque no hemos descubierto los tipos de distribución que siguen nuestras variables, por si quisiéramos transformarlas a una normal, podemos sin ninguna duda aplicar una estandarización de los datos (puesto que sabemos que no afecta negativamente al problema de regresión) siempre y cuando lo tengamos en cuenta a la hora de analizar los resultados.

Se nos pide elegir 5 regresores para la regresión y contamos exactamente con ese número, por lo que no podemos descartar ninguna variable. Aún así, hemos visto que tenemos algunas variables más interesantes que otras. Variables correladas con la salida nos aumentan las posibilidades de obtener un buen regresor, pero debemos evitar usar variables correladas entre sí para evitar la multicolinealidad, y aumentar la interpretabilidad del modelo, pero la potencia en sí de este no cambia.

2. Técnicas de Regresión

Recordamos que la descripción de los datos se encuentra en el apartado 1.1.

Como se comentó en las conclusiones del análisis estadístico:
"Se nos pide elegir 5 regresores para la regresión y contamos exactamente con ese número, por lo que no podemos descartar ninguna variable. Aún así, hemos visto que tenemos algunas variables más interesantes que otras. Variables correladas con la salida nos aumentan las posibilidades de obtener un buen regresor, pero debemos evitar usar variables correladas entre sí para evitar la multicolinealidad, y aumentar la interpretabilidad del modelo, pero la potencia en sí de este no cambia."

Primeramente, mostramos la relación de cada variable respecto a la salida:

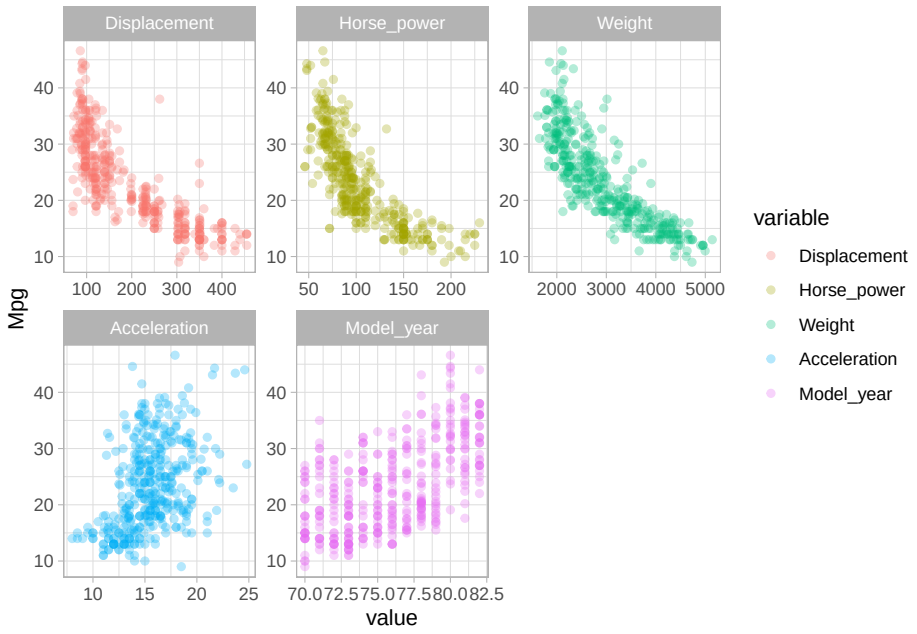


Figura 37

Como dijimos, se aprecia alta correlación entre Displacement, Horse_power, Weight respecto de la salida, probablemente de forma logarítmica.

Las matrices de correlación nos confirmaban esta idea (con coeficientes de Pearson y Kendall)

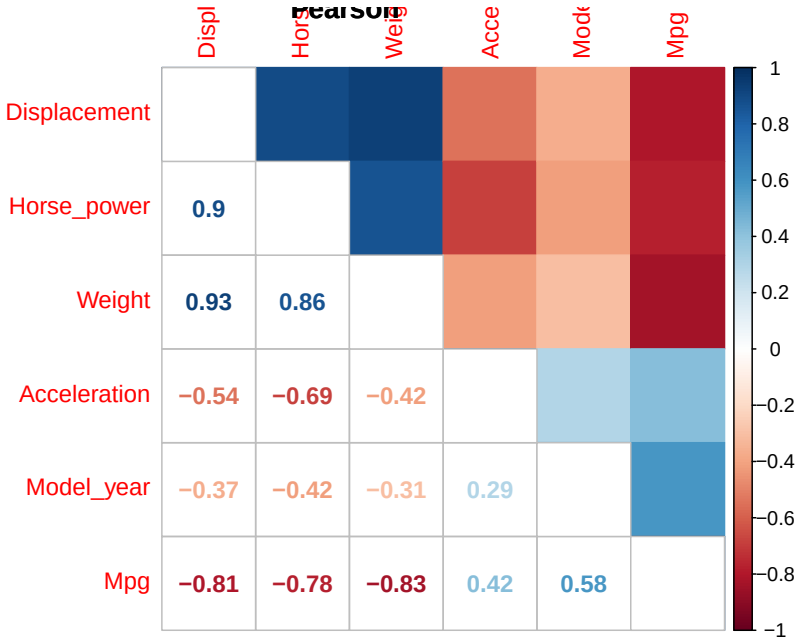


Figura 38

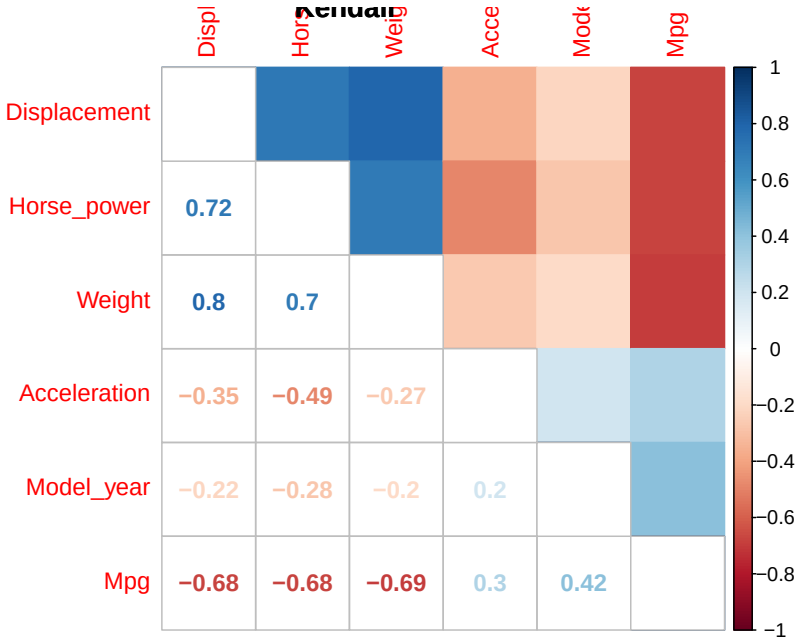


Figura 39

Por tanto, si las ordenáramos por cuáles parecen ser más prometedoras, tendríamos: Weight > Displacement > Horse_power > Model_year > Acceleration

También tenemos que tener en cuenta que las tres primeras variables están correladas entre sí.

2.1. Ajustes de regresión lineal univariables

Vamos a analizar un ajuste con cada una de las características:

```
Call: lm(formula = Mpg ~ Weight, data = auto)

Residuals:
     Min       1Q   Median       3Q      Max
-11.9736  -2.7556  -0.3358   2.1379  16.5194

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 46.216524   0.798673   57.87   <2e-16 ***
Weight      -0.007647   0.000258  -29.64   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.333 on 390 degrees of freedom
Multiple R-squared:  0.6926,    Adjusted R-squared:  0.6918
F-statistic: 878.8 on 1 and 390 DF,  p-value: < 2.2e-16

"-----------------------------------------"

Call: lm(formula = Mpg ~ Displacement, data = auto)

Residuals:
     Min       1Q   Median       3Q      Max
-12.9170  -3.0243  -0.5021   2.3512  18.6128

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  35.12064    0.49443   71.03   <2e-16 ***
Displacement -0.06005    0.00224  -26.81   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.635 on 390 degrees of freedom
Multiple R-squared:  0.6482,    Adjusted R-squared:  0.6473
F-statistic: 718.7 on 1 and 390 DF,  p-value: < 2.2e-16

"-----------------------------------------"

Call: lm(formula = Mpg ~ Horse_power, data = auto)

Residuals:
     Min       1Q   Median       3Q      Max
-13.5710  -3.2592  -0.3435   2.7630  16.9240
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 39.935861   0.717499   55.66   <2e-16 ***
Horse_power -0.157845   0.006446  -24.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.906 on 390 degrees of freedom
Multiple R-squared:  0.6059,    Adjusted R-squared:  0.6049
F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16

"-----------------------------------------"

Call: lm(formula = Mpg ~ Model_year, data = auto)

Residuals:
     Min       1Q   Median       3Q      Max
-12.0212  -5.4411  -0.4412   4.9739  18.2088

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -70.01167    6.64516  -10.54   <2e-16 ***
Model_year    1.23004    0.08736   14.08   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.363 on 390 degrees of freedom
Multiple R-squared:  0.337, Adjusted R-squared:  0.3353
F-statistic: 198.3 on 1 and 390 DF,  p-value: < 2.2e-16

"-----------------------------------------"

Call: lm(formula = Mpg ~ Acceleration, data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-17.989  -5.616  -1.199   4.801  23.239

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.8332     2.0485   2.359   0.0188 *
Acceleration   1.1976     0.1298   9.228   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.08 on 390 degrees of freedom
Multiple R-squared:  0.1792,    Adjusted R-squared:  0.1771
F-statistic: 85.15 on 1 and 390 DF,  p-value: < 2.2e-16
```

Al ser univariable, por ahora no es necesario fijarse en el estadístico F. Para ver el potencial de la variable, debemos darle importancia al p-valor (comprobar de que sea lo suficientemente bajo), y posteriormente ver el R^2 para everiguar el porcentaje de la salida explicada.

En base a los resultados vemos que el test de correlación nos había ayudado correctamente: de forma individual todas las variables tienen dependencia lineal, y el orden de calidad coincide con el orden de fuerza en las correlaciones.

Ya con el uso de la variable Weight vemos que podemos explicar un ~69 % de la salida, un buen valor de partida. El ajuste quedaría de esta manera:

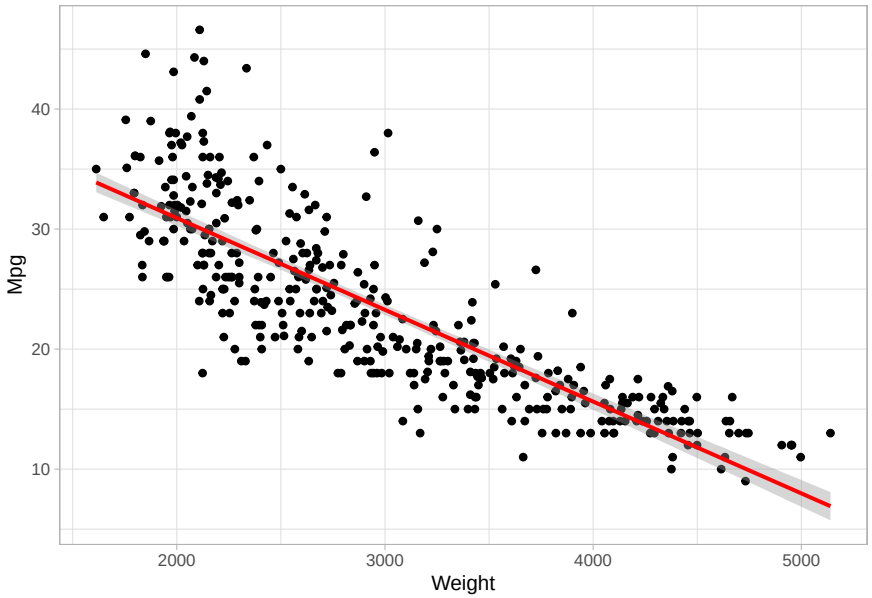


Figura 40

Con los coeficientes:

```
                  2.5 %      97.5 %
(Intercept) 44.646282308 47.78676679
Weight      -0.008154515 -0.00714017
```

Aunque los valores del intervalo del coeficiente de Weight sea bajo, vemos que no incluye el cero (y con el p-valor obtenido anteriormente, lo podemos asegurar con bastante certeza). Probablemente la razón de estos coeficientes tan pequeños es que los datos no están estandarizados (se podría hacer perfectamente, se han dejado con sus rangos normales para interpretarlos mejor) y los valores de las unidades de medida son bastante diferentes (hablamos de rangos de [9.0,46.6] en Mpg frente a [1613,5140] en Weight)

Ya con esto podemos intentan interpretar un poco los datos, tendríamos por ahora la fórmula de regresión lineal:

$$Mpg \sim Weight \tag{1}$$

2.2. Ajustes de regresión lineal multivariable

Aplicamos un método descendente:

```
Call: lm(formula = Mpg ~ ., data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-8.5211 -2.3920 -0.1036  2.0312 14.2874

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.544e+01  4.677e+00  -3.300  0.00106 **
Displacement  2.782e-03  5.462e-03   0.509  0.61082
Horse_power   1.020e-03  1.376e-02   0.074  0.94095
Weight       -6.874e-03  6.653e-04 -10.333  < 2e-16 ***
Acceleration  9.032e-02  1.019e-01   0.886  0.37599
Model_year    7.541e-01  5.261e-02  14.334  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.435 on 386 degrees of freedom
Multiple R-squared:  0.8088,    Adjusted R-squared:  0.8063
F-statistic: 326.5 on 5 and 386 DF,  p-value: < 2.2e-16
```

El p-valor del F estadístico nos dice que al menos hay una variable (realmente ya lo sabíamos de los ajustes univariables) con dependencia linea.

Vemos que hay 3 variables con mal p-valor, empezamos quitando la que lo tiene más alto, Horse_power.

```
Call: lm(formula = Mpg ~ . - Horse_power, data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-8.5182 -2.3948 -0.1085  2.0405 14.2908

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.527e+01  4.106e+00  -3.719 0.000229 ***
Displacement  2.874e-03  5.310e-03   0.541 0.588651
Weight       -6.852e-03  5.967e-04 -11.483  < 2e-16 ***
Acceleration  8.555e-02  7.885e-02   1.085 0.278595
Model_year    7.532e-01  5.118e-02  14.717  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.431 on 387 degrees of freedom
Multiple R-squared:  0.8088,    Adjusted R-squared:  0.8068
```

```
F-statistic: 409.2 on 4 and 387 DF,  p-value: < 2.2e-16
```

El F estadístico está correcto, y seguimos teniendo variables con p-valor grande, quitamos Displacement.

```
Call: lm(formula = Mpg ~ . - Horse_power - Displacement, data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-8.6749 -2.3528 -0.1082  2.0168 14.3022

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -14.936555   4.055512  -3.683 0.000263 ***
Weight        -0.006554   0.000230 -28.502  < 2e-16 ***
Acceleration   0.066359   0.070361   0.943 0.346204
Model_year     0.748446   0.050366  14.860  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.428 on 388 degrees of freedom
Multiple R-squared:  0.8086,    Adjusted R-squared:  0.8071
F-statistic: 546.5 on 3 and 388 DF,  p-value: < 2.2e-16
```

idem. a lo anterior, quitamos Acceleration.

```
Call: lm(formula = Mpg ~ . - Horse_power - Displacement - Acceleration,
    data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-8.8505 -2.3014 -0.1167  2.0367 14.3555

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.435e+01  4.007e+00  -3.581 0.000386 ***
Weight      -6.632e-03  2.146e-04 -30.911  < 2e-16 ***
Model_year   7.573e-01  4.947e-02  15.308  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.427 on 389 degrees of freedom
Multiple R-squared:  0.8082,    Adjusted R-squared:  0.8072
F-statistic: 819.5 on 2 and 389 DF,  p-value: < 2.2e-16
```

El estadístico F sigue bien, y los p-valores de las variables son extremadamente bajos. Nos fijamos en el R^2 y vemos que ha subido considerablemente (un 10 %) respecto al univariable, por lo que este sería nuestro modelo aditivo por ahora.

A partir de ahora deberíamos tener cuidado si el R^2 sigue aumentando, hay que evitar el overfitting en el modelo.

2.3. Inserción de interacciones

Del modelo aditivo solo nos han quedado dos regresores, así que probamos a incluirlos como interacción.

```
Call: lm(formula = Mpg ~ +Weight * Model_year, data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-8.0397 -1.9956 -0.0983  1.6525 12.9896

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       -1.105e+02  1.295e+01  -8.531 3.30e-16 ***
Weight             2.755e-02  4.413e-03   6.242 1.14e-09 ***
Model_year         2.040e+00  1.718e-01  11.876  < 2e-16 ***
Weight:Model_year -4.579e-04  5.907e-05  -7.752 8.02e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.193 on 388 degrees of freedom
Multiple R-squared:  0.8339,    Adjusted R-squared:  0.8326
F-statistic: 649.3 on 3 and 388 DF,  p-value: < 2.2e-16
```

El F estadístico sigue bien y los p-valores son bajos, el nuevo R^2 ha mejorado un 3 %, así que no es demasiado para considerar un overfitting. Probablemente más de un 90 % sería preocupante, pero también tenemos que tener en cuenta que las variables están fuertemente correladas con la salida.

Podríamos probar a añadir alguna interacción más con alguna variable que no hubiera entrado en el modelo aditivo, pero no se espera que mejore:

```
Call: lm(formula = Mpg ~ +Weight * Model_year + Acceleration * Displacement,
    data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-7.3130 -1.8670 -0.0426  1.6109 12.2499

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              -1.131e+02  1.321e+01  -8.564 2.65e-16 ***
Weight                    2.456e-02  4.693e-03   5.234 2.73e-07 ***
Model_year                1.907e+00  1.769e-01  10.778  < 2e-16 ***
Acceleration              7.273e-01  1.282e-01   5.671 2.79e-08 ***
```

```
Displacement               3.605e-02  8.673e-03   4.157 3.98e-05 ***
Weight:Model_year         -4.054e-04  6.281e-05  -6.454 3.29e-10 ***
Acceleration:Displacement -2.953e-03  6.219e-04  -4.748 2.91e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.075 on 385 degrees of freedom
Multiple R-squared:  0.8472,    Adjusted R-squared:  0.8448
F-statistic: 355.7 on 6 and 385 DF,  p-value: < 2.2e-16
```

A pesar de nuestra suposición los p-valores son válidos y el R^2 aumenta un 1 %. Es cuestionable si el aumento de la complejidad del modelo merece con este incremento de R^2. Por simplificar vamos a quedarnos con el modelo aditivo anterior y probar con otra interacción.

Podemos probar combinando la variable Acceleration separadamente con las que ya teníamos (Weight y Model_year).

```
Call: lm(formula = Mpg ~ +Weight * Model_year + Acceleration * Weight,
    data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-7.4473 -1.7994 -0.0496  1.4790 12.1258

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         -1.230e+02  1.298e+01  -9.480  < 2e-16 ***
Weight               2.971e-02  4.419e-03   6.722 6.47e-11 ***
Model_year           1.926e+00  1.742e-01  11.055  < 2e-16 ***
Acceleration         1.341e+00  2.323e-01   5.772 1.61e-08 ***
Weight:Model_year   -4.078e-04  6.197e-05  -6.581 1.53e-10 ***
Weight:Acceleration -3.808e-04  7.537e-05  -5.052 6.76e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.061 on 386 degrees of freedom
Multiple R-squared:  0.8482,    Adjusted R-squared:  0.8462
F-statistic: 431.4 on 5 and 386 DF,  p-value: < 2.2e-16
```

```
Call: lm(formula = Mpg ~ +Weight * Model_year + Acceleration * Model_year,
    data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-7.8674 -1.9539 -0.0617  1.7397 12.3964
```

```
Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)              -4.046e+01  2.833e+01  -1.428  0.15401
Weight                    2.523e-02  4.881e-03   5.170 3.76e-07 ***
Model_year                1.072e+00  3.704e-01   2.895  0.00401 **
Acceleration             -3.956e+00  1.268e+00  -3.120  0.00195 **
Weight:Model_year        -4.263e-04  6.475e-05  -6.584 1.49e-10 ***
Model_year:Acceleration   5.476e-02  1.663e-02   3.293  0.00108 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.117 on 386 degrees of freedom
Multiple R-squared:  0.8426,    Adjusted R-squared:  0.8406
F-statistic: 413.2 on 5 and 386 DF,  p-value: < 2.2e-16
```

Y entre los dos nos podríamos quedar con el primero por tener mejores p-valores y un mejor R^2. Aun así, el incremento es pequeño respecto a nuestro modelo aditivo.

