Manuel Carmona Cabello de Alba
manuel.carmona@cunef.edu

Técnicas de Agrupación y de Reducción de la Dimensión. Master en Data Science para Finanzas

Práctica 2: ¿Qué caracteriza a los todoterrenos?

Índice:

0. Resumen ejecutivo....2
1. Objetivos....3
2. Metodología empleada....3
2.1. Análisis exploratorio datos....3
2.1.1. Variables numéricas....3
2.1.2. Variables nominales....6
2.2. Imputación de valores a los NA....7
2.2.1. Análisis de los NA....7
2.2.2. Imputación de valores....9
2.2.2.1. Primer método: MICE....9
2.2.2.2. Segundo método: Missforest....9
2.2.2.3. Elección del método de imputación....10
2.3. Pertinencia de realizar el análisis factorial....10
2.3.1. Análisis de la matriz de correlaciones....10
2.3.2 Determinante de la matriz de correlaciones....12
2.3.3. Test de esfericidad de Barlett....12
2.3.4. KMO Global....12
2.3.5. KMO Parcial....12
2.4. Resultados quitando las variables plazas y rpm....14
2.4.1. Diferencias imputación....14
2.4.2. Matriz de correlación....14
2.4.3. KMO....15
2.4.4. MSA....15
2.5. Análisis factorial....16
2.5.1. Método componentes principales (PCA)....16
2.5.1.1. Contribución CCPP en las explicaciones de las variables....16
2.5.1.2. Contribución de las variables a los CCPP....18
2.5.2.3 Matriz de componentes no rotados....20
2.6. Rotaciones factoriales....21
2.6.1. Rotación varimax....21
2.6.2. Rotación oblimin....23
2.7 Método del factor principal....24
3. Análisis cluster....25
3.1. ¿Tiene sentido realizar el análisis cluster?....25
3.1.1. Evaluación de la bondad del análisis cluster....27
3.2. Identificación del número de grupos....27
3.3. Aplicación del algoritmo K-means....28
3.4. Interpretación de los grupos....29
4. Conclusiones....32
5. Bibliografía....33
6. Anexo: código....34

Manuel Carmona Cabello de Alba
manuel.carmona@cunef.edu

0. Resumen ejecutivo

El presente informe conforma un proyecto de *data science* en el que se lleva a cabo un estudio de las características fundamentales de los todoterrenos a partir de una base de datos compuesta por algunos de los todoterrenos a la venta en España desde hace unos años, con el objetivo de explicarlas con el menor número de variables posibles a través del análisis factorial.

Para tal propósito, en primer lugar, se lleva a cabo un análisis exploratorio de datos para conocer y describir las variables iniciales. Tras el estudio de los datos, se procede al tratamiento de los valores perdidos y a la elección de un método de imputación de valores.

Una vez con los datos listos para trabajar, se ha estudiado la pertinencia de llevar a cabo el análisis factorial. Tras un resultado positivo en las pruebas realizadas, incluyendo el determinante de la matriz, test de esfericidad de Barlett, KMO y MSA, se ha confirmado la posibilidad de proceder con el análisis factorial. Sin embargo, antes de proceder, se ha comprobado que al excluir las variables rpm y plazas, cuyo MSA resultaba demasiado bajo, los resultados de todas las pruebas han mejorado. Así, se ha decidido continuar excluyendo estas variables.

Se ha realizado un análisis factorial por los métodos del factor común y componentes principales. Siguiendo el criterio de parsimonia y la regla de Kaiser, la solución elegida en un primer momento es la propuesta por el método de componentes principales, con dos factores que explican el 78.5% de la varianza. Sin embargo, dada su difícil interpretación, se ha estudiado una solución rotada por el método Varimax que, ofreciendo el mismo porcentaje de varianza explicada, ha permitido distinguir 2 factores con mayor facilidad de interpretación. Uno de los factores explica mayormente las propiedades más intrínsecas al rendimiento del motor, como son la potencia, la velocidad y la aceleración, mientras que el otro factor hace mas hincapié en otras características como el peso, el precio y el consumo.

Adicionalmente, se ha llevado a cabo un análisis clúster para estudiar la asociación entre los distintos todoterrenos en función de sus características, a fin de obtener información añadida sobre el tema que aquí se trata. Se ha realizado a partir de la solución de 2 componentes principales no rotados, puesto que en este caso la prioridad es la explicación de la varianza antes que la interpretación de las características. Esta agrupación ha permitido distinguir dos segmentos de todoterrenos, unos mas potentes y con un precio medio mas alto, y otro segmento de coches menos potentes y precio medio más bajo.

Finalmente, se ha justificado por qué se han alcanzado los objetivos planteados en el inicio del informe.

En los anexos, el lector podrá encontrar el código empleado en *R* que avala los resultados emitidos por el presente informe.