La fórmula del modelo aditivo que llevamos por ahora es:

$$Mpg \sim Weight + Model_year + Acceleration + Weight * Model_year + Weight * Acceleration \quad (2)$$

Gráficamente:

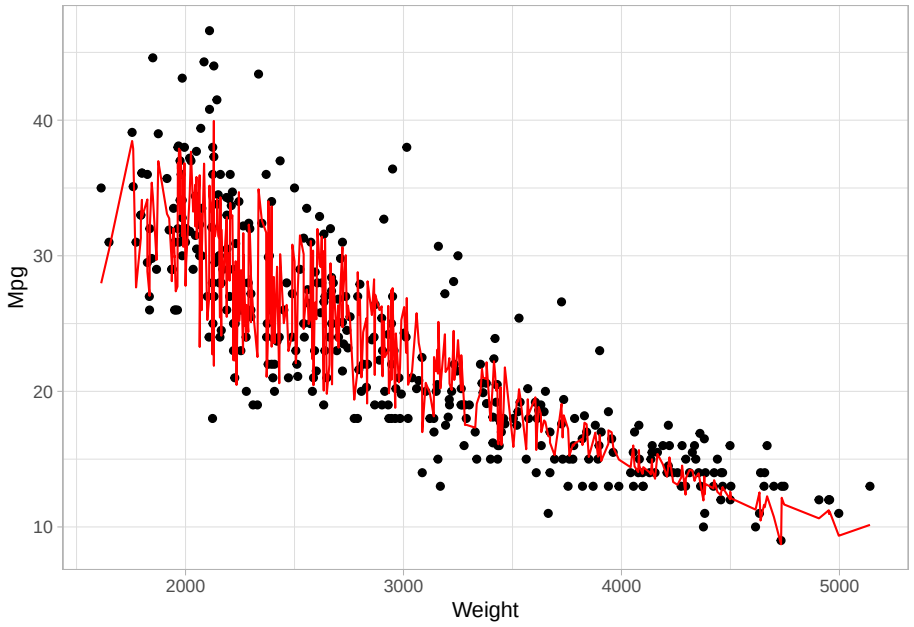


Figura 41

Por el uso multivariable la línea de regresión que se forma podría indicarnos un posible overfitting en el modelo, vamos a dejarlo por ahora e intentar solucionarlo con el modelo no lineal.

2.4. Ajustes de regresión no lineal

Habíamos dicho que las gráficas nos mostraban una tendencia logarítmica, vamos a incluír la de Weight en nuestro modelo aditivo:

```
Call: lm(formula = Mpg ~ +Weight * Model_year + Acceleration * Weight +
    I(log(Weight)), data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-7.6734 -1.7933 -0.0576  1.3154 12.1716

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.191e+02  4.120e+01   2.891  0.00406 **
Weight               2.842e-02  4.227e-03   6.723 6.44e-11 ***
Model_year           1.638e+00  1.728e-01   9.480  < 2e-16 ***
Acceleration         7.236e-01  2.435e-01   2.972  0.00315 **
I(log(Weight))      -3.028e+01  4.914e+00  -6.162 1.81e-09 ***
Weight:Model_year   -2.971e-04  6.186e-05  -4.803 2.24e-06 ***
Weight:Acceleration -1.775e-04  7.919e-05  -2.241  0.02559 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.924 on 385 degrees of freedom
Multiple R-squared:  0.8618,    Adjusted R-squared:  0.8597
F-statistic: 400.2 on 6 and 385 DF,  p-value: < 2.2e-16
```

El estadístico F está bien y los p-valores también, aunque el de la interacción Weight-Acceleration es alto comparado con el resto (aún así sigue siendo aceptable).

Como el R^2 ha subido, por ver si mejora, vamos a quitar esta interacción.

```
Call: lm(formula = Mpg ~ +Weight * Model_year + I(log(Weight)), data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-8.7501 -1.7470 -0.0725  1.3122 12.6776

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         1.715e+02  3.810e+01   4.501 8.98e-06 ***
Weight              2.522e-02  4.119e-03   6.123 2.25e-09 ***
Model_year          1.572e+00  1.708e-01   9.202  < 2e-16 ***
I(log(Weight))     -3.540e+01  4.538e+00  -7.800 5.82e-14 ***
Weight:Model_year  -2.701e-04  6.003e-05  -4.499 9.04e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.972 on 387 degrees of freedom
```

```
Multiple R-squared:  0.8565,    Adjusted R-squared:  0.855
F-statistic: 577.3 on 4 and 387 DF,  p-value: < 2.2e-16
```

Hemos empeorado un 0.5 %, bastante poco, y el modelo es más simple. La dejamos quitada.

Mostramos este ajuste:

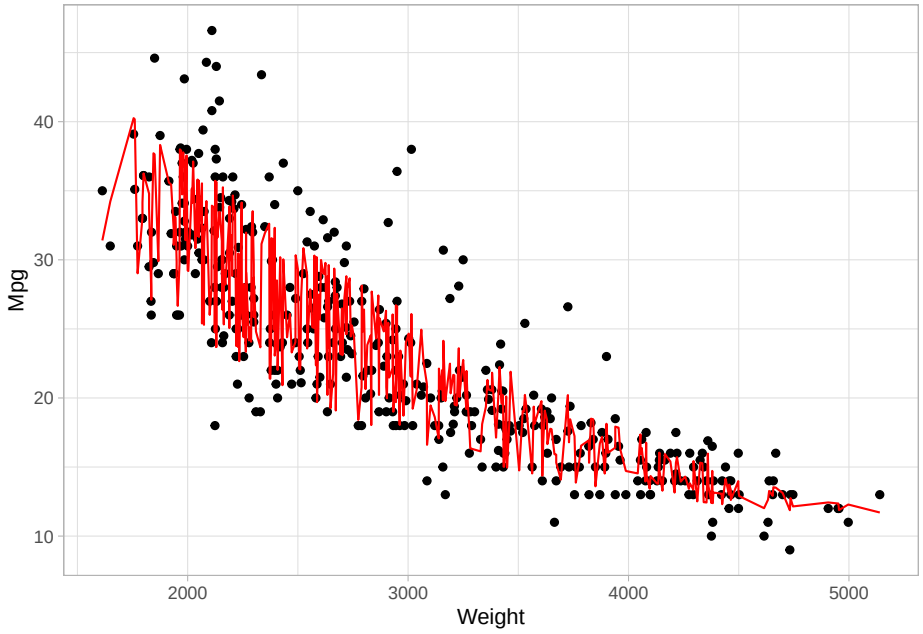


Figura 42

Esta gráfica nos indica que es probable que se esté generando sobreajuste, se ve necesario simplificar el modelo.

Si quitamos la otra interacción:

```
Call: lm(formula = Mpg ~ Weight + Model_year + I(log(Weight)), data = auto)

Residuals:
    Min      1Q  Median      3Q     Max
-9.3384 -1.7476 -0.2122  1.5322 13.2812

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    284.287315  29.392946   9.672  < 2e-16 ***
Weight           0.007772   0.001420   5.473 7.97e-08 ***
Model_year       0.828693   0.044506  18.620  < 2e-16 ***
I(log(Weight)) -43.590633   4.258803 -10.235  < 2e-16 ***
---
```

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

Residual standard error: 3.045 on 388 degrees of freedom
Multiple R-squared:  0.849, Adjusted R-squared:  0.8478
F-statistic:   727 on 3 and 388 DF,  p-value: < 2.2e-16
```

No hemos perdido apenas R^2. Mostramos la gráfica:

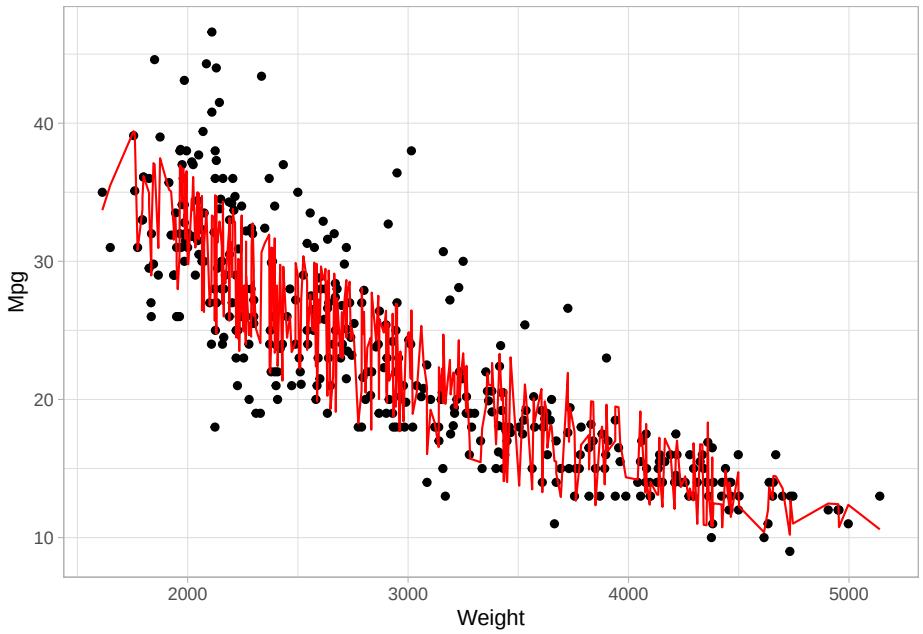


Figura 43

Seguimos con el mismo problema, probablemente se deba a una de las variables. Quitamos Model_year por tener poca correlación con la variable de salida:

```
Call: lm(formula = Mpg ~ Weight + I(log(Weight)), data = auto)

Residuals:
     Min       1Q   Median       3Q      Max
-12.5329  -2.7031  -0.4016   1.7038  16.0835

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    263.812407  40.366256   6.535 1.99e-10 ***
Weight           0.002582   0.001914   1.349    0.178
I(log(Weight)) -31.166013   5.780558  -5.392 1.21e-07 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 4.185 on 389 degrees of freedom
Multiple R-squared:  0.714, Adjusted R-squared:  0.7125
```

```
F-statistic: 485.6 on 2 and 389 DF,  p-value: < 2.2e-16
```

El p-valor de Weight nos indica que hay que quitarla, y al no estar incluída ninguna interacción, no es un término de jerarquía, por lo que podemos hacerlo. Se puede porque la variable sigue siendo independiente, solamente no está modelada de forma lineal, sino logarítmicamente.

```
Call: lm(formula = Mpg ~ I(log(Weight)), data = auto)

Residuals:
     Min       1Q   Median       3Q      Max
-12.4315  -2.6752  -0.2888   1.9429  16.0136

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    209.9433     6.0002   34.99   <2e-16 ***
I(log(Weight)) -23.4317     0.7534  -31.10   <2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 4.189 on 390 degrees of freedom
Multiple R-squared:  0.7127,    Adjusted R-squared:  0.7119
F-statistic: 967.3 on 1 and 390 DF,  p-value: < 2.2e-16
```

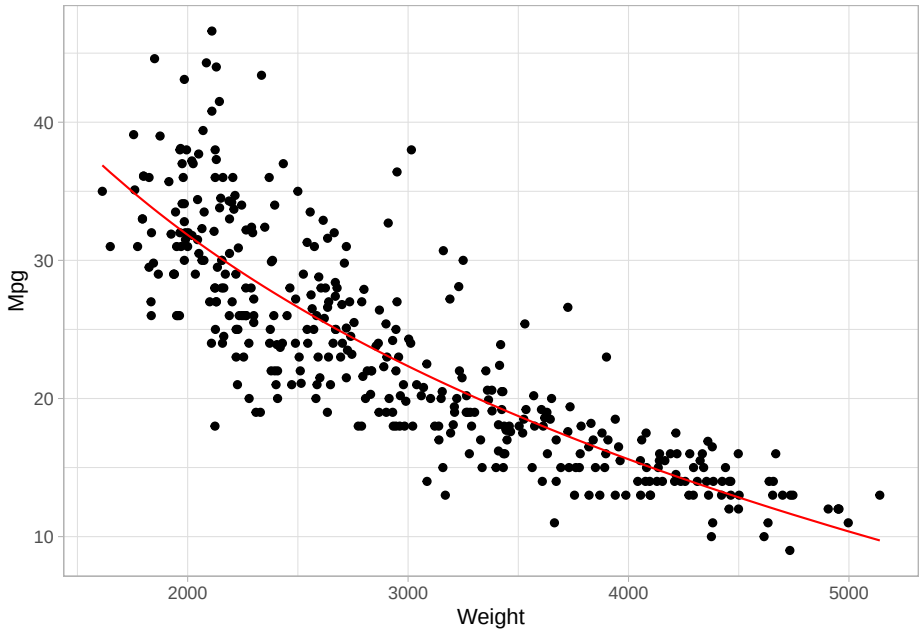


Figura 44

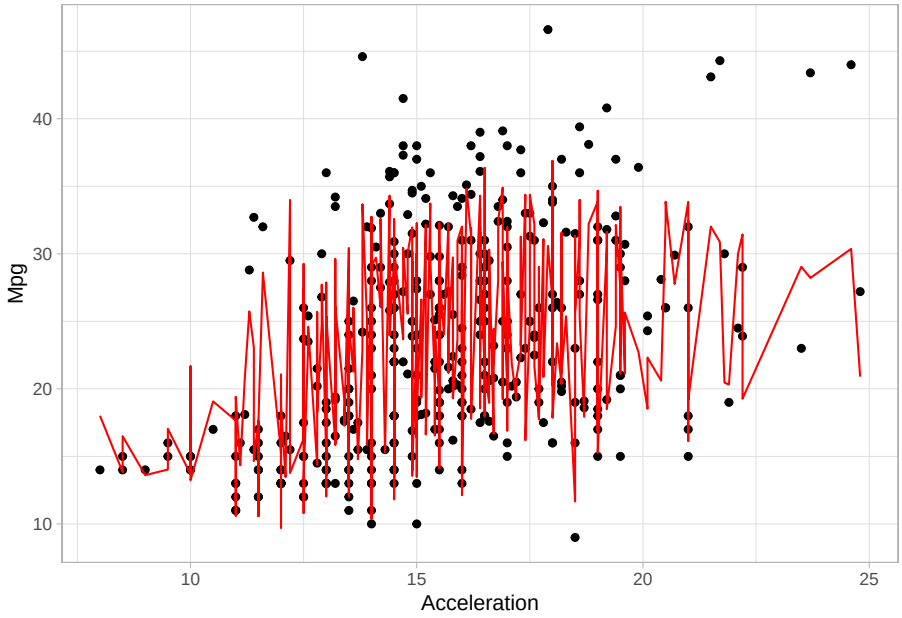


Figura 45

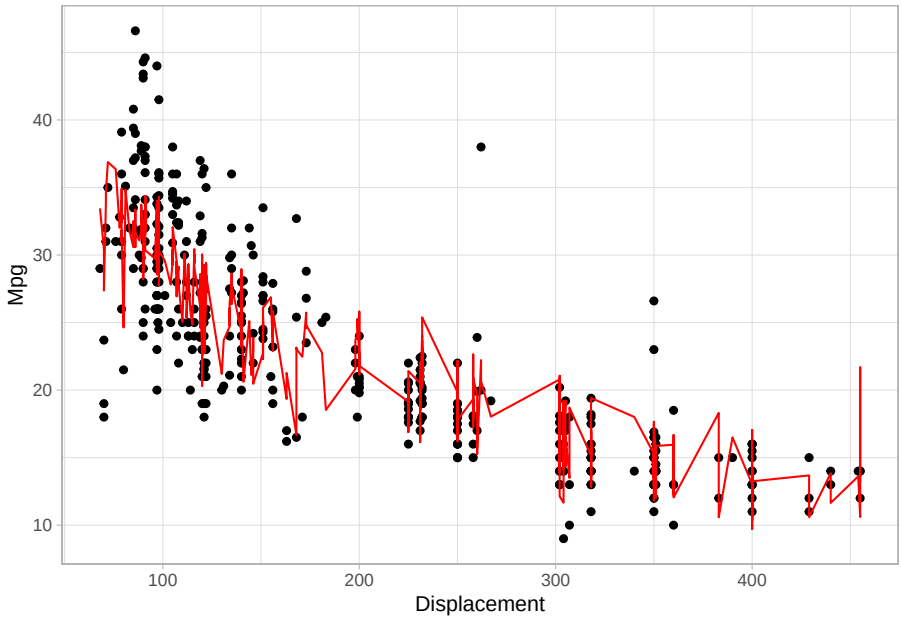


Figura 46

Vemos un empeoramiento significativo en la calidad de R^2 respecto al modelo multivariable, pero la forma del modelo no está tan ajustada a los datos y parece sensato mantenerlo así.

Aún así, no resulta lógico intentar predecir el Mpg de un coche únicamente en base al peso, alguna de las otras variables deberían ayudarnos en la predicción. Por ejemplo, alguna característica del motor, como la cilindrada o los caballos de vapor.

Para resumir, mostramos el modelo con mejor R^2 tras hacer múltiples pruebas, e intentando evitar un overfitting:

```
Call: lm(formula = Mpg ~ Acceleration + I(log(Weight)) + I(log(Displacement)),
    data = auto)

Residuals:
     Min       1Q   Median       3Q      Max
-12.9074  -2.6174  -0.4104   1.9500  16.5596

Coefficients:          Estimate Std. Error t value Pr(>|t|)
(Intercept)          171.61778   12.14751  14.128  < 2e-16 ***
Acceleration           0.19717    0.08914   2.212   0.0276 *
I(log(Weight))       -16.94003    2.27727  -7.439 6.59e-13 ***
I(log(Displacement))  -3.19963    1.26881  -2.522   0.0121 *
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 4.104 on 388 degrees of freedom
Multiple R-squared:  0.7256,    Adjusted R-squared:  0.7235
F-statistic: 342.1 on 3 and 388 DF,  p-value: < 2.2e-16
```

Los p-valores no son muy fuertes, pero siguen siendo aceptables, y gráficamente el modelo se ve un poco mejor:

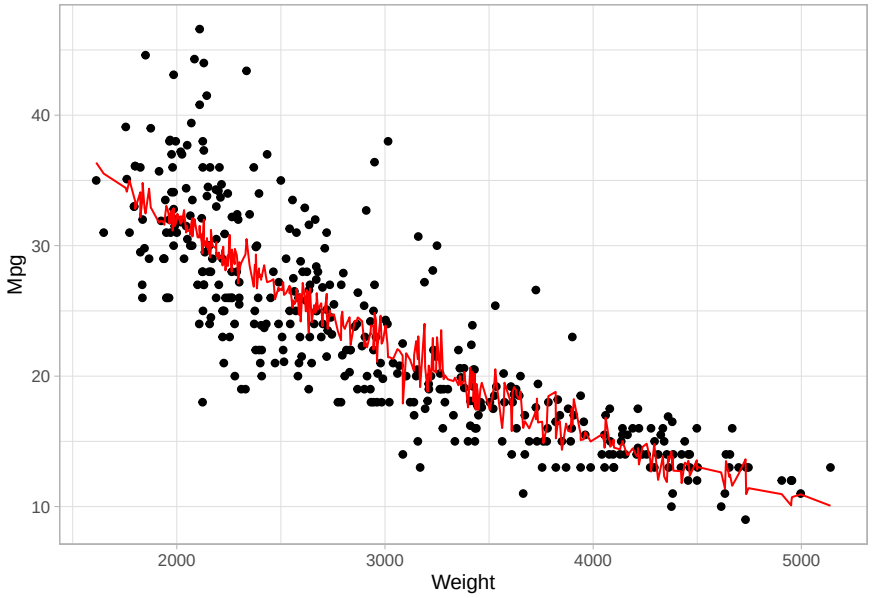


Figura 47

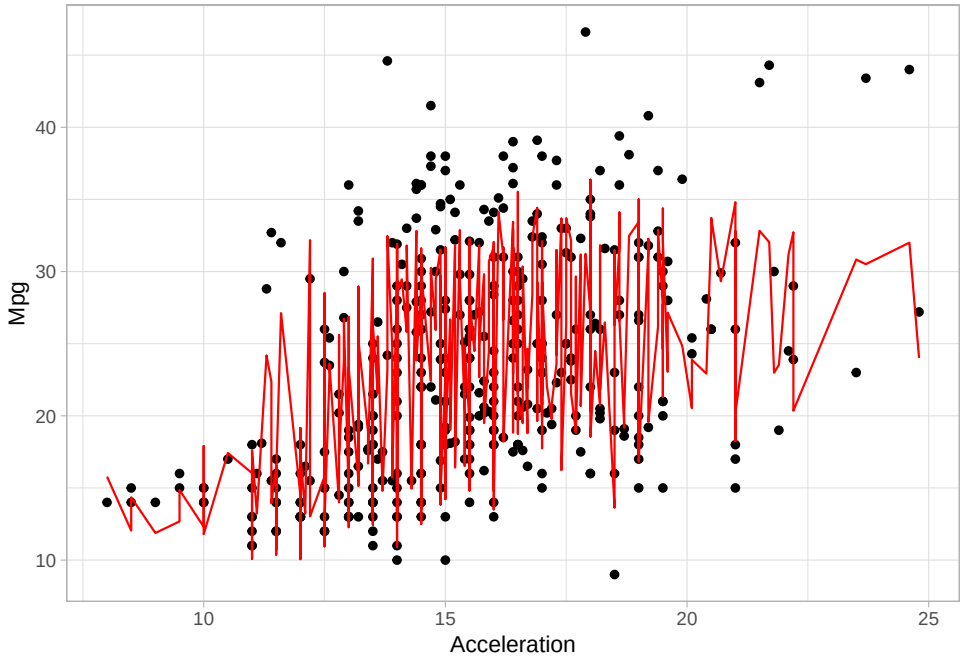


Figura 48

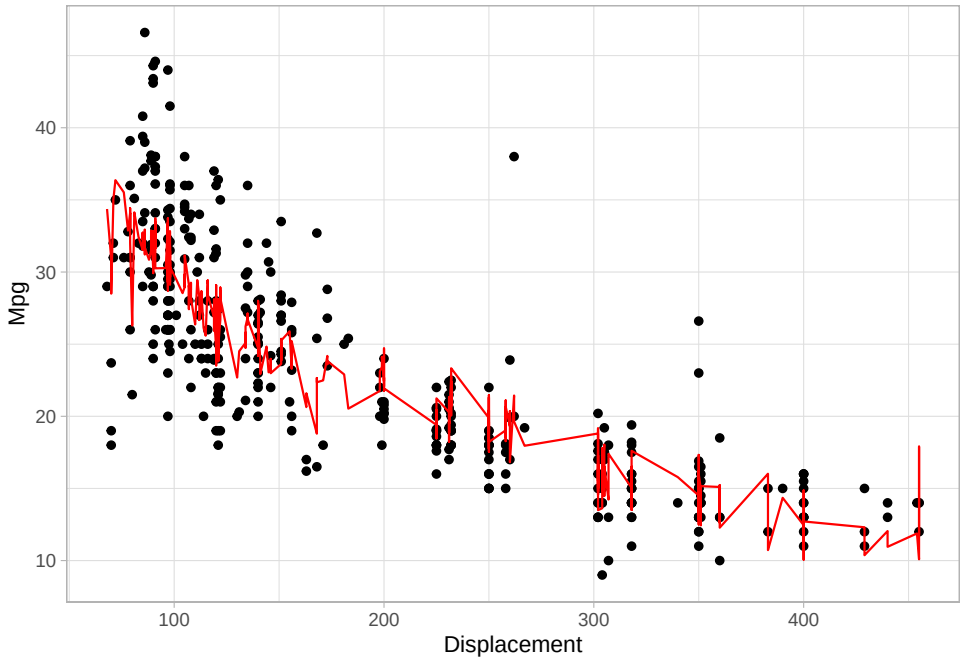


Figura 49

Pensando en el problema, y tras el análisis hecho en el apartado de EDA, creemos que usar Model_year para predecir Mpg no parece buena idea. La gráfica de la variable nos muestra mucha dispersión en los datos y, aunque sí se ve un cierta tendencia lineal, no parece suficiente para usarla. Claramente nos ajusta mejor los datos pero parece que nos estamos pegando a ellos.

De cara a comprobar este razonamiento en el cross-validation, vamos a guardar dos modelos:

Modelo con mejor R^2

$$Mpg \sim Weight + Model\_year + I(log(Weight) \tag{3}$$

Modelo intentando evitar el overfitting

$$Mpg \sim Acceleration + I(log(Weight)) + I(log(Displacement)) \tag{4}$$

2.5. Ajustes con KNN

Sabemos que la función por defecto usa la distancia de Minkowski y escala los datos a igual rango. También usa un K de 7, pero sería recomendable probar con varios.

Vamos a probar con diferentes modelos, primero el multivariable con todas

```
Mpg ~ .
    1.880835
```

Y probando con varios obtenemos el menor error eliminando únicamente Acceleration:

```
Mpg ~ . - Acceleration
    1.856269
```

Que visualmente nos quedaría:

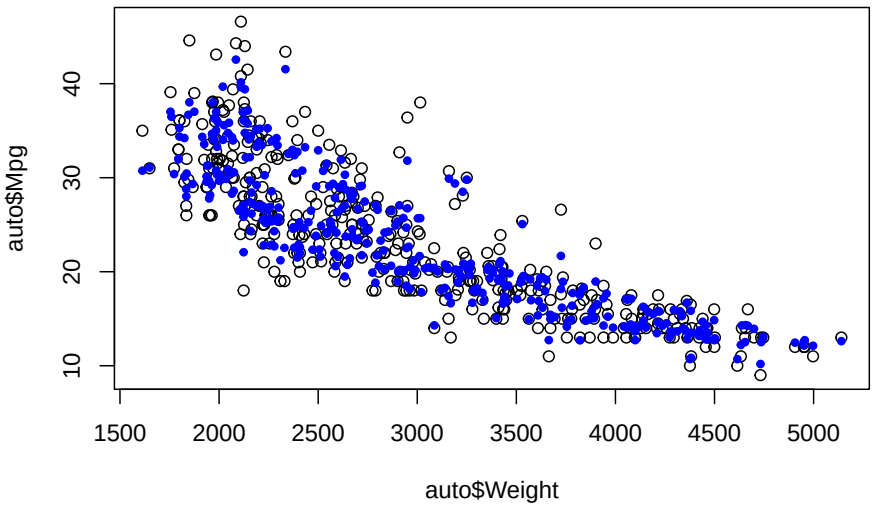


Figura 50

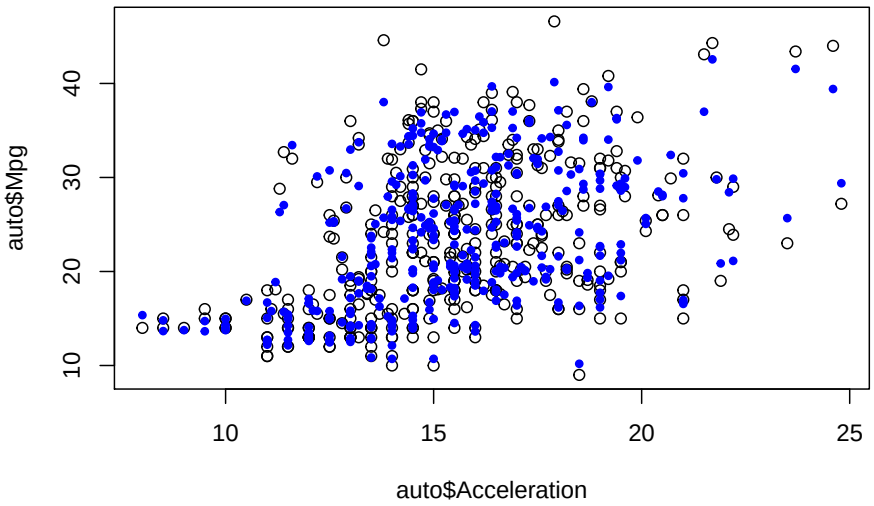


Figura 51

Si probamos el modelo no lineal obtenido en los pasos anteriores (con mejor R^2) nos un error bastante malo.

```
Mpg ~ Weight + Model_year + I(log(Weight))
    2.104086
```

El método para evitar el overfitting que usamos en el apartado anterior probablemente no funcione con KNN por seguir una metodología totalmente diferente. El ajuste de KNN para regresión no tiene nada que ver con los modelos LM. Podemos aún así comprobarlo:

```
Mpg ~ Acceleration + I(log(Weight)
    2.938051
```

Gráficamente este quedaría de la siguiente manera:

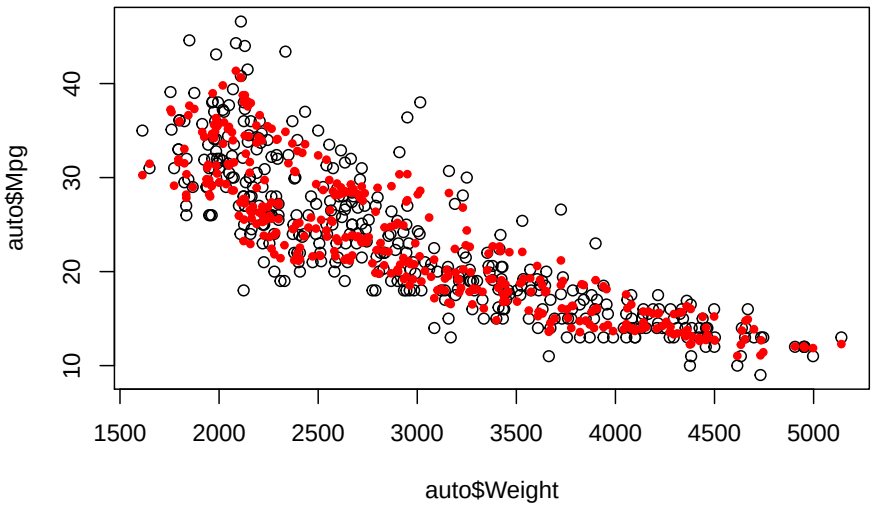


Figura 52

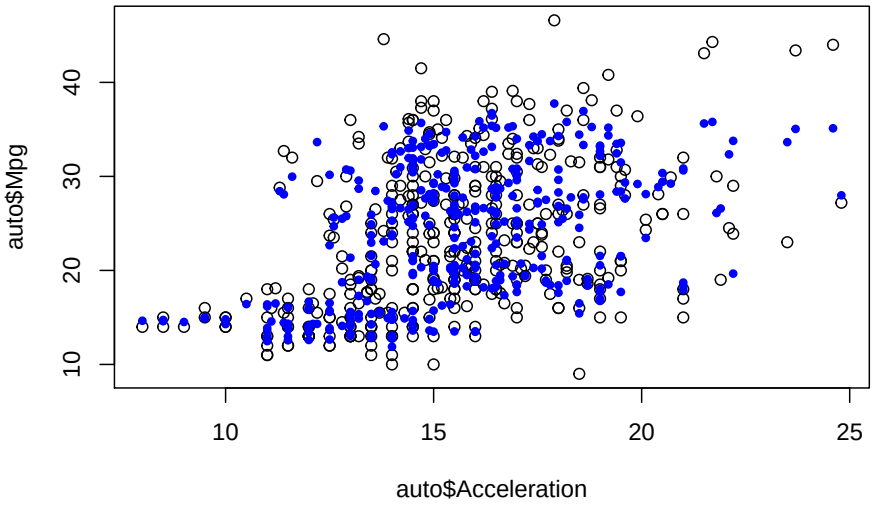


Figura 53

Por completitud, podríamos usarlo también en cross-validation para comprobarlo con un conjunto de test.

Tendríamos por tanto los siguientes modelos para KNN:

$$Mpg \sim . - Acceleration \tag{5}$$

$$Mpg \sim Acceleration + I(log(Weight)) + I(log(Displacement)) \tag{6}$$

Comparándolos gráficamente vemos que son similares, aunque el que intenta evitar el overfitting (en color azul en la gráfica), tiene menor dispersión:

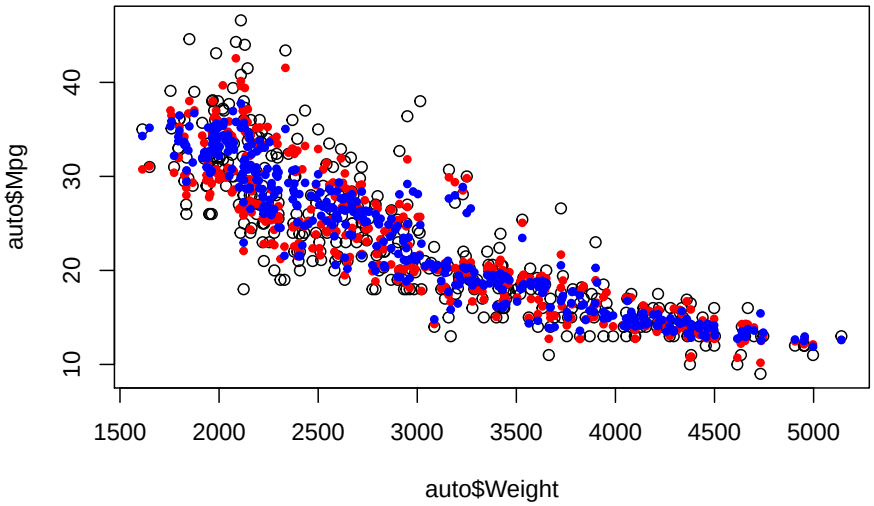


Figura 54

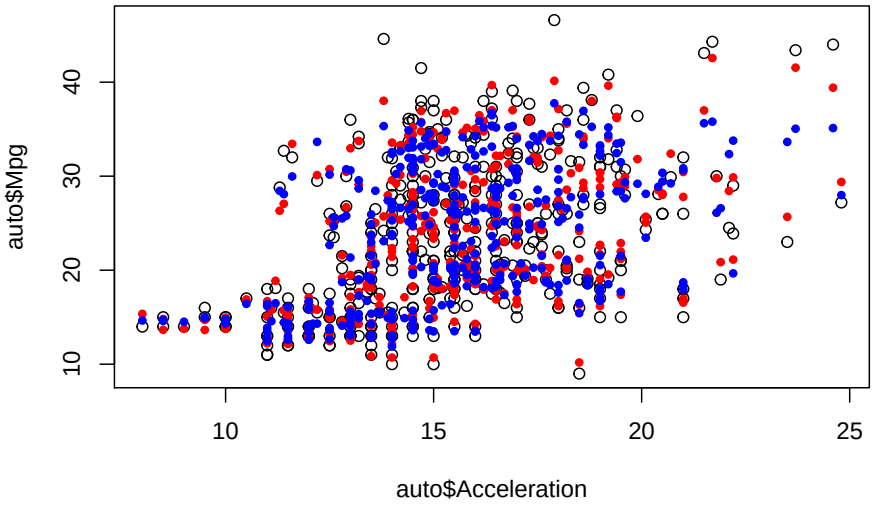


Figura 55

2.6. Comparativa de los ajustes anteriores con cross-validation

Recordamos los modelos obtenidos.

LM:

$$Mpg \sim Weight + Model\_year + I(log(Weight) \tag{7}$$

$$Mpg \sim Acceleration + I(log(Weight)) + I(log(Displacement)) \tag{8}$$

KNN:

$$Mpg \sim . - Acceleration \tag{9}$$

$$Mpg \sim Acceleration + I(log(Weight)) + I(log(Displacement)) \tag{10}$$

Con el proceso de cross-validation dividimos el dataset en N subconjuntos (folds) y repetimos el entrenamiento N veces. Cada entrenamiento se aplica reservando uno de los subconjuntos como test y entrenando con el resto. Al final, el error obtenido para el modelo es la media de los errores en cada fold. La elección del número de folds es importante y si el problema lo permite (en términos de gasto computacional), se debería probar con varios. En este caso hemos utilizado 5 folds.

Con esto conseguimos no desperdiciar el conocimiento del conjunto de test y no guiarnos por una única evaluación del modelo.

Aplicando este proceso obtenemos los siguiente:

```
Regresión 1:  9.333066
Regresión 2: 17.10519
KNN 1:        7.291517
KNN 2:       18.43846
```

Los resultados nos muestra que los modelos con los que obtuvimos mejores valores de R^2 y RSME en sus apartados han acabado con mejor RSME tras el cross-validation. También apreciamos que con KNN conseguimos ligeramente mejores resultados que regresión lineal/no-lineal.

Por completitud, mostramos también los resultados en training:

```
Regresión 1:  9.159891
Regresión 2: 16.62285
KNN 1:        3.659828
KNN 2:        8.642349
```

Que nos muestran que ninguno de los modelos LM estaba haciendo overfitting. Adicionalmente, en KNN existe una diferencia significativa entre training y test.

2.7. Comparativa de tests

Para comparar los algoritmos vamos a aplicar test estadísticos en base a los resultados obtenidos en múltiples datasets. Para asegurar la igualdad de condiciones los algoritmos hacen uso de parámetros genéricos y utilizan las mismas particiones de cross-validation.

Estas son las tablas de resultados que tenemos para test:

out_test_lm	out_test_kknn
0.1909091	0.1000000
0.1000000	1.0294118
0.1000000	0.4339071
0.1000000	0.3885965
0.1548506	0.1000000
0.1000000	0.3061057

Aplicamos el test de **Wilconxon** a LM y KNN:

```
 V    V
78 - 93

p-value:  0.7660294
```

Obtenemos un ranking de 78 para LM y 93 para KNN, con un p-valor de 0.77 (o nivel de confianza del 33 %).

Esto nos dice que gana KNN pero puesto que el p-value no es lo suficientemente grande no podemos afirmar con un nivel alto de significación que las diferencias entre los tests sean notorias.

Ahora aplicamos en test de **Friedman** a los dos algoritmos anteriores junto al algoritmo M5:

```
Friedman rank sum test

Friedman chi-squared = 8.4444, df = 2, p-value = 0.01467
```

El p-value es <0.05 por lo que podemos concluir que al menos hay un par de algoritmos de calidad diferente.

Vemos cuáles de ellos lo son haciendo el test post-hoc de **Holm**:

```
Pairwise comparisons using Wilcoxon signed rank exact test

  1     2
2 0.580 -
3 0.081 0.108

P value adjustment method: holm
```

Con el test post-hoc de HOLM podemos asegurar que 3-1 (M5 vs LM) son diferentes. También podemos afirmar M5 respecto de KNN pero con un nivel de confianza menor.

De KNN y LM no podemos afirmar nada puesto que el p-valor es extremadamente grande.

3. Clasificación: Análisis Estadístico de Datos

3.1. Introducción

Para el problema de clasificación hacemos uso del dataset **haberman** [3], que codifica el ratio de supervivencia de pacientes operados de cáncer de pecho en el Hospital Universitario de Chicago, en base a las siguientes características:

1. **Age**: Indica la edad del paciente en el momento de la operación.
2. **Year**: Los dos últimas cifras del año en el que se operó el paciente.
3. **Positive**: Número de nodos auxiliares positivos detectados. Esta variable hace referencia a los ganglios linfáticos que dan positivos como presentes de cáncer. A mayor número de nodos detectados, mayor es la gravedad del cáncer.

   Aunque normalmente la primera zona de propagación del cáncer son estos nodos, no es la única medida de la seriedad, pues este puede propagarse a otras zonas del cuerpo. En principio no deberíamos descartar la posibilidad de que puede haber casos de no supervivencia con bajo número de positivos.

   Viendo que solo tenemos esta medida del cáncer en el dataset es posible que la operación que recibieron los pacientes sea algún tipo de cirugía de ganglios linfáticos, donde el cirujano intenta extraer los nodos afectados por el tumor. Por consiguiente, cuanto mayor es la cantidad de nodos detectados, más complicaciones pueden acarrearse de la operación [4, 5].

El objetivo es poder clasificar, en base a los tres atributos, si los pacientes pueden sobrevivir 5 años o más:

4. **Survival**: Sí/No indicando la supervivencia del paciente tras 5 años.

Contamos por tanto con un problema de clasificación binario en base a tres características, y con un número total de 306 instancias.

La descripción del problema nos da alguna información adicional sobre las variables:

1. **Age**: Variable numérica discreta, contamos con valores enteros en el rango [30,83].
2. **Year**: Variable numérica discreta, contamos con valores enteros en el rango [58,69].
3. **Positive**: Variable numérica discreta, contamos con valores enteros en el rango [0,52].
4. **Survival**: Variable binaria.

Hipótesis de partida

- **H.1**: Habrá menor ratio de supervivencia cuanto mayor sea el número de nodos positivos encontrados.
- **H.2**: Habrá mayor ratio de supervivencia cuanto más joven sea el paciente.
- **H.3**: El rango de Year es pequeño. La influencia de esta variable creemos que podría darse solo si durante ese período se hubieran descubierto técnicas mejores de cirugía. Este razonamiento va orientado de cara a la población y no a la muestra. Puesto que contamos con datos de un solo hospital durante pocos años, es posible que el equipo de cirugía hubiera sido el mismo para la mayoría de pacientes.

- **H.4**: Podría haber relación entre la edad y el número de positivos, posiblemente indicando lo tardío que se descubre el cáncer.
- **H.5**: La bibliografía nos dice que el cáncer puede aparecer a diferentes edades con diferentes factores de riesgo (alcoholismo, herencia genética...). Podría ser que el número de variables con las que contamos sea insuficiente para la clasificación. (Hipótesis no demostrable en el EDA).

3.2. Análisis Estadístico de Datos

R por defecto nos carga las variables Age, Year y Positive como numéricas y Survival como carácter. Transformamos Survival a factor categórico, el resto de variables las mantenemos en su formato.

3.2.1. Análisis univariable

La cabecera de los datos nos quedan por tanto de la siguiente manera:

Age	Year	Positive	Survival
38	59	2	No
39	63	4	No
49	62	1	No
53	60	2	No
47	68	4	No
56	67	0	No

Con las siguientes medidas estadísticas principales:

```
      Age             Year          Positive       Survival
 Min.   :30.00   Min.   :58.00   Min.   : 0.000   No :225
 1st Qu.:44.00   1st Qu.:60.00   1st Qu.: 0.000   Yes: 81
 Median :52.00   Median :63.00   Median : 1.000
 Mean   :52.46   Mean   :62.85   Mean   : 4.026
 3rd Qu.:60.75   3rd Qu.:65.75   3rd Qu.: 4.000
 Max.   :83.00   Max.   :69.00   Max.   :52.000
```

En las distribuciones de los clasificadores nos fijaremos más adelante. Aquí hacemos notar que los valores de salida en nuestros datos están bastante desbalanceados, solo un 26.5 % de los paciente sobrevivieron a los 5 años.

El dataset cuenta con valores 17 repetidos, concretamente las siguientes ocurrencias:

Age	Year	Positive	Survival	Age	Year	Positive	Survival
37	63	0	No	55	58	1	No
37	63	0	No	55	58	1	No
38	60	0	No	56	60	0	No
38	60	0	No	56	60	0	No
41	65	0	No	57	64	0	No
41	65	0	No	57	64	0	No
43	64	0	Yes	61	59	0	No
43	64	0	Yes	61	59	0	No
44	61	0	No	61	59	0	No
44	61	0	No	62	66	0	No
48	58	11	Yes	62	66	0	No
48	58	11	Yes	63	63	0	No
50	61	0	No	63	63	0	No
50	61	0	No	65	64	0	No
54	62	0	No	65	64	0	No
54	62	0	No	67	66	0	No
				67	66	0	No

Existen dos posibilidades para el origen de estos datos:

1. Errores en la introducción de los datos. Entradas repetidas por error.
2. Son entradas de pacientes distintos casualmente con las mismas características.

Apreciamos que en la mayoría de instancias el número de Positive es cero. En el apartado 3.3 se explica como este es un valor bastante frecuente en los datos pero que posiblemente se deba a que es una medida de los nodos **auxiliares** y no un error de codificación.

Como en este caso tenemos muy pocas variables (y un número moderado de entradas, 306), es probable que los pacientes coincidan en las características. Además, podemos ver que las entradas en la mayoría de los casos solo están duplicadas (solo hay una entrada triplicada).

Por tanto proseguimos sin eliminar estas instancias duplicadas.

Mostramos scatterplots univariables:

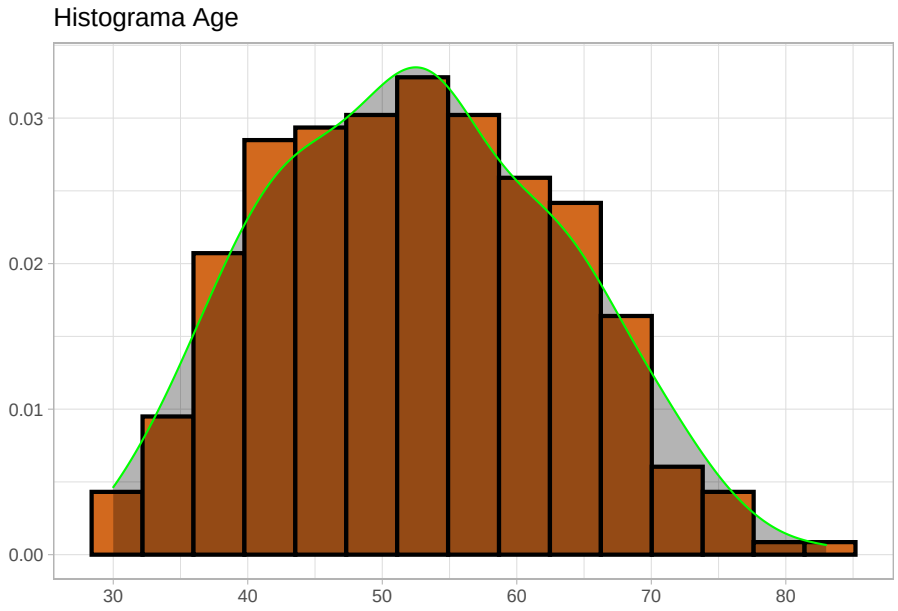


Figura 56

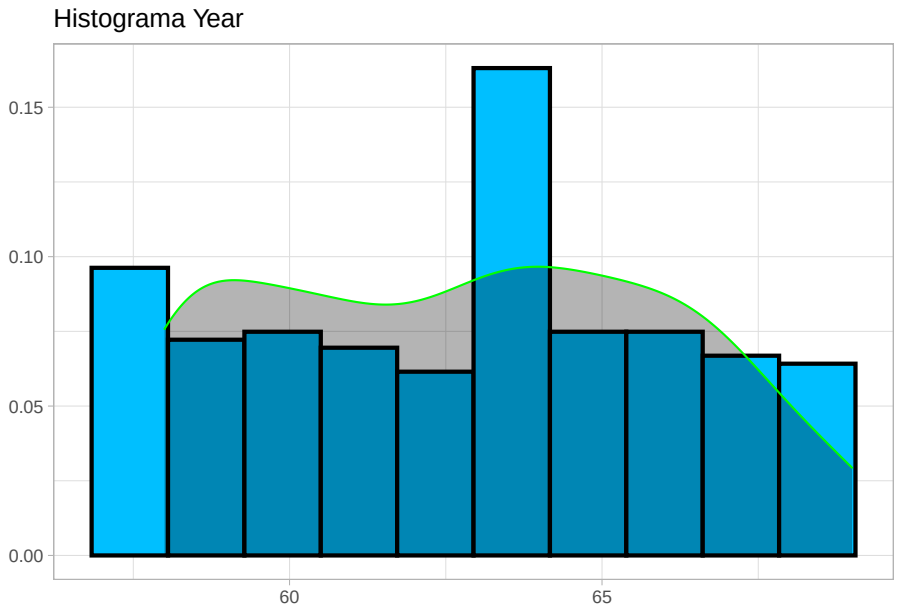


Figura 57

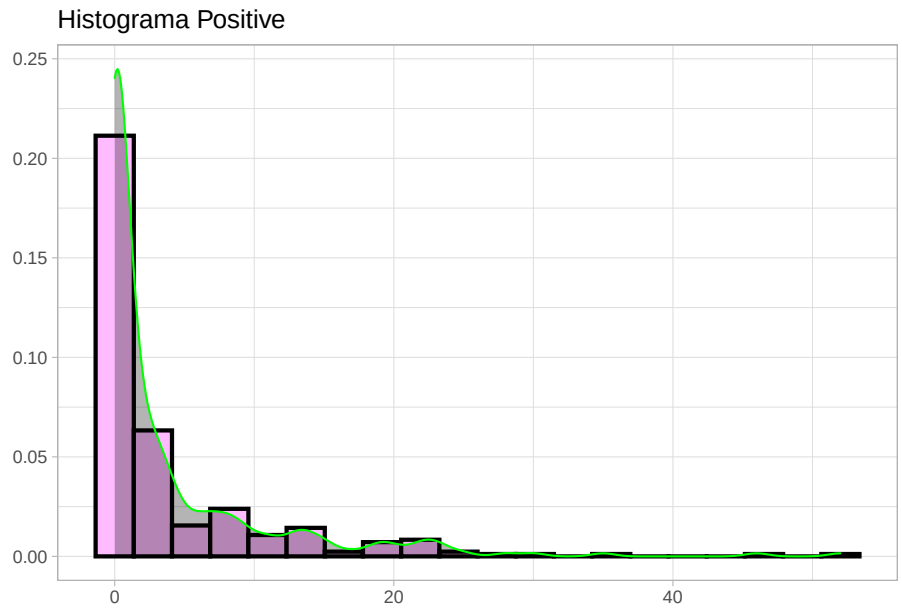


Figura 58

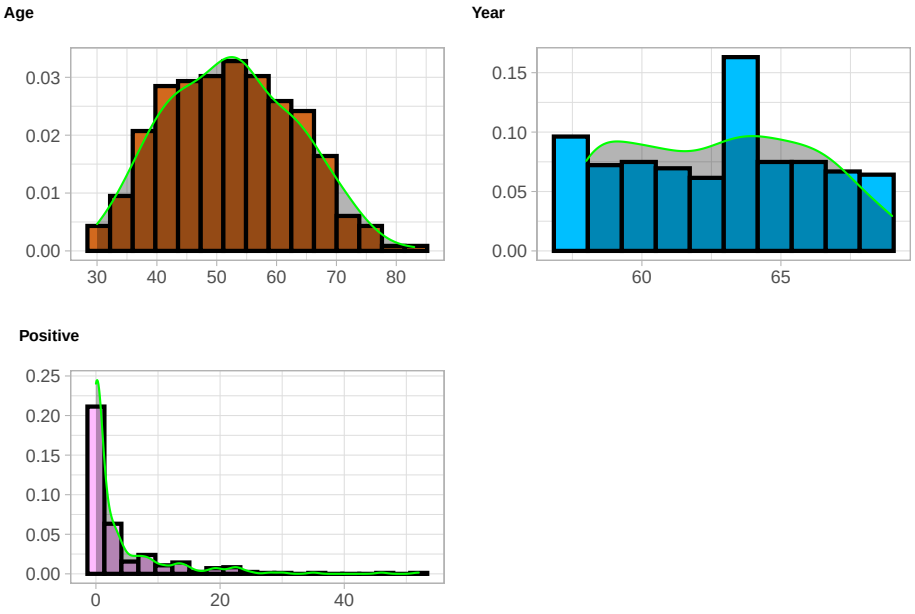


Figura 59

Y boxplots sobre las distribuciones de los datos:

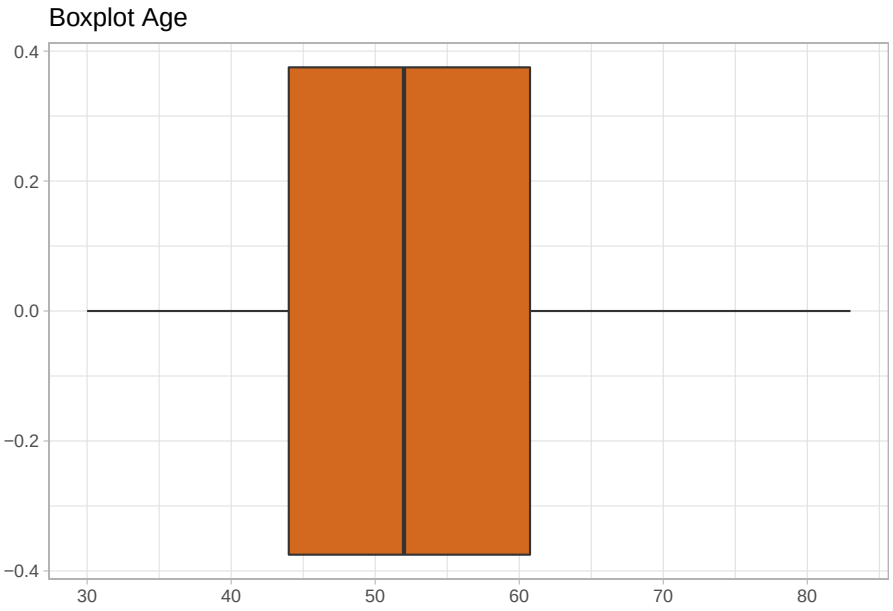


Figura 60

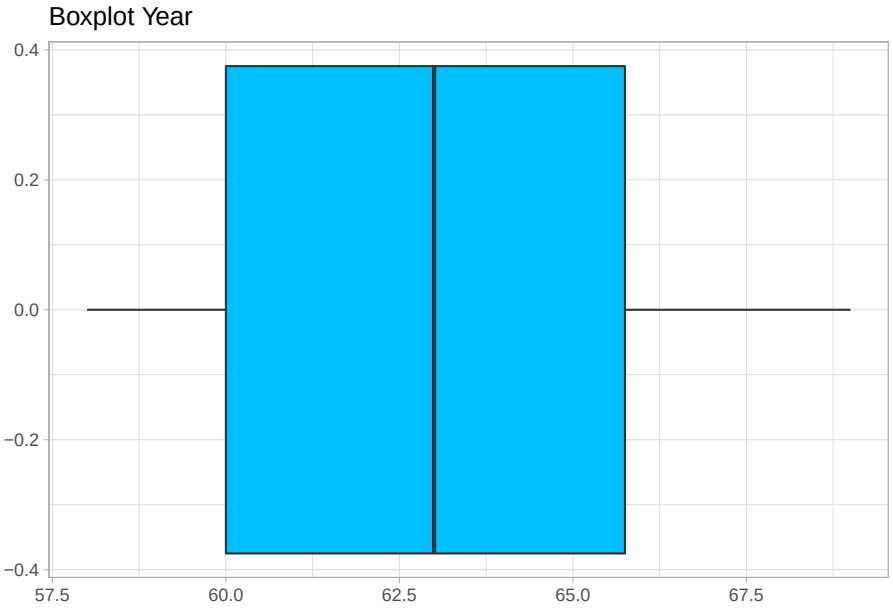


Figura 61

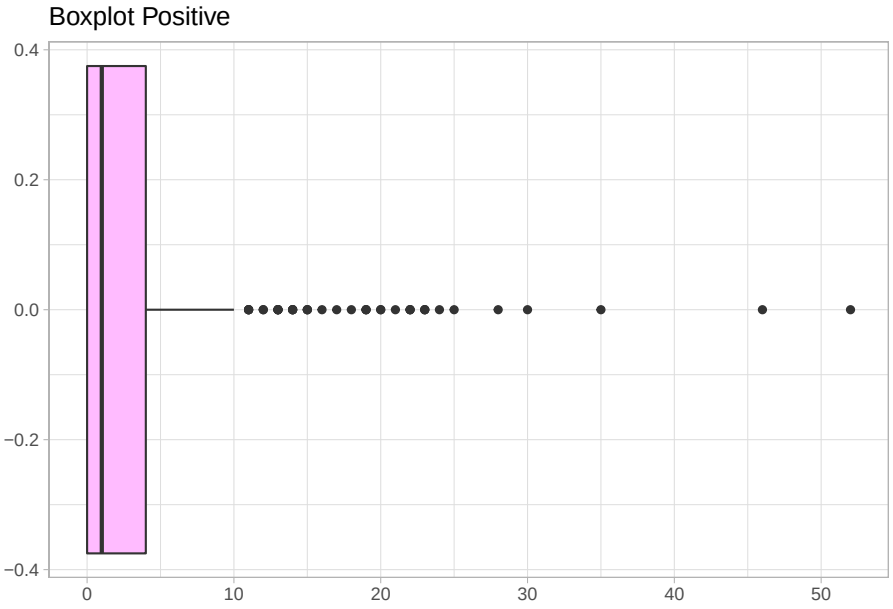


Figura 62

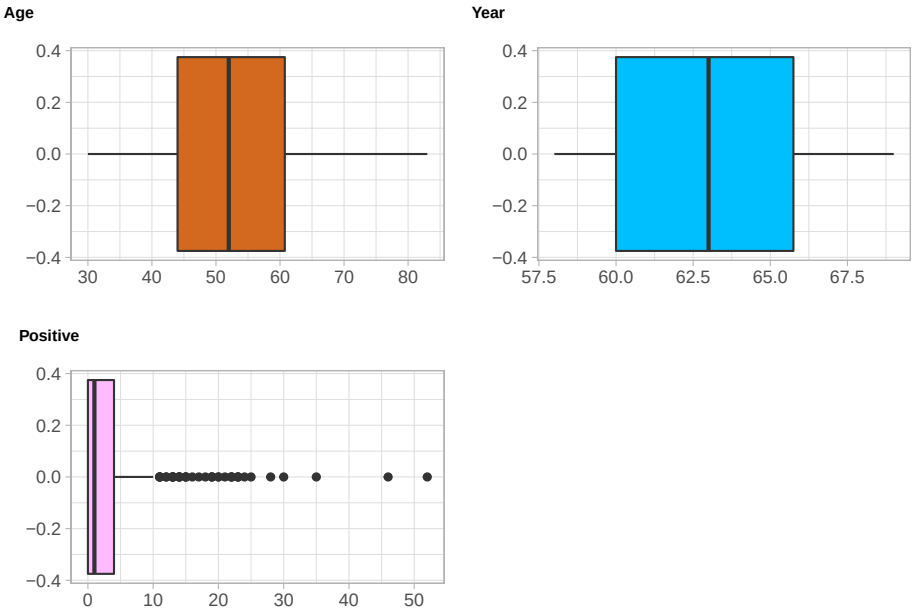


Figura 63

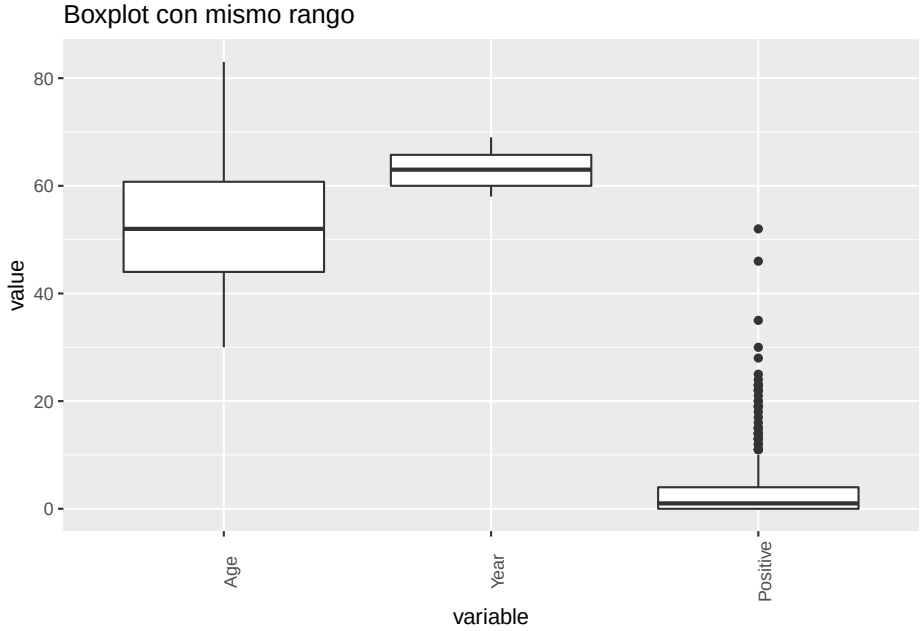


Figura 64

Podemos comparar los rangos intercuartiles si estandarizamos antes el dataset

```
     Age     Year Positive
1.550430 1.769555 0.556355
```

También podemos ver la distancia entre mínimos y máximos

```
     Age     Year Positive
4.905839 3.385235 7.232616
```

Ya la descripción del problema nos lo decía, los rangos en los que se distribuyen los datos son muy diferentes entre sí. Es necesario aplicar un proceso de estandarización antes de clasificar.

3.2.2. Missing values

Nos cuestionamos la ocurrencia de instancias con cero en el número de positivos. Podríamos pensar que se trata de una codificación de missing values si nos aseguramos que la operación consistía en eliminar estos nodos positivos.

Si revisamos la información que tenemos, estos nodos positivos se denominan auxiliares, y una mayor investigación del problema por internet nos asegura de que estos valores de cero no se corresponden a missing values.

Pese a ello, lo apropiado habría sido ponerse en contacto con los creadores del dataset y preguntar por la forma de codificar los datos que habían usado.

Si hubiéramos descubierto que sí lo son, y tras ver que una gran parte de las instancias contienen este valor, habríamos tenido que buscar algún tipo de imputación para rellenar estos valores. Puesto que tendríamos un número pequeño de valores reales, probablemente habríamos optado por KNN o interpolación lineal.

Age Para esta variable no contamos con valores de todos los años, y hemos visto que en general no están equitativamente distribuidos:

```
Años:   30 31 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
Conteo:  3  2  2  7  2  2  6 10  6  3 10  9 11  7  9  7 11  7 10 12  6 14
Años:   53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
Conteo: 11 13 10  7 11  7  8  6  9  7  8  5 10  5  6  2  4  7  1  4  2  2
Años:   75 76 77 78 83
Conteo:  1  1  1  1  1
```

Year Aquí si contamos con valores en todos los años, aunque con más instancias en los iniciales:

```
Años:   58 59 60 61 62 63 64 65 66 67 68 69
Conteo: 36 27 28 26 23 30 31 28 28 25 13 11
```

Positive Esta variable parece llevar una distribución exponencial y probablemente por ello aparezcan tantas posibles anomalías.

3.2.3. Análisis sobre las distribuciones

Ninguna variable parece seguir una distribución semejante a una distribución normal. Lo aseguramos con un test estadístico (Shapiro-Wilk test):

vars	statistic	p_value	sample
Age	0.9894580	0.0260466	306
Year	0.9467912	0.0000000	306
Positive	0.6153079	0.0000000	306

También lo mostramos gráficamente con plots Q-Q, donde se ve que las distribuciones no siguen los cuartiles normales, mayormente en las colas:

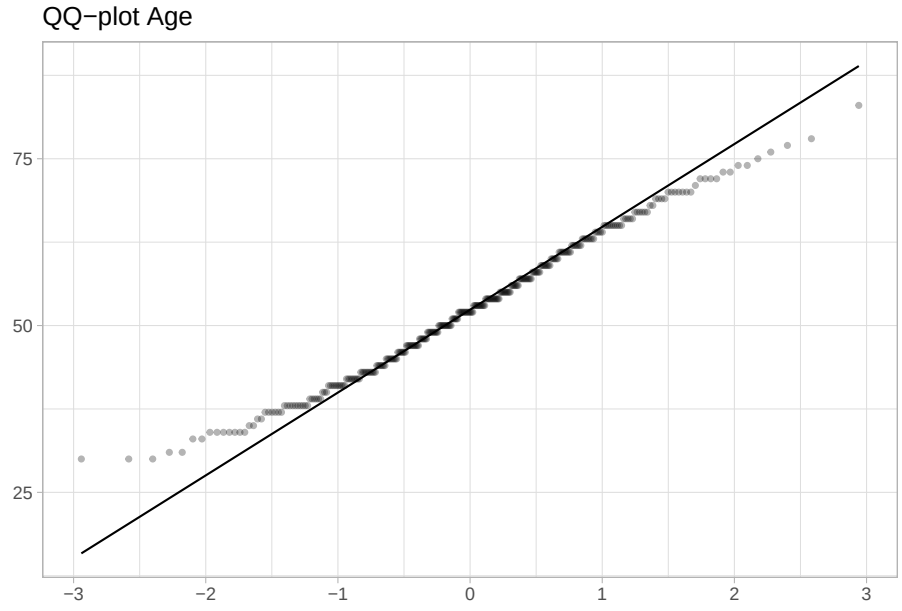


Figura 65

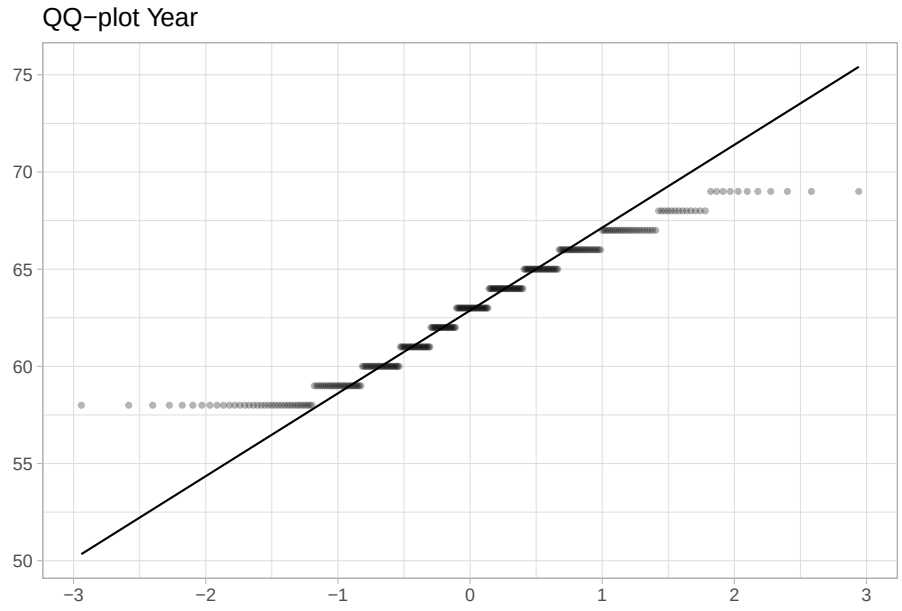


Figura 66

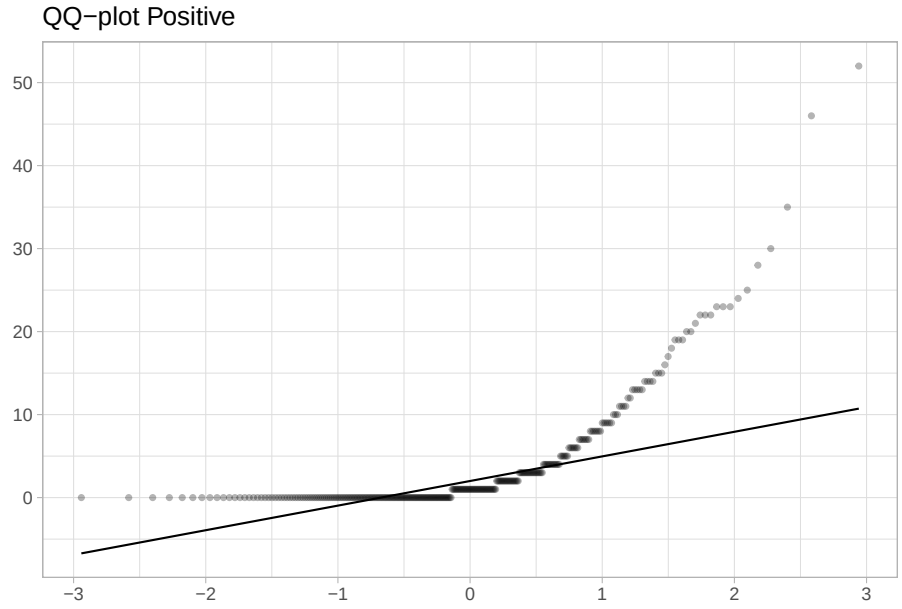


Figura 67

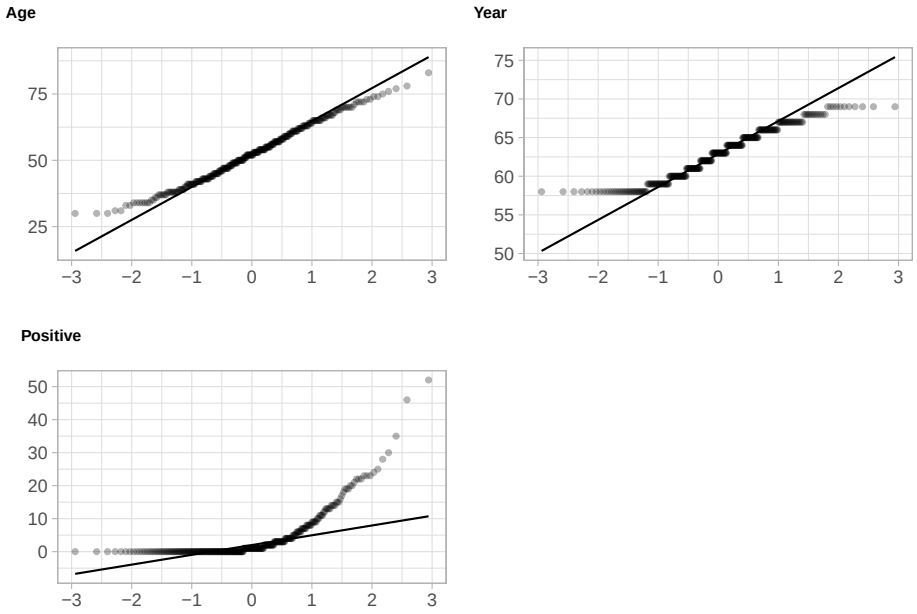


Figura 68

La variable Positive parece seguir una distribución exponencial. Podemos hacer un plot de los supuestos cuartiles para hacernos una idea de cómo se asemeja:

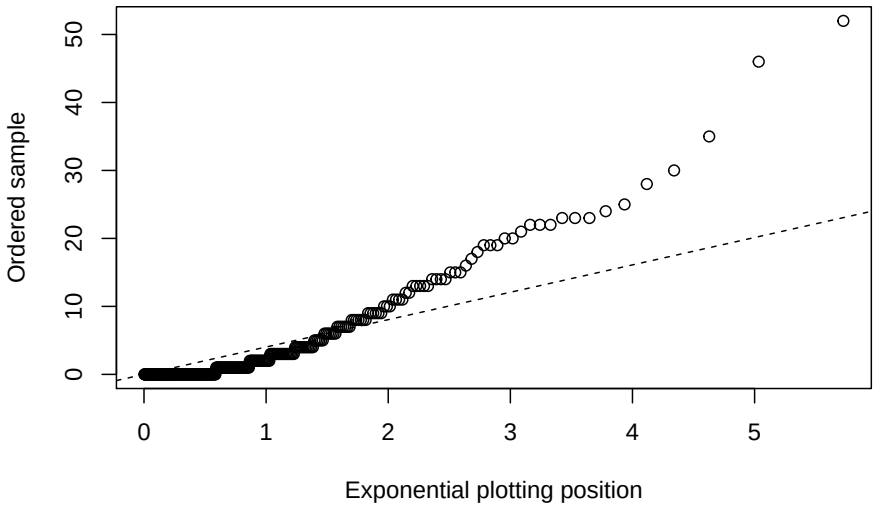


Figura 69

Ya el gráfico nos muestra que no se va a asemejar, pero podríamos verificarlo con más precisión haciendo uso de un test de Kolmogorov-Smirnov.

Skewness Claramente la única variable con skewness es Positive, con un grado positivo bastante alto:

```
Positive:  2.969176
Year:  0.07836828
Age:  0.1457859
```

3.2.4. Transformaciones

El paquete *caret* nos sugiere una estandarización a media cero y desviación típica 1. Para un problema de clasificación esto es totalmente necesario puesto que no queremos que los diferentes rangos de las variables hagan que haya información de más peso que otra.

A excepción de KNN, los métodos de clasificación que vamos a usar necesitan la normalidad en los datos. En un caso real, si quisiéramos aplicar sí o sí esos métodos deberíamos averiguar previamente la distribución exacta que siguen esos datos para transformarla apropiadamente.

Por otro lado, transformaciones de Yeo-Johnson o BoxCox para reducir la skewness en Positive carece de lógica puesto que no sigue una forma normal.

3.2.5. Anomalías

La única variable en la que podríamos considerar anomalías es Positive. Tanto para la edad como para los años no tiene sentido, además de que hemos visto en los boxplots que en ellas todos los valores caen en el 95 % de la distribución.

A la hora de considerar los outliers en Positive, tal y como habíamos mencionado en la descripción del problema, debemos recordar que un alto número de nodos detectados complica la operación y el pronóstico para el paciente.

Contrariamente a esta idea, podemos ver que para aquellas instancias con un gran número de positivos la cantidad de sobrevivientes en nuestro dataset está equilibrada:

```
No Yes
17  23
```

Viendo que la distribución está equilibrada en estos posibles valores anómalos y tampoco tenemos conocimiento suficiente sobre el problema para considerar cuándo un número de positivos es demasiado alto, proseguimos manteniéndolos en nuestro dataset.

3.2.6. Análisis de correlación

Como este es un problema de clasificación, necesitamos eliminar aquellas variables correladas para que la información se aporte de manera equitativa. Las gráficas no nos han dado ninguna señal de una posible correlación, pero debemos asegurarnos de forma estadística.

Tenemos que tener en cuenta que las variables no siguen distribuciones normales. Aunque el coeficiente de Pearson no asume normalidad (si asume varianza y covarianza finitas), podemos usar el coeficiente de Kendall para los cálculos. Independientemente del método usado vamos a obtener las mismas correlaciones en este dataset, solo varía la fuerza con la que se dan.

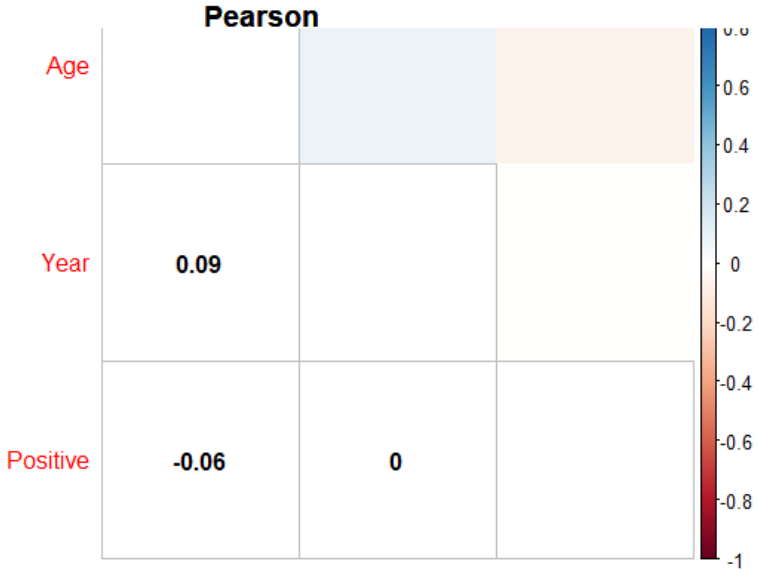


Figura 70

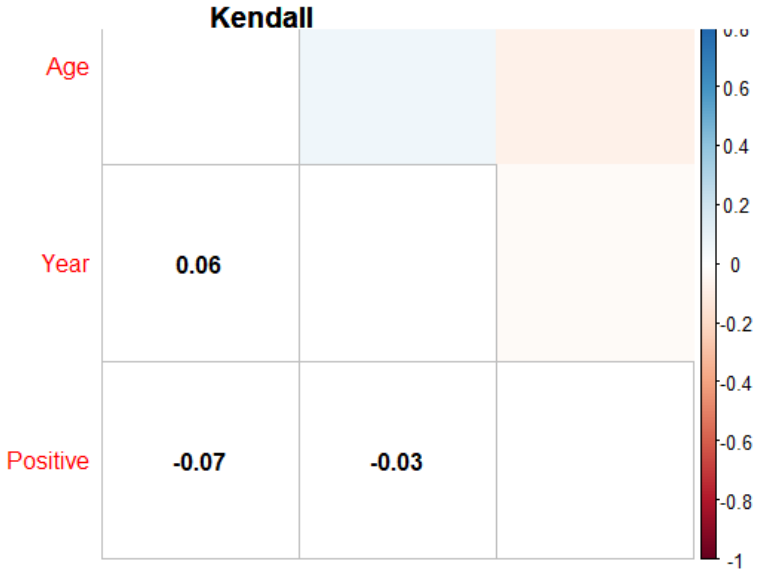


Figura 71

Las matrices de correlación nos muestran que no existe correlación alguna entre las variables, y un con un conjunto de scatterplot lo podemos ver gráficamente:

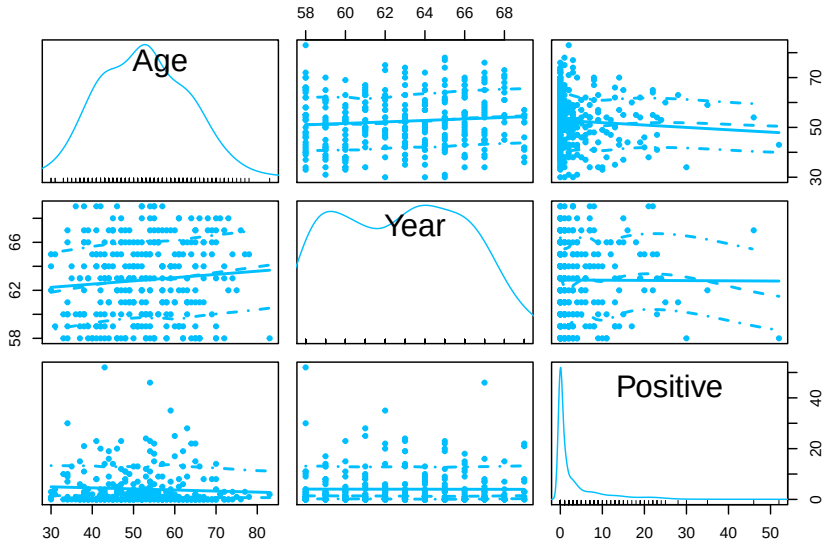


Figura 72

Adicionalmente, mostramos la distribución de las variables con su clasificación:

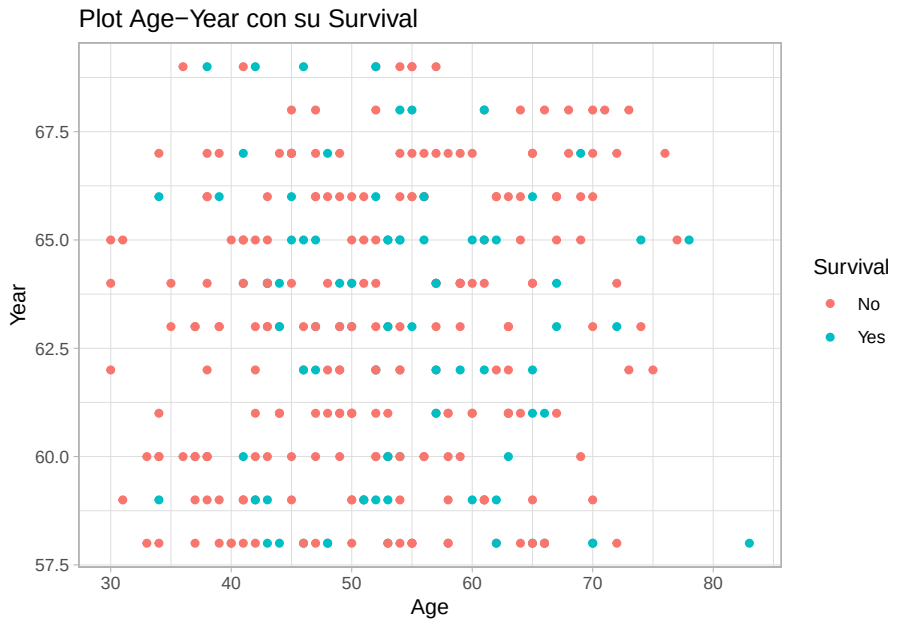


Figura 73

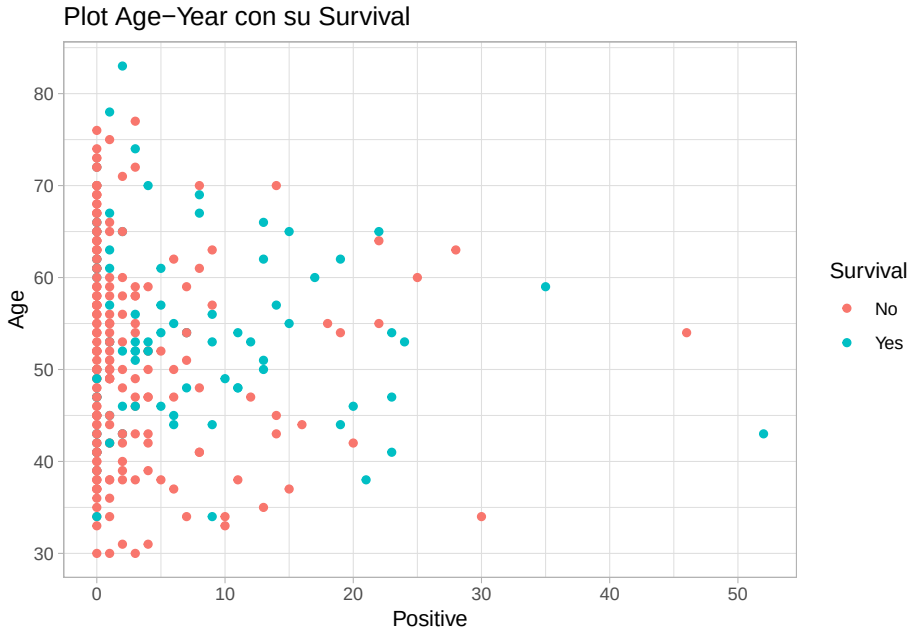


Figura 74

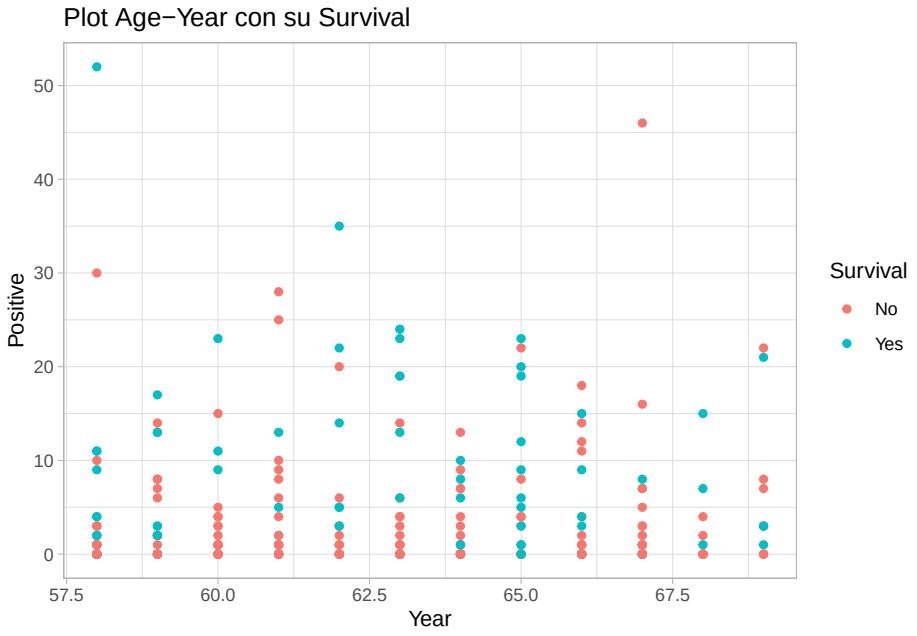


Figura 75

No se aprecia ninguna relación visual que nos ayude a clasificar el Survival.

3.2.7. Tratamiento de variables y ordenaciones

Volvemos a mostrar la cabecera de los datos:

Age	Year	Positive
38	59	2
39	63	4
49	62	1
53	60	2
47	68	4
56	67	0

Para este dataset contamos con tres clasificadores con información de distinto tipo y bien organizada, por lo que no necesitamos hacer ningún tipo de ordenación/tratamiento. No existe ninguna relación entre variables sobre la información que codifican (en el sentido de que podrían agruparse).

La variable Year solo indica las dos últimas cifras del año de operación, pero como todas las instancias son del mismo siglo nos resulta más conveniente tenerla así.

Sobre esta variable, existe la posibilidad de hacer una discretización en intervalos, lo cuál podría ayudarnos en la clasificación. Pero observando las gráficas, vemos que no parece haber ningún agrupamiento o tendencia entre las edades y las etiquetas. Por tanto, antes de hacer esto se debería consultar con el experto, por si hubiera un significado especial al agruparlas, y puesto que no tenemos conocimiento suficiente sobre la materia como para deducirlo nosotros, se mantienen como están.

3.2.8. Resolución de hipótesis

Nos habíamos planteado las siguientes hipótesis

- **H.1**: Habrá menor ratio de supervivencia cuanto mayor sea el número de nodos positivos encontrados.

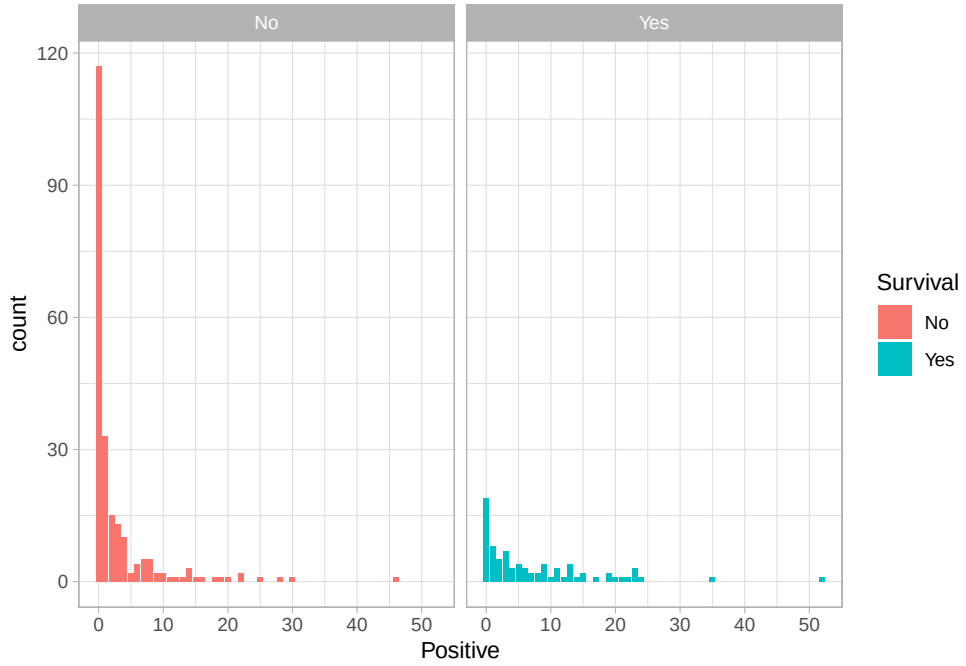


Figura 76