Manuel Carmona Cabello de Alba
manuel.carmona@cunef.edu

1. Objetivos

El objetivo principal de este informe es conocer las características fundamentales de los todoterrenos a partir de una base de datos de todo terrenos a la venta en España hace unos años. De acuerdo con el principio de parsimonia intentaremos explicar las características de este tipo de vehículos con el menor número posible de variables explicativas. Para ello realizaremos un análisis factorial en caso de sea posible realizar este tipo de análisis.

Como objetivo secundario pretendemos agrupar a los todos terrenos en grupos con características homogéneas o similares mediante las técnicas de análisis clúster estudiadas.

2. Metodología empleada

2.1. Análisis exploratorio de datos.

Tras proceder a importar los datos, observamos la composición del dataset, con el objetivo de ver que datos tenemos, de que tipo son y comprobar la existencia de NA.

Procedemos a visualizar la estructura del Dataset.

El dataset cuenta con 15 variables, 12 numéricas y 3 de tipo nominal.

2.1.1. Variables numéricas:

- **PVP euro**: Precio en euros del coche

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9113   18140   24867   26696   31169   69461
```

La mitad de los todos terrenos tienen un precio igual o inferior a 25.000 euros, solo un 25% de los todoterrenos tienen un precio superior a los 31.200 euros con un máximo en 69.461 euros.

- **Número de Cilindros**

A continuación, mostramos una tabla con la distribución de la variable número de cilindros

```
##   x freq relativa frecuencia acumulada
## 1 4   91         0.728               0.728
## 2 6   31         0.248               0.976
## 3 8    3         0.024               1.000
```

La variable cilindros solo toma 3 valores en el dataset (4, 6 y 8 cilindros). Un 72,8% de los todoterrenos tienen sólo 4 cilindros, un 25% tiene 6 cilindros y solo un 2% de los mismos tienen 8 cilindros.

- **cc**: Cilindrada en cm cúbicos

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1298    2184    2497    2570    2835    5216
```

La mitad de los todos terrenos tienen menos de 2.500 centímetros cúbicos, sólo un 25% de los todoterrenos tienen más de 2850 centímetros cúbicos con un máximo en 5216 centímetros cúbicos.

- **Potencia**: medida en número de caballos (cv)

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    64.0    95.0   112.0   117.1   125.0   225.0
```

La mitad de los todoterrenos tienen 112 caballos, y solo un 25% tienen más de 125, con un máximo de 225 caballos. Por tanto, podemos ver como la dispersión en cuarto cuartil es muy grande con respecto al resto de cuartiles (la diferencia entre el primer cuartil y la mediana es 27 caballos frente a los 100 caballos de dispersión entre el tercer cuartil y el máximo).

- **RPM**: Revoluciones por minuto

A continuación, mostramos una tabla y un histograma con la distribución en intervalos de las revoluciones por minuto (en miles).

```
##    rpm_cut Freq frecuencia_relativa frecuencia_acumulada
##    [3.6,4)    9               0.072                0.072
##    [4,4.4)   43               0.344                0.416
##  [4.4,4.8)   27               0.216                0.632
##  [4.8,5.2)    3               0.024                0.656
##  [5.2,5.6)   26               0.208                0.864
##    [5.6,6)    9               0.072                0.936
##    [6,6.4)    6               0.048                0.984
##  [6.4,6.8)    2               0.016                1.000
```

De acuerdo con la tabla un 63,2% de los todos terrenos tienen entre 3.6 y 4.8 mil revoluciones por minuto, estado un 55% del total entre 4 y 4.8.

Hasta 5.6 miles de revoluciones hay un 86,4% del total de todos terrenos, estando un 20,8% de las observaciones en el intervalo de 5.2 a 5.6 miles de revoluciones. El 13,4% restante se encuentra en los intervalos de 5.6 a 6.8 miles de revoluciones.

- **Peso**: Peso en kg

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     930    1462    1750    1675    1909    2320       2
```

La mitad de los todos terrenos tienen un peso igual o inferior a 1.750 kilos, teniendo solo un 25% de los todoterrenos más de 1.909 kilos con un máximo de 2320 kilos.

- **Plazas**: Número de plazas

La distribución del número de plazas es la siguiente:

```
##   x freq freq relativa frecuencia_acumulada
## 1 2    6         0.048                0.048
## 2 4   27         0.216                0.264
## 3 5   61         0.488                0.752
## 4 6    2         0.016                0.768
## 5 7   23         0.184                0.952
## 6 8    4         0.032                0.984
## 7 9    2         0.016                1.000
```

Cerca de la mitad de los todoterrenos del dataset (un 48,8%) tienen 5 plazas, habiendo solo un 25% del total con más de plazas.

De los todoterrenos que tienen más de 5 plazas, la mayoría tienen 7 plazas, habiendo solo varios de 6,8 y 9 plazas. En cuanto a los todoterrenos que tienen menos de 5 plazas, la mayoría tienen 4 plazas (un 21,6% del total).