```
No Yes
17  23
```

Vemos que la hipótesis no es cierta. Creemos que se debe a que el número de nodos no es el clasificador más importante.

- **H.2**: Habrá mayor ratio de supervivencia cuanto más joven sea el paciente.

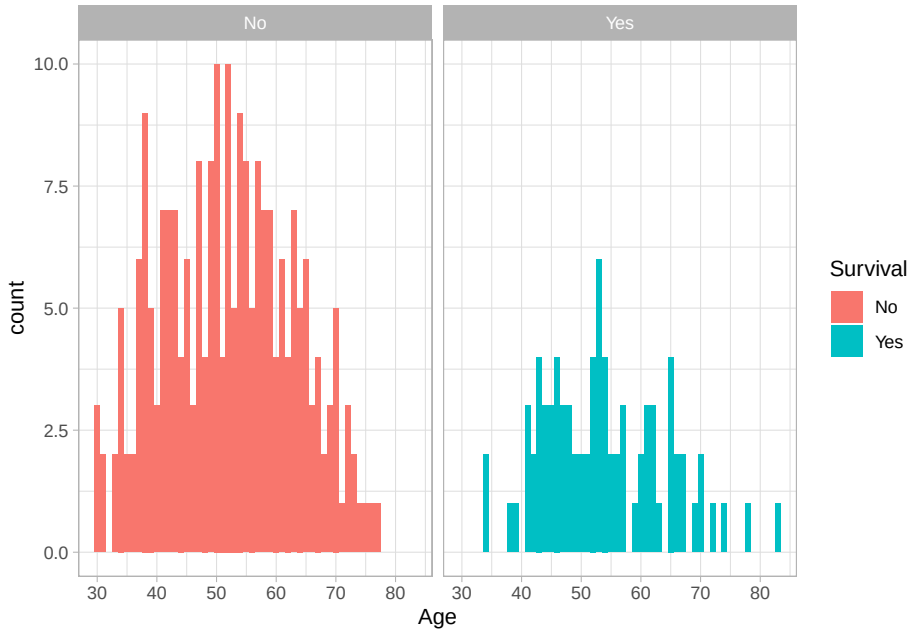


Figura 77

Si miramos los pacientes con edades <40 nos sale todo lo contrario:

```
No Yes
36   4
```

- **H.3**: El rango de Year es pequeño. La influencia de esta variable creemos que podría darse solo si durante ese período se hubieran descubierto técnicas mejores de cirugía. Este razonamiento va orientado de cara a la población y no a la muestra. Puesto que contamos con datos de un solo hospital durante pocos años, es posible que el equipo de cirugía hubiera sido el mismo para la mayoría de pacientes.

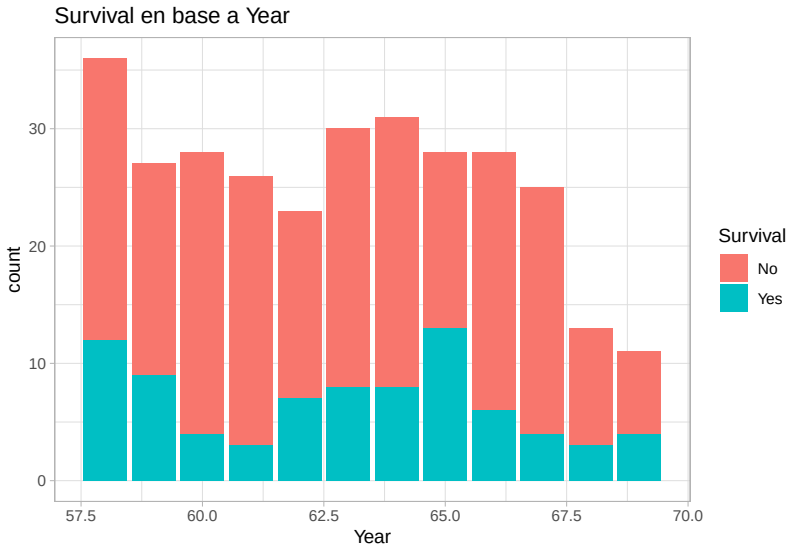


Figura 78

Decrementa el número de datos en años superiores, aunque la proporción es bastante similar con los datos que tenemos, por lo que no podemos confirmar la hipótesis:

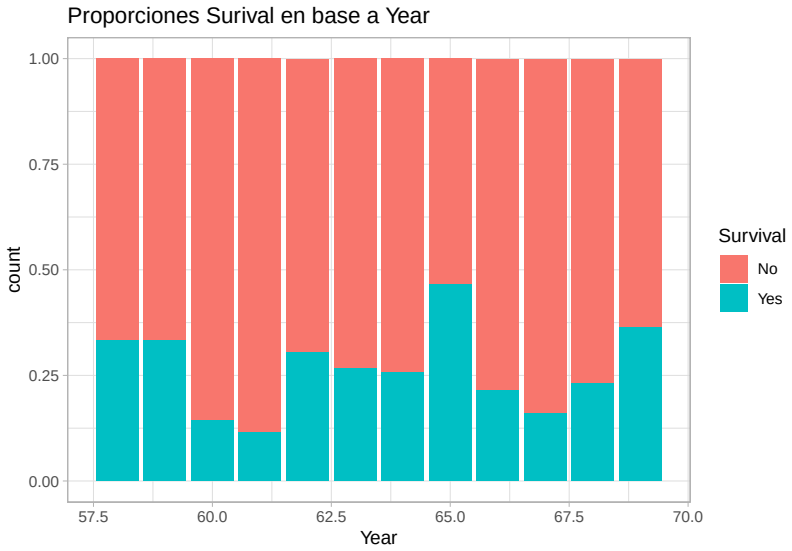


Figura 79

- **H.4**: Podría haber relación entre la edad y el número de positivos, posiblemente indicando lo tardío que se descubre el cáncer.

Un scatterplot no nos muestra visualmente ninguna aparente relación, y el análisis de correlación no nos había indicado nada:

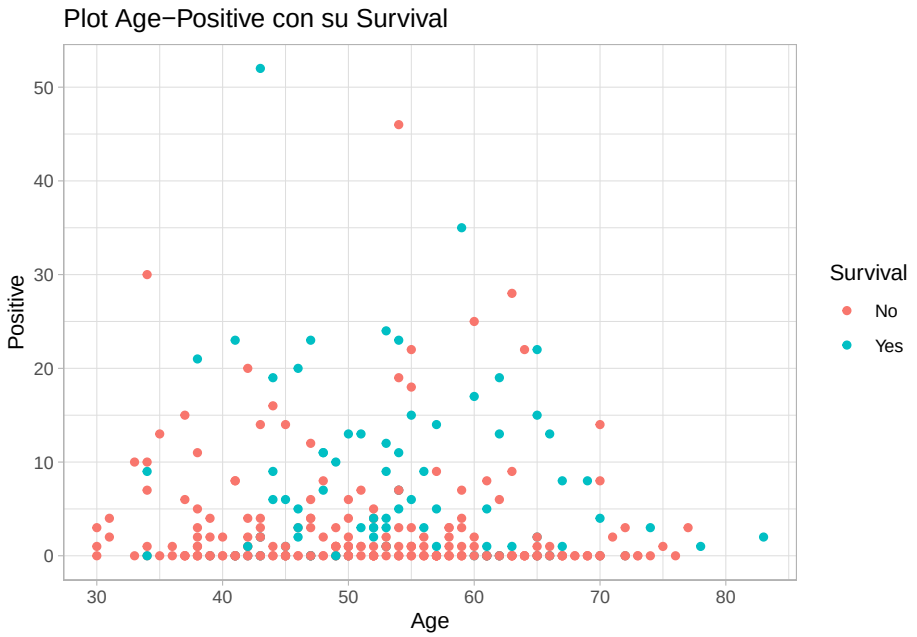


Figura 80

3.3. Conclusiones

Para terminar, concluímos diciendo que tenemos un dataset con pocas variables, pero con ninguna correlación entre ellas, favoreciéndonos el problema de clasificación que nos atañe. También hemos visto ausencia de normalidad en los clasificadores que hará que no cumplamos con las asunciones de LDA.

Además, el dataset se encuentra desbalanceado, contando con más instancias de no supervivientes (en torno al 76 % de los datos). También se aprecian algunas instancias repetidas, sobre las que en principio creemos que se debe a una casualidad debido al bajo número de variables cuyos rangos de valores son pequeños en cada una, pero carecemos de información adicional que nos corrobore la hipótesis.

Adicionalmente, se hace notar una alta cantidad de instancias con valor de Positive cero. Por la descripción de la variable creemos que es un valor correcto y no una codificación de missing value.

El único tratamiento realizado ha sido un preprocesado aplicando una estandarización, preparando el dataset a los algoritmos que se van a utilizar. Por la forma no normal de la distribución en Positive, no deberíamos aplicar algoritmos que la requieran.

4. Técnicas de Clasificación

Antes de empezar, y aunque no se ha implementado en la práctica, hacemos notar que por la descripción del problema parece que cometemos un error mayor cuando clasificamos mal la clase Yes. Por tanto, se podría considerar penalizar más los falsos negativos, de manera que los algoritmos de clasificación intenten cometer menores errores de este tipo.

4.1. Algoritmo KNN

Recordamos los gráficos 1-1 con las clasificaciones, vistos en el EDA.

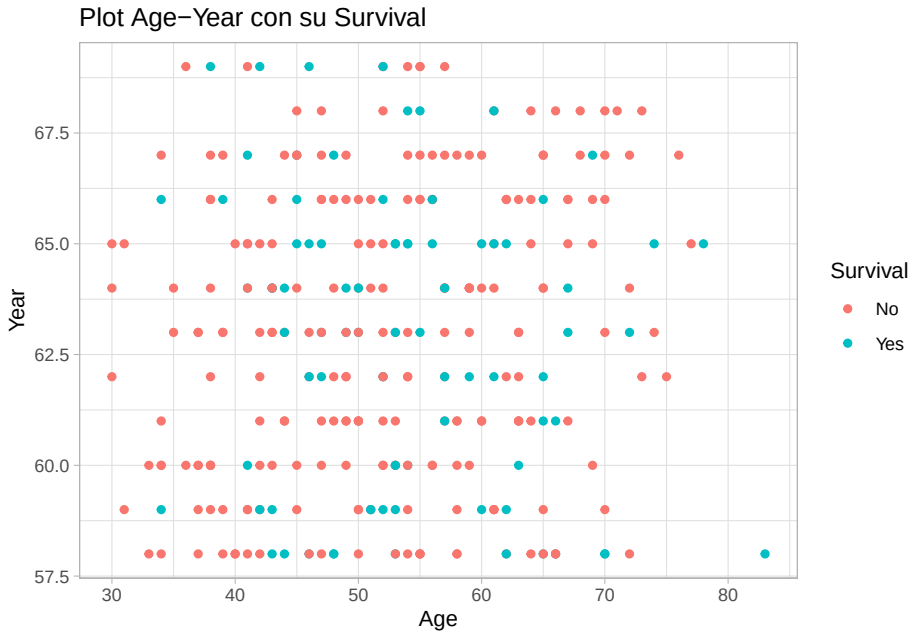


Figura 81

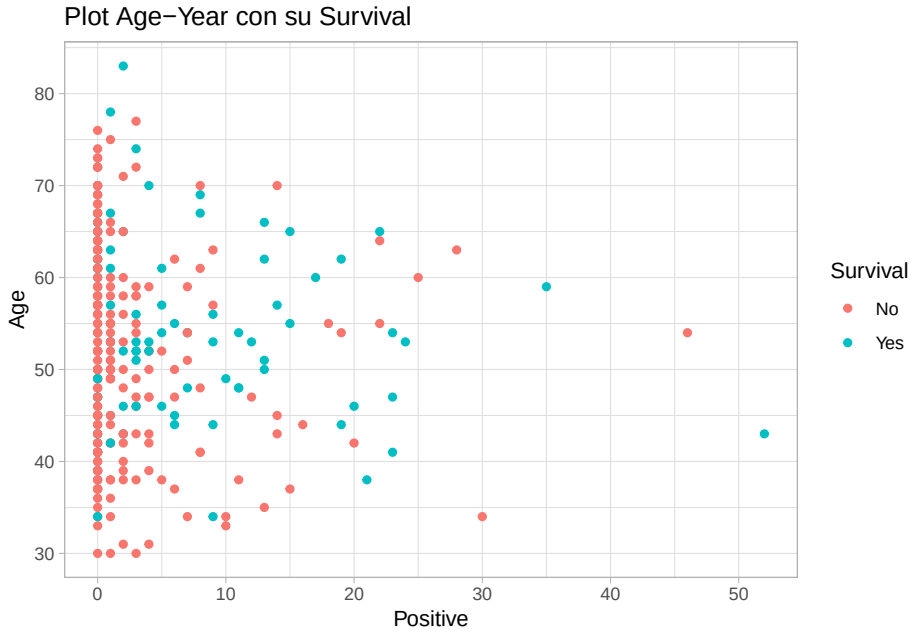


Figura 82

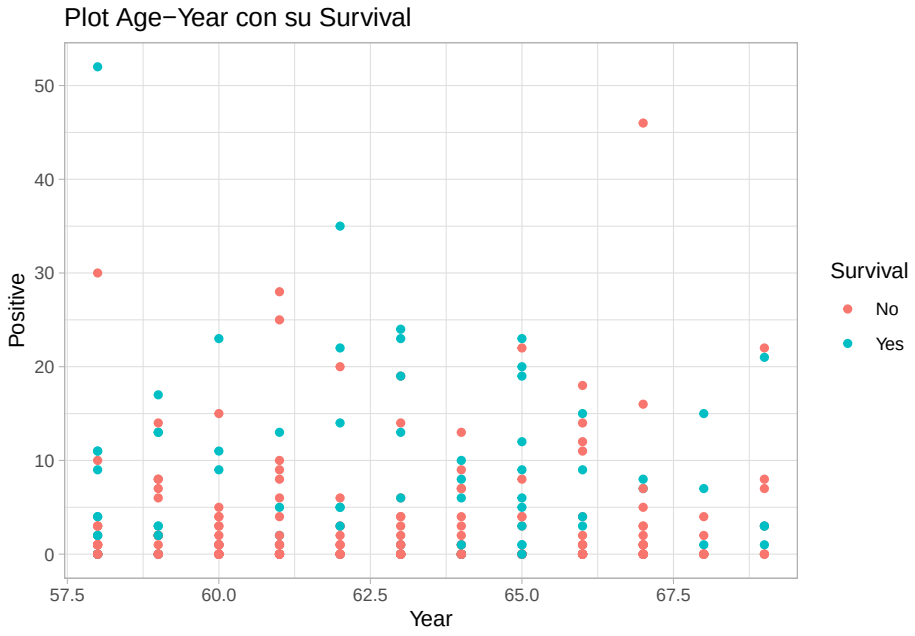


Figura 83

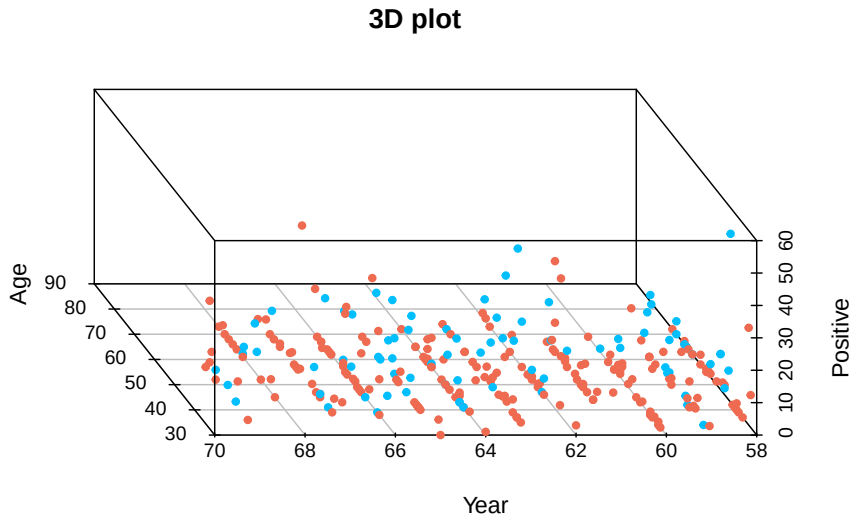


Figura 84

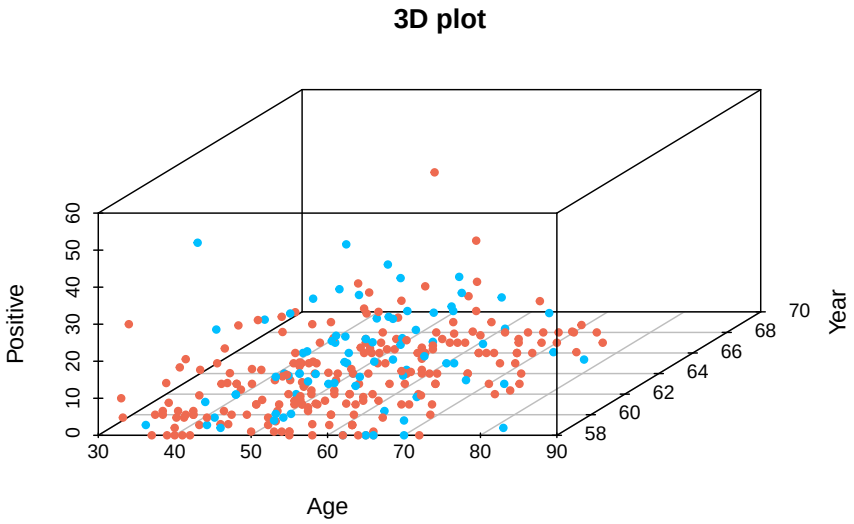


Figura 85

De cara a un algoritmo KNN, apreciamos los datos muy entremezclados, con mayor tendencia a agruparse los no supervivientes (por su alta frecuencia) que los que sí, pero nada en especial que nos llame la atención.

Debido a esto vamos a empezar con un valor de K relativamente bajo y vamos a ir aumentándolo poco a poco. Tenemos que tener en cuenta que un K mayor puede ocasionar overfitting, pero usando técnicas de cross-validation podemos detectarlo con mayor facilidad.

Los resultados usando el paquete caret son los siguientes[1]:

```
k-Nearest Neighbors

275 samples
  3 predictor
  2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 248, 248, 247, 247, 248, 247, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   3  0.6466931  0.05241865
   4  0.6612434  0.07153373
   5  0.6727513  0.05451908
   6  0.6907407  0.08157828
   7  0.6907407  0.07832535
   8  0.7017196  0.08311583
   9  0.7164021  0.13328924
  10  0.7089947  0.12077424
  11  0.7236772  0.15850208
  12  0.7161376  0.14004893
  13  0.7269841  0.16958772
  14  0.7197090  0.14121236
  15  0.7271164  0.16518141

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 15.
```

Vemos que al estar los datos tan entremezclados ni siquiera con un K pequeño aprende bien, es ya con un K medianamente alto (= 15) donde obtiene mayor accuracy en training.

Una vez más probablemente esto se deba a la gran mezcla de los datos, de forma que necesite la "opinión" de un gran número de vecinos para poder predecir con mayor confianza el nuevo valor.

[1] Los datos han sido preprocesados con una estandarización antes de aplicar cualquiera de los algoritmos de la práctica

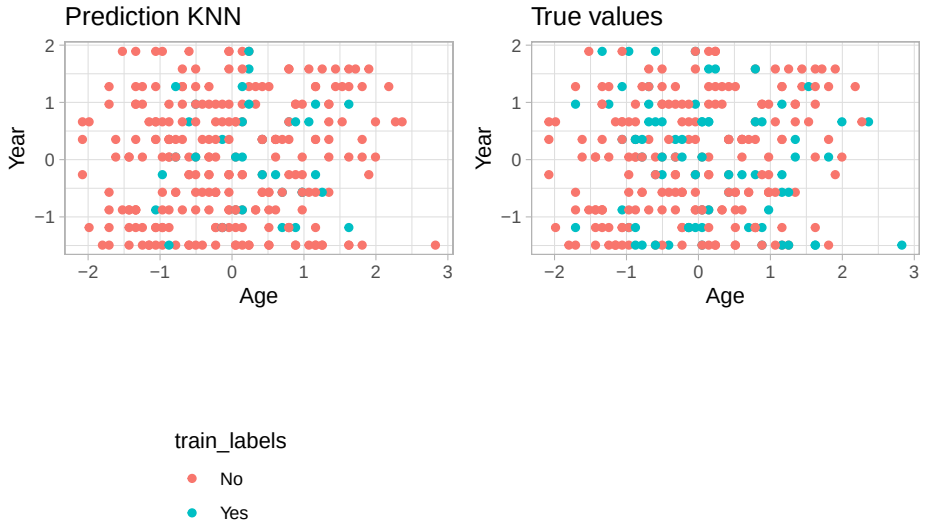


Figura 86: Predicción con K=15 en training

Vemos que el uso de un K alto hace que perdamos puntos de Yes, puesto que no hay suficientes vecinos de la misma clase que refuercen la opinión sobre esa zona del espacio.

Una evaluación con el subconjunto reservado inicialmente como test nos muestra una calidad extrañamente superior que la de training.