- **Cons90**: Consumo 90 km/h

A continuacion mostramos una tabla y un histograma con la distribucion en intervalos del consumo a 90 km por hora.

```
##  consumo90_cut Freq frecuencia_relativa frecuencia_acumulada
##      [6.6,8.1)   42          0.36521739            0.3652174
##      [8.1,9.6)   39          0.33913043            0.7043478
##     [9.6,11.1)   26          0.22608696            0.9304348
##    [11.1,12.6)    5          0.04347826            0.9739130
##    [12.6,14.1)    3          0.02608696            1.0000000
```

De acuerdo con el dataset a una velocidad de 90 km/h, 7 de cada 10 todoterreno tiene un nivel consumo de bajo a medio y 2 de consumo alto y 1 de consumo muy alto.

- **Cons120**: Consumo 120 km/h

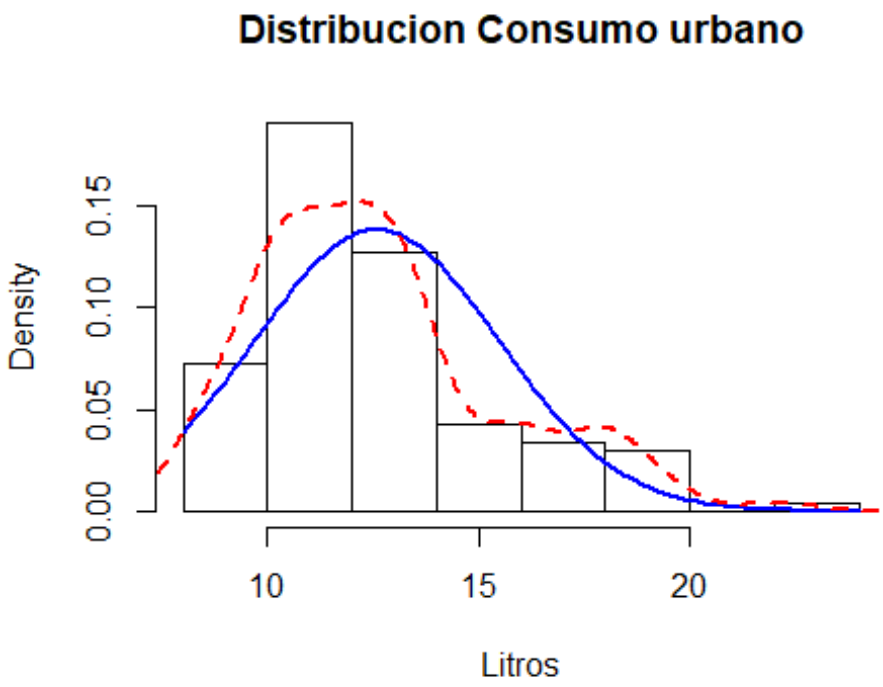
A continuacion mostramos una tabla y un histograma con la distribucion en intervalos del consumo a 90 km por hora.

```
##  consumo120_cut Freq frecuencia_relativa frecuencia_acumulada
##      [8.4,10.4)   22          0.20370370            0.2037037
##     [10.4,12.4)   39          0.36111111            0.5648148
##     [12.4,14.4)   29          0.26851852            0.8333333
##     [14.4,16.4)   14          0.12962963            0.9629630
##     [16.4,18.4)    4          0.03703704            1.0000000
```

De acuerdo con el dataset a una velocidad de 120 km/h, aproximadamente 6 de cada 10 de los todos terrenos tiene un nivel consumo de entre 8.4 y 12.4 litros, 3 un consumo de entre 12.4 y 15 litros y 1 con un consumo superior a 15 litros.

- **Consurb**: Consumo urbano

A continuacion mostramos un histograma con los datos de consumo urbano de los todos terrenos:



De acuerdo con el histograma, la gran mayoría de los todoterrenos tienen un consumo urbano inferior a 15 litros, siendo el porcentaje de todoterrenos con consumos superiores a los 15 litros muy bajo. La mayor acumulación de coches se encuentra en el intervalo de entre 10 y 14 litros.