```
Test evaluation:
 Accuracy     Kappa
0.8387097 0.2439024

Confusion matrix (in test split):
knnPred No Yes
    No  25   5
    Yes  0   1


Etiquetas reales:
No  No  Yes No  No  No  Yes No  No  No  No  No  No  Yes No  No  Yes Yes
No  No  No  No  No  No  No  Yes No  No  No  No

Predicciones KNN:
No  No  No  No  No  No  No  No  No  No  No  No  No  No  No  No  No  Yes
No  No  No  No  No  No  No  No  No  No  No  No
```

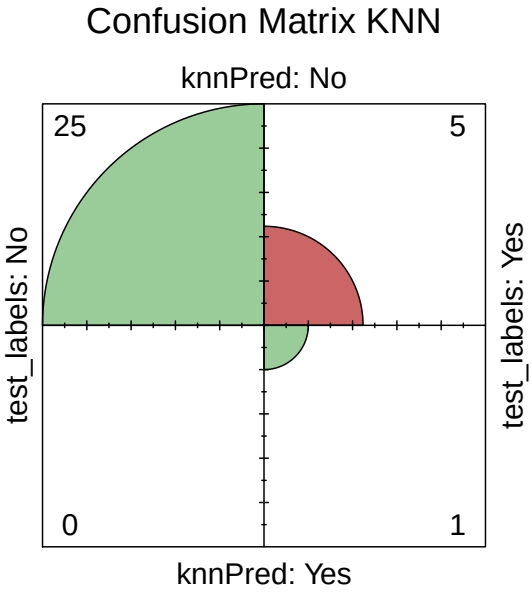


Figura 87: Matriz de confusión sobre el conjunto de test

Este comportamiento no es el habitual en aprendizaje automático, parece que casualmente el conjunto de test es bastante fácil de ajustar y por eso se obtienen mejores resultados que en training.

Vemos que no se produce ni un solo falso negativo en nuestro conjunto de test. El valor alto de K hace que se prediga con mayor facilidad este valor *No* de etiqueta y por ese desbalanceo se obtengan tan buenos resultados.

Si enfocamos el problema como una predicción no limitada al hospital y a los años de las muestras, este hecho es en sí es un poco preocupante, pues no sabemos si esta tendencia es constante en todos los pacientes. Pese a ello, debemos suponer que la muestra que tenemos es representativa y por tanto válida.

Por otro lado, como estamos tratando con un problema médico, esto quizás podría ser incluso un hecho positivo, ya que los falsos positivos sería algo que querríamos evitar a toda costa (todo depende del enfoque hacia el que se haya orientado el estudio).

Por comparar, podemos también evaluar con otros valores de K en test. Puesto que hemos obtenido los mejores resultados en training con un K de 15, que es un valor relativamente alto, podemos probar con uno bajo y uno intermedio (3 y 7).

```
k-Nearest Neighbors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 247, 248, 247, 248, 248, 247, ...
Resampling results:

  Accuracy   Kappa
  0.6654762  0.09645955

Tuning parameter 'k' was held constant at a value of 3

Test evaluation:
Accuracy     Kappa
0.8064516 0.2900763

Confusion matrix (in test split):
          test_labels
knn3Pred   No Yes
       No  23   4
       Yes  2   2
```

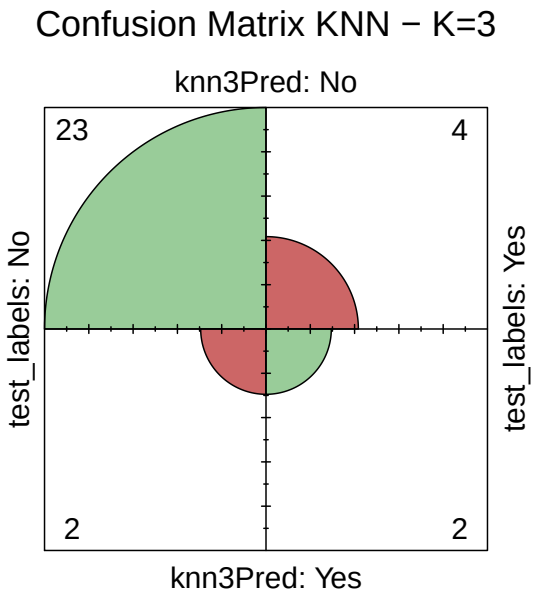


Figura 88: Matriz de confusión sobre el conjunto de test

```
k-Nearest Neighbors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 248, 247, 247, 248, 247, 247, ...
Resampling results:

  Accuracy   Kappa
  0.6874339  0.08134908

Tuning parameter 'k' was held constant at a value of 7

Test evaluation:
Accuracy     Kappa
0.8064516 0.1696429

Confusion matrix (in test split):
        test_labels
knn7Pred No Yes
     No  24   5
     Yes  1   1
```

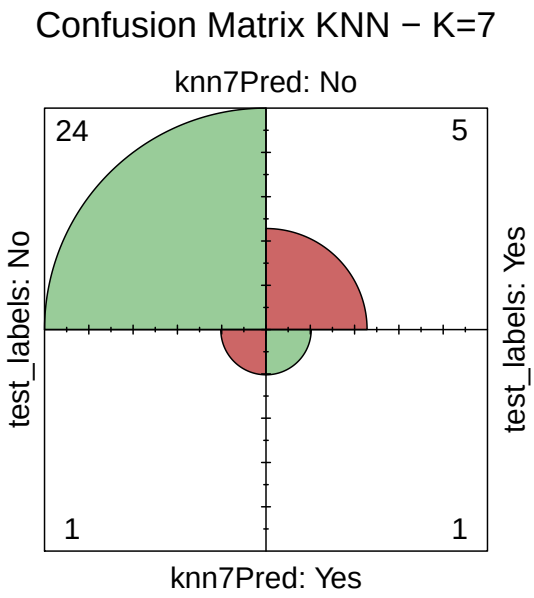


Figura 89: Matriz de confusión sobre el conjunto de test

K=7 vemos que es el que más sufre al evaluar en test, y ambos (tal y como nos había indicado la primera ejecución con CV) tienen una calidad bastante inferior (tanto en training como en test) a un K=15.

4.2. Algoritmo LDA

4.2.1. Asunciones

Comprobamos asunciones:

1. **Distribución aleatoria**: No nos queda más remedio que creer que sí.
2. **Cada predictor sigue una distribución normal**: Ya vimos en el EDA que esto no era cierto. El test de Shapiro nos aseguraba que no había normalidad en ninguna de las variables y los QQ-plots nos lo hacían ver claramente. Técnicamente sabiendo esto no deberíamos usar LDA, pero puesto que esto es un proyecto seguimos.

   Por otro lado, las variables Age y Year no parecen seguir una distribución demasiado "rara" (en comparación con una normal), por lo que es posible que al menos obtengamos resultados de calidad aceptable.
3. **Las clases siguen la misma matriz de covarianza**: Lo comprobamos a continuación.

Calculamos la diagonal de la matriz de correlación para cada una de las clases, obteniendo:

```
Para clase Yes:
      Age      Year  Positive
0.9439366 1.0656924 1.7401564

Para clase No:
      Age      Year  Positive
1.0423639 0.9998632 0.7266195
```

Estos valores nos parecen indicar que al menos la variable Positive parece tener distintas varianzas, pero es preferible asegurarlo con un test estadístico.

Puesto que nuestras variables no siguen una distribución normal, no podemos hacer el test de homogeneidad de Barlett. Utilizamos por tanto el de Levene:

`Age:`

	Df	F value	Pr(>F)
group	1	1.799898	0.1807261
	304		

`Year:`

	Df	F value	Pr(>F)
group	1	0.0624405	0.8028481
	304		

`Positive:`

	Df	F value	Pr(>F)
group	1	18.78912	1.99e-05
	304		

Indicándonos que solo se puede asegurar que la variable Positive **no** tiene homogeneidad entre clases diferentes.

Mostramos las distribuciones entre clases gráficamente:

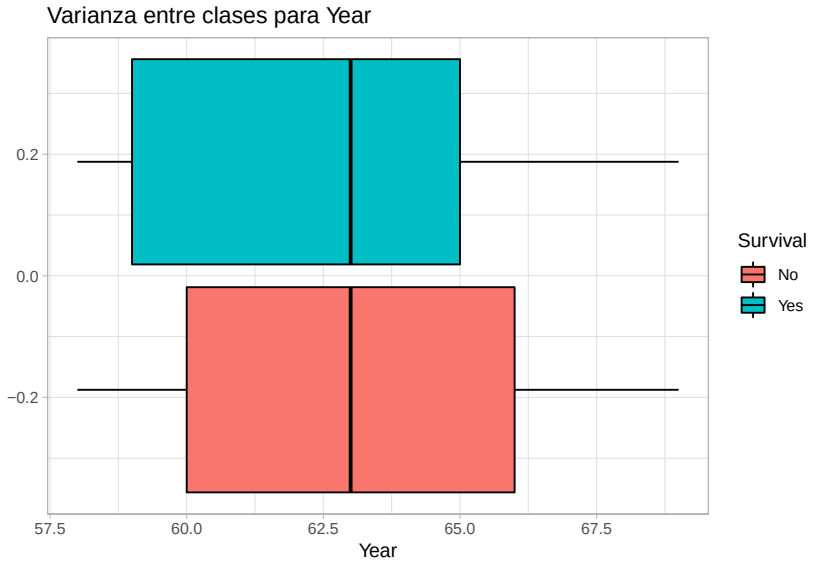


Figura 90

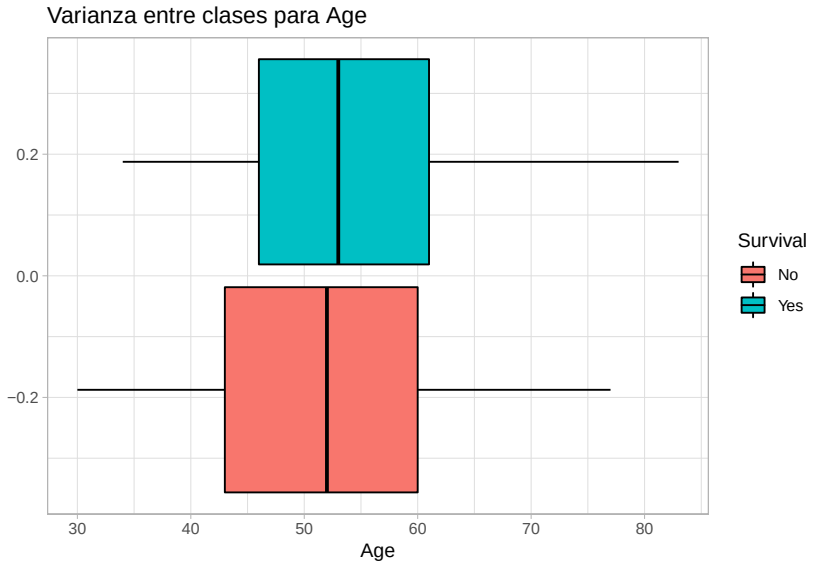


Figura 91

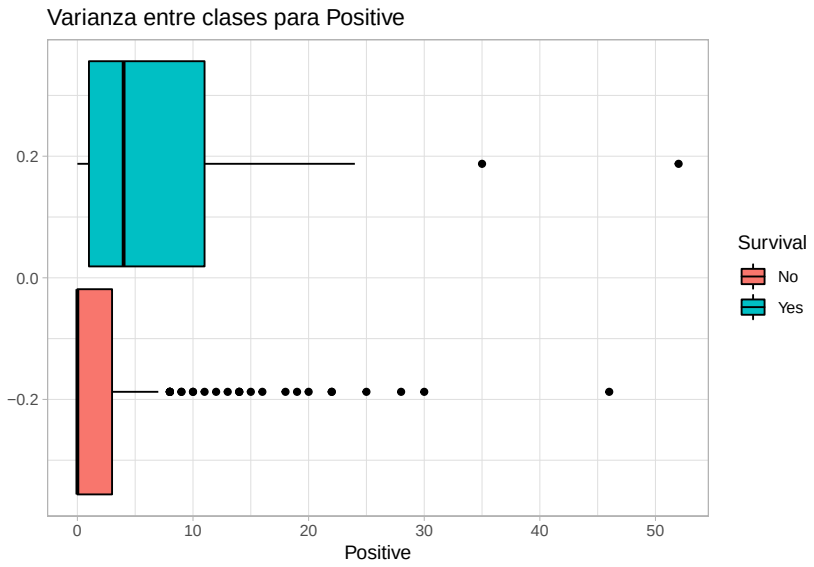


Figura 92

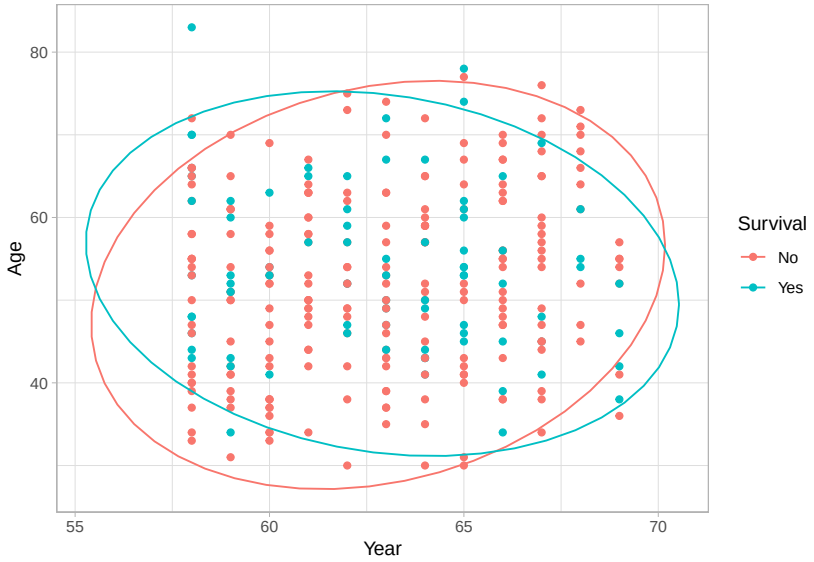


Figura 93

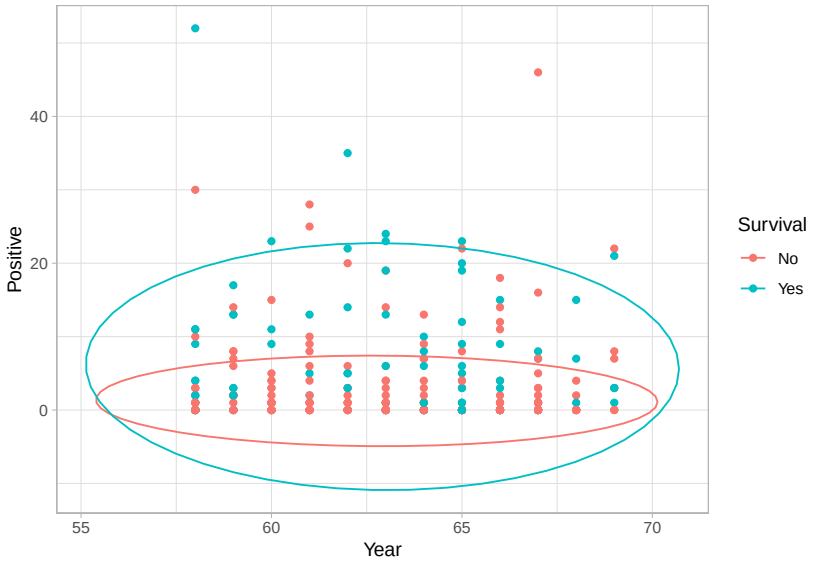


Figura 94

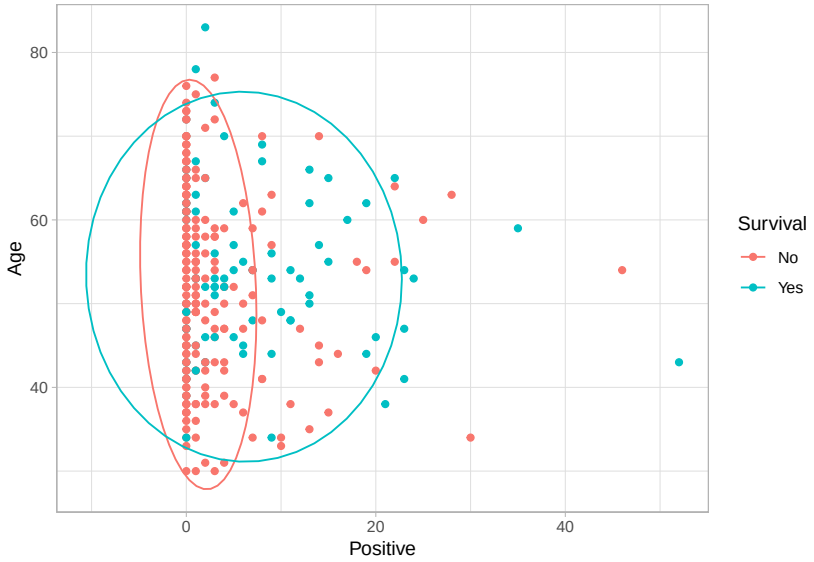


Figura 95

Se nota que la causa de que no se rechace el test para esta variable es la gran cantidad de datos con Positive igual a 0.

Por tanto para LDA no podemos hacer uso de la variable Positive, puesto que además de la falta de normalidad se incumpliría la asunción número 3, por lo que haremos uso de las otras dos.

Para terminar, aunque solo es recomendable y no son cualidades necesarias para obtener solución en LDA:

- Tenemos más instancias que predictores, por varios órdenes de magnitud.
- Los predictores son independientes.
- No tenemos varianzas cercanas a cero.

4.2.2. Aplicación del algoritmo LDA

```
Call:
lda(x, y)

Prior probabilities of groups:
       No       Yes
0.7272727 0.2727273

Group means:
            Age        Year
No  -0.05947324 -0.00398263
Yes  0.11192259 -0.03680916

Coefficients of linear discriminants:
            LD1
Age   0.9781149
Year -0.2588776
Cross-Validated (10 fold) Confusion Matrix

(entries are percentual average cell counts across resamples)

          Reference
Prediction   No  Yes
       No  72.7 27.3
       Yes  0.0  0.0

 Accuracy (average) : 0.7273
```

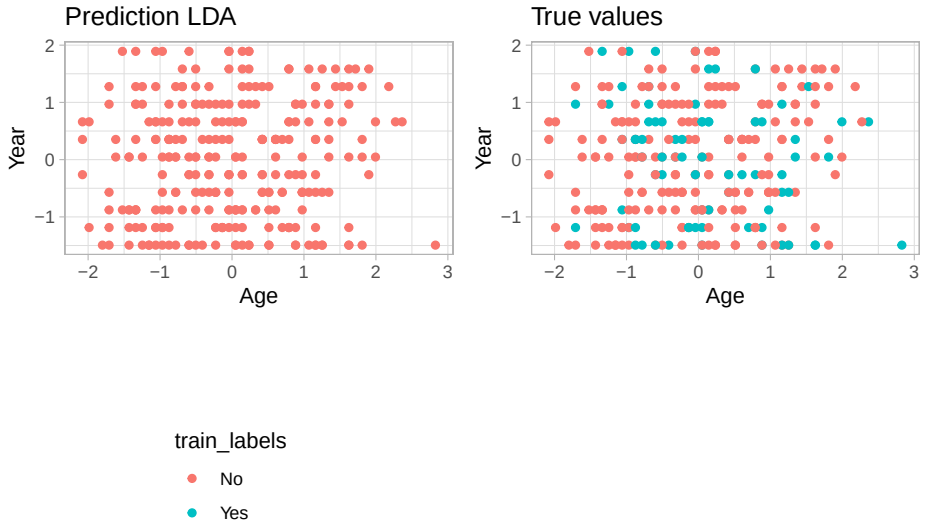


Figura 96: Predicciones LDA sobre training

El gráfico de los discriminantes no muestra una buena separación entre las clases, estando ambos centrados sobre 0.5 y similarmente esparcidos.

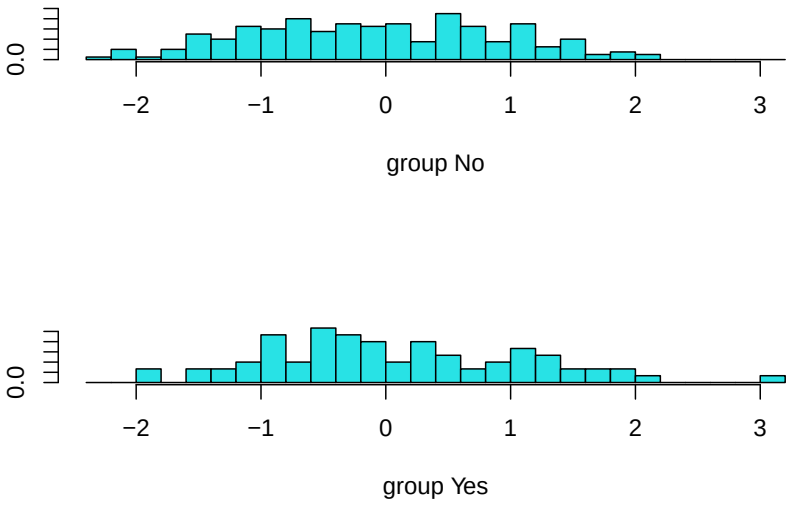


Figura 97: Histograma de los coeficientes de LDA

Y aunque es difícil verlo en 2D, se puede apreciar que el hiperplano que se genera con LDA no separa bien las clases:

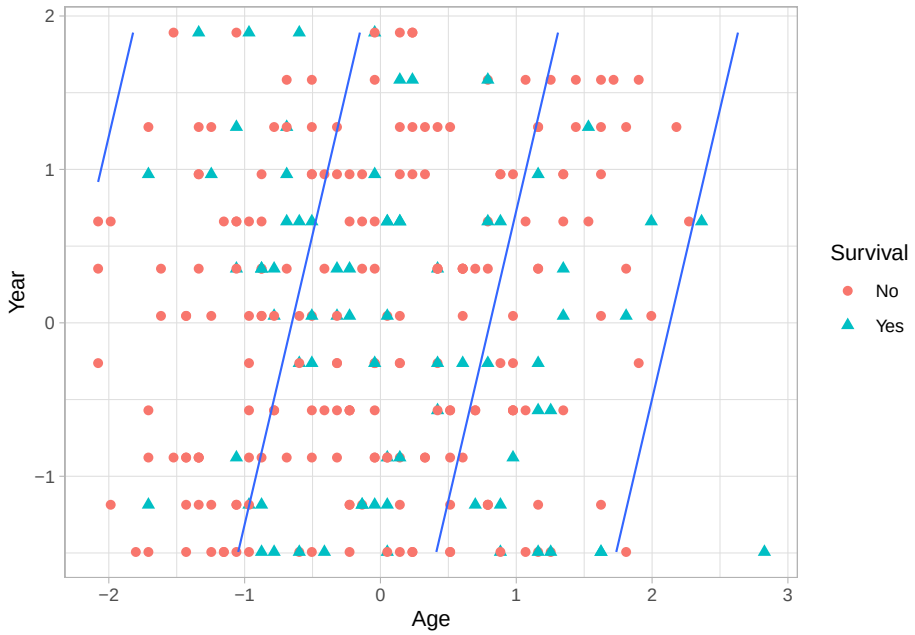


Figura 98
Estamos pintando un contorno 3D y por eso nos salen múltiples líneas en el gráfico

Resultados en test:

```
Test evaluation:
Accuracy     Kappa
0.8064516 0.0000000

Confusion matrix (in test split):
       test_labels
ldaPred No Yes
    No  25   6
    Yes  0   0


Etiquetas:
No  No  Yes No  No  No  Yes No  No  No  No  No  No  No  Yes No  No  Yes Yes
No  No  No  No  No  No  No  Yes No  No  No  No

Predicciones LDA:
No No No No No No No No No No No No No No No No No No No No No No No No No
No No No No No No
```

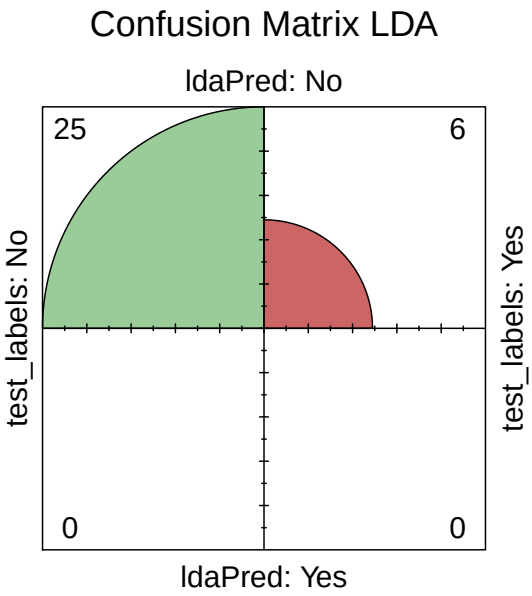


Figura 99: Matriz de confusión sobre el conjunto de test

Tenemos un dataset bastante desbalanceado, y al menos con nuestros datos LDA no predice para la clase Yes. Con esto se asegura un alto accuracy en nuestro entrenamiento, pero está claro que este comportamiento lo vuelve un mal modelo en este problema, ya que (en base únicamente a las predicciones) no parece que haya aprendizaje.

Adicionalmente los resultados en test nos devuelven un kappa igual a cero, dándonos a entender que estos son poco fiables y que se podría obtener esa misma calidad con aleatoriedad pura.

4.3. Algoritmo QDA

4.3.1. Asunciones

QDA tiene las mismas asunciones de LDA salvo que relaja la necesidad de que para una variable las clases tengan igual covarianza. Esto nos permite usar la variable Positive que habíamos descartado en LDA.

Por tanto, nos quedan los requisitos de:

1. **Distribución aleatoria**: Dábamos por hecho que sí.
2. **Distribución normal**: No la tenía ninguna variable.

Donde técnicamente el no cumplir normalidad no imposibilita que se encuentre solución, pero ya no nos lo asegura.

Y adicionalmente tenemos de forma recomendada que:

- **El número de predictores debe ser menor que el número de instancias de cada clase**: Del EDA sabemos que esto es cierto.
- **Los predictores dentro de cada clase no deben estar correlacionados**: Corroboramos que no se da en las siguientes matrices de correlación.

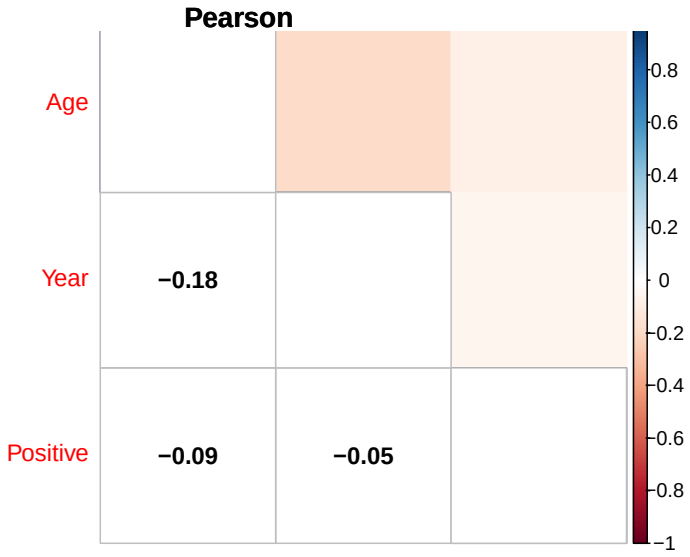


Figura 100: Matriz de correlación para la clase *Yes*

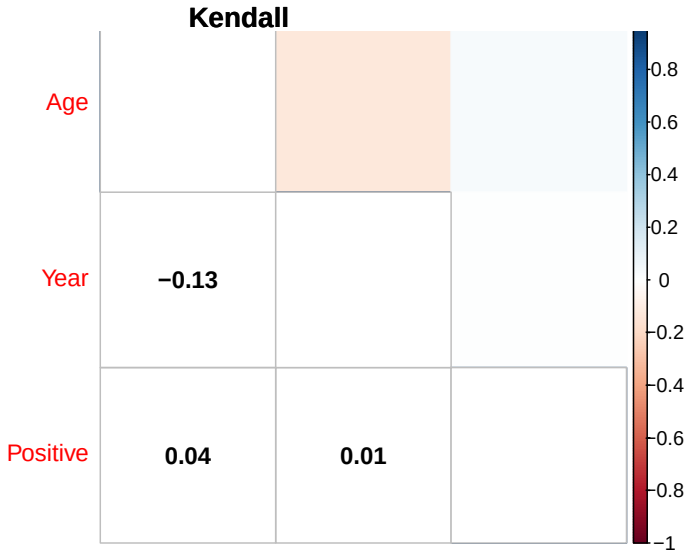


Figura 101: Matriz de correlación para la clase *Yes*

Figura 102: Matriz de correlación para la clase *No*

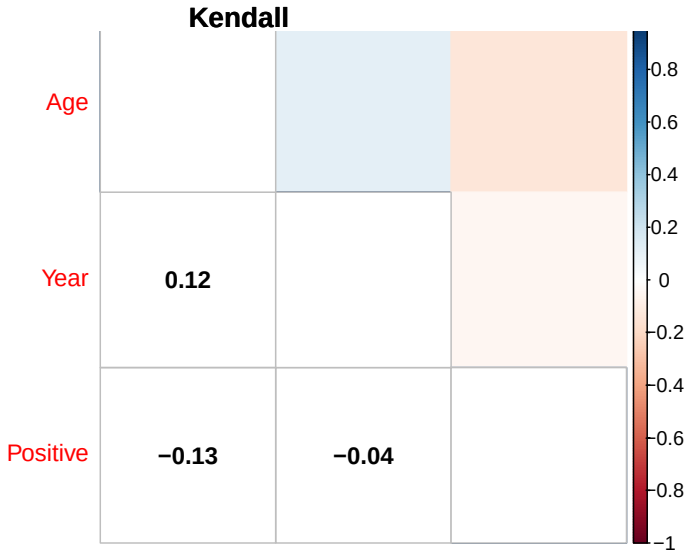


Figura 103: Matriz de correlación para la clase *Yes*

4.3.2. Aplicación del algoritmo QDA

```
Call:
qda(x, y)

Prior probabilities of groups:
       No       Yes
0.7272727 0.2727273

Group means:
            Age        Year   Positive
No  -0.05947324 -0.00398263 -0.1503750
Yes  0.11192259 -0.03680916  0.4933742
Cross-Validated (10 fold) Confusion Matrix

(entries are percentual average cell counts across resamples)

          Reference
Prediction   No  Yes
       No  68.7 21.5
       Yes  4.0  5.8

 Accuracy (average) : 0.7455
```

Figura 104: Predicciones QDA sobre training

```
Test evaluation:
Accuracy     Kappa
0.8064516 0.0000000

Confusion matrix (in test split):
       test_labels
qdaPred No Yes
    No  25   6
    Yes  0   0
```

```
Etiquetas:
No  No  Yes No  No  No  Yes No  No  No  No  No  No  No  Yes No  No  Yes Yes
No  No  No  No  No  No  No  Yes No  No  No  No

Predicciones QDA:
No No No No No No No No No No No No No No No No No No No No No No No No No
No No No No No No
```

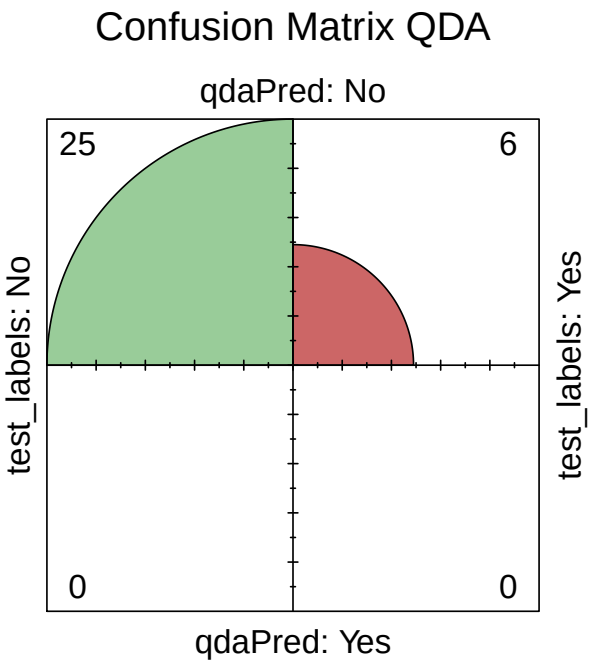


Figura 105: Matriz de confusión sobre el conjunto de test

Obtenemos los mismos resultados en test que LDA, pero en este caso vemos que sobre training sí se predice la clase *Yes*. Por tanto, podemos suponer que el hiperplano que forma no es tan restrictivo como el de LDA, pero debería tener una forma similar.

4.4. Comparativa de algoritmos

4.4.1. Para el dataset *haberman*

Si nos fijamos únicamente en los resultados obtenidos para este problema, los tres algoritmos obtienen el mismo accuracy en nuestro conjunto de test[2]. Aunque las etiquetas de este conjunto contienen elementos de ambas clases, podemos ver que se predice mayoritariamente la clase *No*. Como se había mencionado nuestro dataset está bastante desbalanceado, por lo que era más probable que se predijera esa clase con mayor facilidad.

```
Etiquetas:
No  No  Yes No  No  No  Yes No  No  No  No  No  No  No  Yes No  No  Yes Yes
No  No  No  No  No  No  No  Yes No  No  No  No

Predicciones KNN:
No  No  No  No  No  No  No  No  No  No  No  No  No  No  No  No  No  No  Yes
No  No  No  No  No  No  No  No  No  No  No  No

Predicciones LDA:
No No No No No No No No No No No No No No No No No No No No No No No No
No No No No No No

Predicciones QDA:
No No No No No No No No No No No No No No No No No No No No No No No No
No No No No No No



Test evaluation KNN:
 Accuracy     Kappa
0.8387097 0.2439024

Test evaluation LDA:
 Accuracy     Kappa
0.8064516 0.0000000

Test evaluation QDA:
 Accuracy     Kappa
0.8064516 0.0000000
```

[2]Puesto que hemos usado el paquete caret, no podemos comparar con los datos de accuracy que nos proporciona su salida. Los datos aquí mostrados hacen referencia a una evaluación de los modelos devueltos sobre el conjunto inicialmente reservado de test.

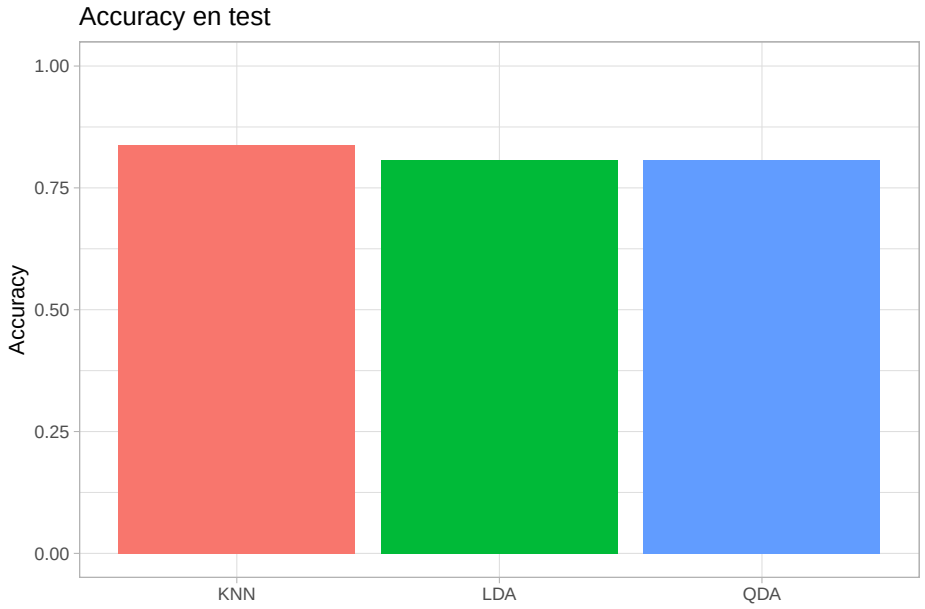


Figura 106

Hacemos notar también que los valores de Kappa son todos bajos, en menor medida para KNN.

Pese a esto, y ya no solo por tener mejores resultados, sino por no cumplir las asunciones necesarias de obtener resultados de calidad en LDA y QDA, para este problema optaríamos por usar el algoritmo KNN.

A partir de las gráficas 3D de las figuras 84 y 85 se apreció una difícil separación de las clases, por lo que el algoritmo de vecinos más cercanos nos resulta una aproximación más lógica de entre las utilizadas.

4.4.2. Comparativas generales

Para comparar la calidad genérica de los algoritmos vamos a aplicar test estadísticos en base a los resultados obtenidos en múltiples datasets.

Estas son las tablas de resultados que tenemos para test:

Dataset	out_test_knn	out_test_lda	out_test_qda
appendicitis	0.8966667	0.8690909	0.8109091
australian	0.6838235	0.8579710	0.8028986
balance	0.9024546	0.8624101	0.9167905
bupa	0.6865775	0.6837924	0.5991759
contraceptive	0.5448653	0.5091561	0.5173102
haberman	0.7462069	0.7481720	0.7512903
hayes-roth	0.5666667	0.5500000	0.5875000
heart	0.6692308	0.8481481	0.8296296
iris	0.9642857	0.9800000	0.9733333
led7digit	0.7510204	0.7420000	0.6975000
mammographic	0.7977698	0.8241269	0.8194042
monk-2	0.9743632	0.7703433	0.9235535
newthyroid	0.9071429	0.9164502	0.9629870
pima	0.7348861	0.7709930	0.7412403
tae	0.3838095	0.5245833	0.5425000
titanic	0.7850353	0.7760304	0.7733032
vehicle	0.6291452	0.7813305	0.8522409
vowel	0.6428571	0.6030303	0.9191919
wine	0.6959559	0.9944444	0.9888889
wisconsin	0.9735023	0.9592185	0.9519476

Comenzamos aplicamos el test de Wilcoxon a cada pareja de algoritmos:

LDA vs QDA: Obtenemos un ranking de 144 para LDA y 96 para QDA, con un p-value de 0.75 (o nivel de confianza del 25 %).

```
V = 96, p-value = 0.7562
alternative hypothesis: true location shift is not equal to 0

  V    V
114 - 96
```

Esto nos dice que LDA obtiene mejores resultados pero puesto que el p-value es extremadamente grande no podemos afirmar con garantía estadística que las diferencias entre los tests sean notorias.

LDA vs KNN: Ahora obtenemos un ranking de 90 para LDA y 120 para QDA, con un p-value de 0.59 (o nivel de confianza del 41 %).

```
V = 120, p-value = 0.5958
alternative hypothesis: true location shift is not equal to 0

 V     V
90 - 120
```

Seguimos teniendo un p-value demasiado grande para poder asegurar la diferencia.

QDA vs KNN: Por último tenemos un ranking de 69 para LDA y 141 para KNN, con un p-value de 0.18 (o nivel de confianza del 82 %).

```
V = 141, p-value = 0.1893
alternative hypothesis: true location shift is not equal to 0

 V    V
69 - 141
```

Aunque buscaríamos al menos un 95 % de confianza, podríamos afirmar al 82 % que los resultados de ambos algoritmos sí son significativamente diferentes.

Una comparativa múltiple de los tres algoritmos con el test de **Friedman** es la siguiente:

```
Friedman rank sum test
Friedman chi-squared = 0.7,
                  df = 2,
             p-value = 0.7047
```

El p-value es mayor que 0.05 por lo que no podemos concluir que haya al menos un par de algoritmos de calidad diferente.

Aunque el resultado del test de Friedman ya nos indica que un análisis post-hoc es innecesario, puesto que los resultados que se obtengan no van a asegurar la diferencia en la calidad de los algoritmos, por completitud en la memoria aplicamos el post-hoc de **Holm**:

```
1 = KNN, 2 = LDA, 3 = QDA

Pairwise comparisons using Wilcoxon signed rank exact test

1    2
2 1.00 -
3 0.53 1.00

P value adjustment method: holm
```

Vemos que los p-value son lo más altos posibles, por lo que carece de sentido intentar diferenciar los algoritmos. Aunque podemos notar, tal y como habíamos visto en los test de Wilcoxon, que la diferencia KNN-QDA probablemente sea mayor que el resto de parejas.

Referencias

[1] http://lib.stat.cmu.edu/datasets/cars.desc.

[2] https://www.ajdesigner.com/phphorsepower/horsepower_equation_trap_speed_method_increase_horsepower.php.

[3] http://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival.

[4] https://www.cancer.org/cancer/breast-cancer/treatment/surgery-for-breast-cancer/lymph-node-surgery-for-breast-cancer.html.

[5] https://en.wikipedia.org/wiki/Lymph_node#.

[6] https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/#.

[7] https://stats.stackexchange.com/questions/82162/cohens-kappa-in-plain-english.

[8] http://thatdatatho.com/2018/02/19/assumption-checking-lda-vs-qda-r-tutorial-2/.