- **Velocidad**: Velocidad máxima

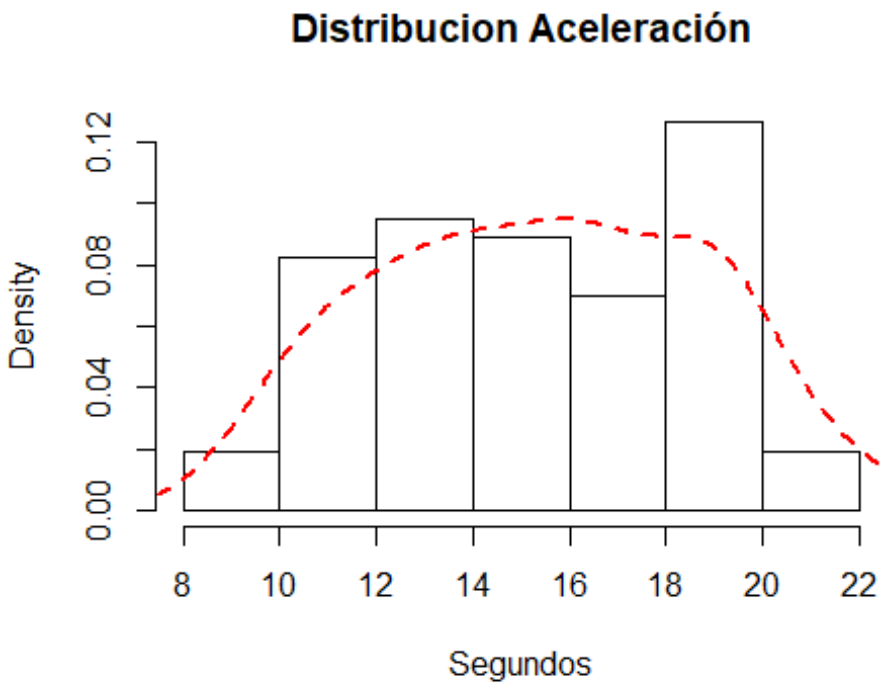
A continuacion mostramos un histograma con los datos de consumo urbano de los todoterrenos:

```
##  velocidad_cut Freq frecuencia_relativa frecuencia_acumulada
##      [120,130)    9         0.073770492           0.07377049
##      [130,140)   19         0.155737705           0.22950820
##      [140,150)   38         0.311475410           0.54098361
##      [150,160)   15         0.122950820           0.66393443
##      [160,170)   20         0.163934426           0.82786885
##      [170,180)   13         0.106557377           0.93442623
##      [180,190)    7         0.057377049           0.99180328
##      [190,200)    1         0.008196721           1.00000000
```

- 2/3 de los todos terrenos tienen una velocidad máxima inferior a 160 km/h, estando aproximadamente un 1/3 del total en el intervalo de 140 a 150 km por hora.
- Del 1/3 restante, la mayoría de los todos terrenos se encuentran en el intervalo de 160 a 180 km hora. Solo un 6% tienen velocidades superiores a los 180 km por hora.

- **Aceleración**: Tiempo de aceleración de 0 a 10

A continuacion mostramos un histograma con los tiempos de aceleración de los todoterrenos:



Como se puede apreciar la distribución de esta variable es bastante uniforme. Salvo para los casos más extremos (tiempos inferiores a 10 segundos o superiores a 20 segundos).

2.1.2. Variables nominales:

- **Marca**: son las marcas de los distintos todoterrenos.

Mostramos en una tabla las marcas con mas y menos coches del Dataset:

```
##           V1 V2
## 1     NISSAN 19
## 2     SUZUKI 19
## 3 LAND ROVER 15
## 4 MITSUBISHI 15
```

```
## 5       JEEP 10
## 6       OPEL  9

##             V1 V2
## 12 ASIA MOTORS  3
## 13         KIA  2
## 14        LADA  2
## 15        TATA  2
## 16   CHEVROLET  1
## 17    DAIHATSU  1
```

Las marcas con más todoterrenos son: Nissan, Suzuki, Landrover, Mitsubishi y Jeep. Las marcas con menos: Daihatsu, Chevrolet, Tata, Lada y Kia.

- **Modelo**: son los distintos modelos de las marcas.

En primer lugar, analizamos si hay modelos duplicados.

Hay 13 modelos que aparecen más de dos veces en el dataset. 12 que aparecen 2 veces y uno que aparece 3.

Hemos calculado las diferencias de precio para los modelos repetidos como la diferencia entre el precio máximo y el precio mínimo y hemos comprobado como para todos los modelos hay diferencias de precio. En la mitad de modelos también hay diferencias en cuanto al número de asientos, es decir, son un mismo tipo de coche, pero con la opción de tener 2 plazas más. En otros casos se puede observar un aumento de peso del vehículo, con el respectivo aumento del consumo.

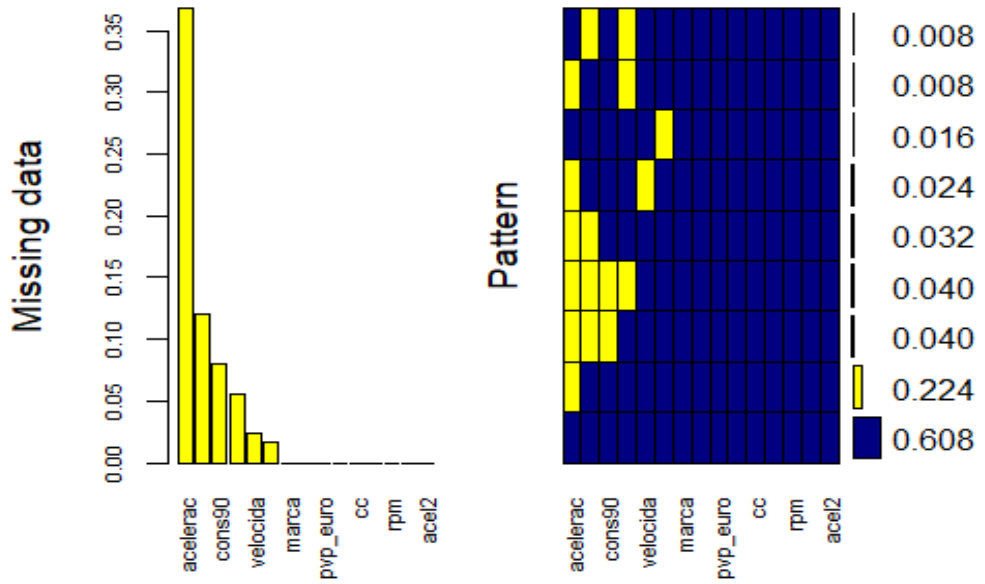
Las diferencias de precio entre coches de un mismo modelo, con las mismas características de potencia, cilindrada, y numero de cilindros se debe fundamentalmente a los extras que suelen incorporar los coches y que no están incluidos en la base de datos, como son: tapicería de cuero, equipos de sonido, faros, aparcamiento asistido, arranque sin llave, entre otros.

2.2 Imputación de valores a los NAs

2.2.1. Análisis de los NA

En este apartado vamos a analizar la distribucion de los NAs en el Dataset para posteriormente eliminar las observaciones o imputar valores.

Visualizamos la distribucion de los NA en las variables



Análisis del grafico missing Data:

La variable acelerac, tiene 46 NA, un 36% sobre el total de las observaciones. Existe otra variable que aporta información sobre la aceleración, pero realmente no aporta mucha información puesto que es una variable categórica que solo distingue entre aquellos todoterrenos que aceleran de 0 a 100 en menos de 10 segundos de aquellos que no, siendo solo 3 los que aceleran de 0 a 100 en menos de 10 segundos. Así, no voy a considerar la opción de eliminar la variable acelerac puesto que podría perderse información valiosa.

Del resto de variables las que más datos perdidos tienen son consumo 120, consumo90 y consumo urbano con entre un 12 y un 5% de missing values. Las variables velocidad y peso tienen un porcentaje de Na muy pequeño.

En términos globales del dataset:

- En un 60,80% de los casos no hay NAs.
- En un 22,40% de los casos solo hay NAs en la variable aceleración
- En un 4% de los casos hay NAS para las variables aceleración, consumo 120, consumo 90 y también 4% para estas variables y el consumo urbano, o lo que es lo mismo, un 8% de las observaciones tienen 3 o más NAs. Estos son los casos más problemáticos pues habría que estimar 3 o 4 valores distintos que imputar a partir de los valores del resto de variables.
- Un 3,2% de las observaciones con 2 NAs (aceleración y consumo 120)
- Un 2,4% de las observaciones con 2 NAs (aceleración y consumo urbano)
- Un 1,6% de las observaciones con NAs en la variable peso

Procedemos a eliminar del Dataset los todos terrenos con 3 y 4 valores en NA, sería un total de 10 observaciones (un 8% del total), ya que los distintos algoritmos de imputación de valores que vamos a utilizar tendrían que hacer muchas suposiciones lo que aumentaría el error de imputación significativamente.

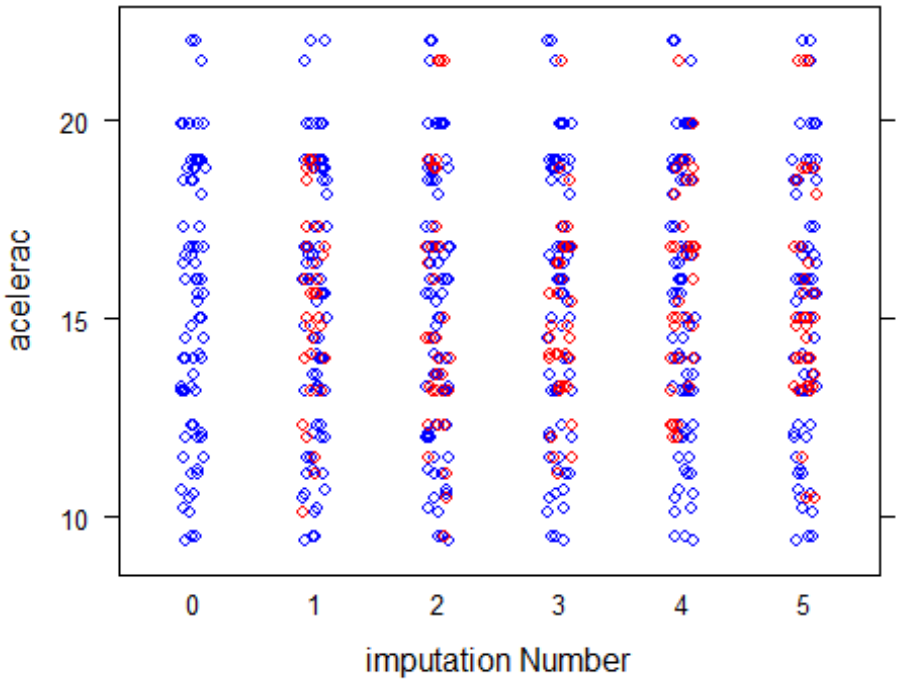
2.2.2. Imputación de valores

Vamos a imputar valores a los NAs de las variables aceleración, consumo urbano, velocidad y peso mediante dos métodos distintos. Em ambos métodos procederemos de la siguiente manera:

Dejaremos fuera las variables categóricas: marca, modelo, acel2(tiempo de aceleración).

2.2.2.1. Primer método: MICE

MICE, *Multivariate Imputation by chained equations*, asume que los datos faltantes son Desaparecidos al azar (MAR - missing at random), lo que significa que la probabilidad de que falte un valor depende solo del valor observado y se puede predecir. Por defecto usa la regresión lineal para predecir valores perdidos continuos y la logística se usa para valores categóricos faltantes. Una vez que se completa este ciclo, se generan múltiples conjuntos de datos (*Tutorial on 5 Powerful R Packages used for imputing missing values, 2016*). Estos conjuntos de datos difieren solo en los valores perdidos imputados y los juntaremos realizando el promedio.



Este grafico nos muestra la imputación de valores para variable aceleración que es la más NA tiene en los distintos dataset que genera la función mice. Los valores observados están en azul y los imputados en rojo, como se puede apreciar gráficamente la distribucion de la imputación es muy similar.

Por ello la imputación de valores mediante esta función será igual a la media de los resultados de los distintos datasets.

2.2.2.2. Segundo método: Missforest

MissForest es una implementación de algoritmo random forest. Es un método de imputación no paramétrico aplicable a varios tipos de variables. El método no paramétrico no hace suposiciones explícitas sobre la forma funcional de f (cualquier función arbitraria). En cambio, trata de estimar f de modo que pueda estar lo más cerca posible de los puntos de datos sin parecer poco práctico. Así, el algoritmo construye un modelo random forest para cada variable. Luego utiliza el modelo para predecir

valores perdidos en la variable con la ayuda de los valores observados (*Tutorial on 5 Powerful R Packages used for imputing missing values, 2016*).

NRMSE - *normalized mean squared error* - es el error cuadrático medio normalizado. Nos sugiere que el error de imputación es de un 0,37%.

2.2.2.3. Elección del método de imputación

Para elegir el método más preciso vamos a analizar la diferencia entre la matriz de correlaciones de las observaciones iniciales completas con la obtenida por ambos métodos:

- Diferencias de la imputación mediante paquete mice función PMM

Calculamos el % de variación entre la matriz inicial y la obtenida por el método mice.

```
## [1] 0.142
```

- Diferencias imputación mediante paquete missForest mediante random forest

Calculamos el % de variación entre la matriz inicial y la obtenida por el método missForest.

```
## [1] 0.138
```

Como se puede comprobar la diferencia de la matriz de correlaciones mediante el random forest es de un 13.8% frente al 14.2% del pmm. Elegiremos la imputación de valores realizada por el método de random forest puesto que demuestra ser más preciso.

2.3. Pertinencia de realizar el análisis factorial.

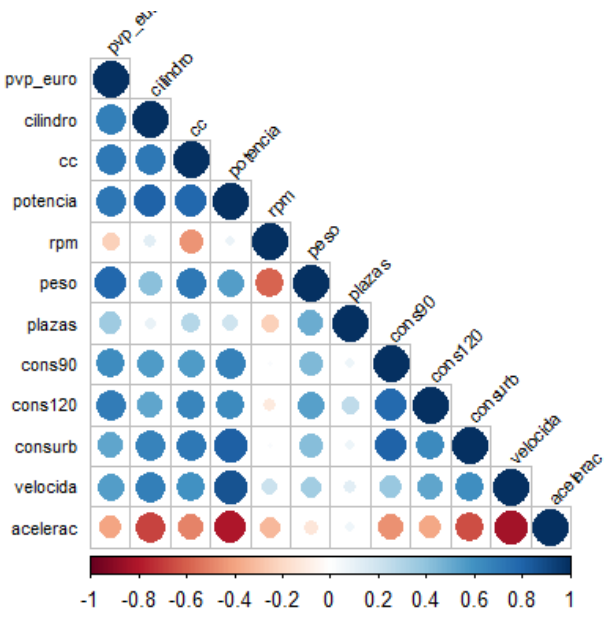
Para analizar la pertinencia del análisis debemos examinar que se cumplan los criterios de colinealidad y multicolinealidad entre las variables que se introducirán al modelo.

Para ello vamos a examinar la matriz de correlaciones y a calcular distintos indicadores (test esfericidad de barlett, kmo global, kmo parcial)

2.3.1. Análisis de la matriz de correlaciones

Se determinarán las posibles asociaciones o interrelaciones existentes entre todas las variables estudiadas, a partir de la matriz de observaciones. Cuando se haya determinado tal matriz conviene estudiarla de cara a comprobar si sus características son adecuadas o no para llevar a cabo el análisis. En el caso que nos ocupa, sería importante que las variables analizadas están fuertemente asociadas.

Podemos visualizar a la matriz de correlaciones mediante un correlograma:

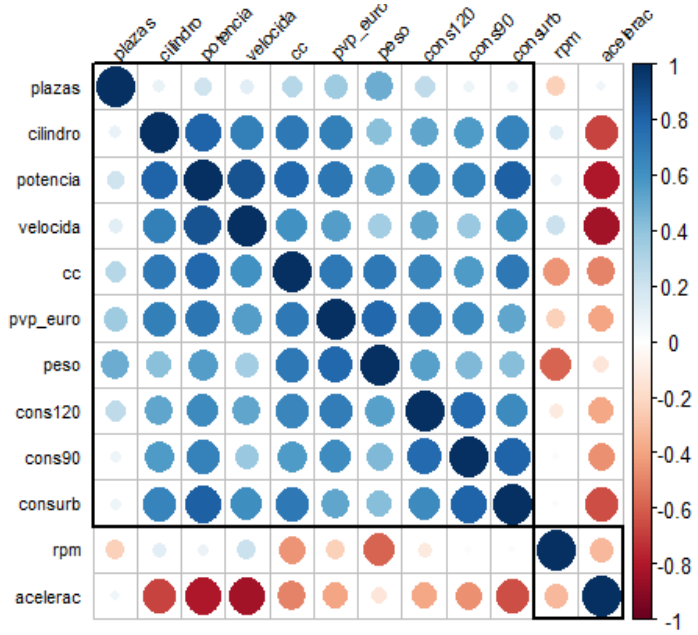


Observamos las correlaciones entre las distintas variables, así como los niveles de significación. Observamos que existe correlación entre todas las variables. Como buscamos colinealidad, esperamos que los valores fuera de la diagonal sean mayores a 0.3 (baja colinealidad), más tendientes a 0.5 (colinealidad media) y de forma ideal igual o mayor a 0.7(alto grado de colinealidad).

En el caso de esta matriz, podemos observar claramente la existencia de colinealidad en todos los casos, siendo la norma un grado de colinealidad medio-alto.

Observamos que la variable aceleración se comporta de forma distinta al resto de variables, por lo que si agrupamos variables en dos grupos quedaría aislada por su comportamiento frente al resto.

Podemos hacer un clúster de variables, es decir agrupar variables en función de su correlación.



Los resultados del clúster son dos grupos: el primer grupo formado por las variables aceleración y RPM y en el segundo grupo el resto de variables.

La división se debe a que todas las variables tienen una relación lineal positiva, excepto la variable aceleración que tiene una relación lineal negativa con el resto de variables y rpm cuya asociación con el resto de variables es bastante baja.

2.3.2. Determinante de la matriz de correlaciones

Calculamos el determinante de la matriz de correlaciones para comprobar si existe multicolinealidad:

```
## [1] 4.80422e-07
```

El determinante es cercano a cero, lo que indica un alto grado de multicolinealidad entre las variables involucradas en la matriz, confirmando nuestras observaciones de la matriz de correlaciones.

2.3.3. Test de esfericidad de Barlett

El test de esfericidad de Bartlett busca contrastar la hipótesis nula de que la matriz de correlaciones es igual a una matriz de identidad. Lo que nos interesa para efectos de buscar multicolinealidad, por lo tanto, es rechazar la hipótesis nula, y aceptar la hipótesis alternativa de que la matriz es distinta a una matriz de identidad, y por ende hay un nivel suficiente de multicolinealidad entre las variables. Este procedimiento es particularmente útil cuando el tamaño muestral es pequeño.

Aplicamos el test de esfericidad de Barlett y obtenemos el siguiente resultado:

```
## $chisq
## [1] 1588.222
##
## $p.value
## [1] 3.279328e-288
##
## $df
## [1] 66
```

Siendo una distribución chi2 con 45 grados de libertad y H0=las variables son independientes, con un pvalor de 6.038611e-249 se rechaza H0 y, por tanto, se asume la multicolinealidad de las variables.

2.3.4. KMO global

El índice KMO compara la magnitud de los coeficientes de correlación observados con la magnitud de los coeficientes de correlación parcial. Este estadístico varía entre 0 y 1, pudiendo clasificar la calidad del índice de la siguiente manera: - KMO>0.9: muy bueno - 0.9>KMO>0.8: bueno - 0.8>KMO>0.7: aceptable - 0.7>KMO>0.6: mediocre - 0.6>KMO: malo. Para valores por debajo de 0.5 hay que considerar cambiar de variables o de técnica, ya que es muy poco probable que los modelos funcionen si esta prueba no se cumple.

```
## [1] 0.75
```

Devuelve un valor aceptable, de 0.75. Es un indicador de buen nivel de multicolinealidad entre las variables.

2.3.5. KMO Parcial

Ahora calculamos el KMO parcial o MSA para cada una de las variables con el objetivo de matizar y corroborar el resultado de la prueba del KMO, identificando las variables que no son susceptibles de ser reducidas y aquellas que si lo son.

```
##             pvp_euro     cilindro          cc    potencia         rpm
## pvp_euro  1.00000000  0.419703878 -0.10264536  0.07028610 -0.09188485
## cilindro  0.41970388  1.000000000  0.59862487 -0.06841597  0.53506814
## cc       -0.10264536  0.598624865  1.00000000  0.47503945 -0.82202105
## potencia  0.07028610 -0.068415968  0.47503945  1.00000000  0.42480966
## rpm      -0.09188485  0.535068143 -0.82202105  0.42480966  1.00000000
## peso      0.44928576  0.003430242 -0.23176951  0.38272257 -0.52177587
## plazas    0.04955362 -0.185701858  0.15661156  0.04870735  0.24608430
## cons90    0.32555992 -0.051512524 -0.16330658  0.36986337  0.05335118
## cons120   0.08262768 -0.161855867  0.31504955 -0.39124213  0.12741092
## consurb  -0.38912131 -0.028177874  0.30042255  0.03126883  0.11807025
## velocida  0.18966485 -0.120347631 -0.01149963  0.48741725  0.14743884
## acelerac  0.08821661 -0.112280799  0.06007924 -0.20475532  0.10355446
##                   peso      plazas      cons90     cons120     consurb
## pvp_euro  0.449285760  0.04955362  0.32555992  0.08262768 -0.38912131
## cilindro  0.003430242 -0.18570186 -0.05151252 -0.16185587 -0.02817787
## cc       -0.231769507  0.15661156 -0.16330658  0.31504955  0.30042255
## potencia  0.382722571  0.04870735  0.36986337 -0.39124213  0.03126883
## rpm      -0.521775871  0.24608430  0.05335118  0.12741092  0.11807025
## peso      1.000000000  0.31417190 -0.06967903  0.03592802  0.13820150
## plazas    0.314171904  1.00000000 -0.18786545  0.13602841 -0.02703232
## cons90   -0.069679031 -0.18786545  1.00000000  0.68955739  0.59743608
## cons120   0.035928020  0.13602841  0.68955739  1.00000000 -0.22013839
## consurb   0.138201500 -0.02703232  0.59743608 -0.22013839  1.00000000
## velocida -0.013439672 -0.11466852 -0.69724612  0.56781719  0.20130615
## acelerac  0.249527471  0.06078398 -0.22769402  0.22788187  0.01138992
##             velocida    acelerac
## pvp_euro  0.18966485  0.08821661
## cilindro -0.12034763 -0.11228080
## cc       -0.01149963  0.06007924
## potencia  0.48741725 -0.20475532
## rpm       0.14743884  0.10355446
## peso     -0.01343967  0.24952747
## plazas   -0.11466852  0.06078398
## cons90   -0.69724612 -0.22769402
## cons120   0.56781719  0.22788187
## consurb   0.20130615  0.01138992
## velocida  1.00000000 -0.47804991
## acelerac -0.47804991  1.00000000
```

La diferencia entre la correlación global y la correlación parcial es la influencia del resto de variables. Si la correlación es de 0.84 entre 2 plazos, si pudiese excluir el resto de observaciones, la correlación global seria 0.84 con lo cual el resto de variables influyen poco.

Los coeficientes de correlación parcial deben tener un valor bajo para que las variables compartan factores comunes.

Los elementos de la diagonal de esta matriz son similares al estadístico KMO para cada par de variables e interesa que están cercanos a 1. Esto se cumple en el ejemplo.

Obtenemos el MSA:

```
## [1] "pvp_euro = 0.800616747745888"
## [1] "cilindro = 0.819281724210901"
## [1] "cc = 0.676306822661564"
## [1] "potencia = 0.804883427175625"
## [1] "rpm = 0.450655802860536"
## [1] "peso = 0.591386071493205"
## [1] "plazas = 0.431818535320323"
## [1] "cons90 = 0.639446851679356"
## [1] "cons120 = 0.671474437640282"
```

```
## [1] "consurb = 0.82762601726438"
## [1] "velocida = 0.709892950754928"
## [1] "acelerac = 0.87828623249243"
```

Observamos que los MSA son aceptables en general, excepto para las variables rpm y plazas cuyos valores son inferiores a 0,50 por tanto vamos a proceder a eliminar estas variables de nuestro análisis.

2.4 Resultados quitando las variables plazas y rpm

Debido a los resultados del MSA vamos a quitar las variables plazas y rpm de nuestro análisis, por ello volvemos a realizar la imputación de valores NA sin tener en cuenta estas variables (debido a su baja correlación con el resto de variables).

Veremos si disminuyen las diferencias de imputación al quitar estas variables.

2.4.1. Diferencias imputación

Error mice

```
## [1] 0.095
```

Error rf

```
## [1] 0.084
```

Comparamos las diferencias de imputación: con las variables plazas y rpm (situación anterior) y sin estas variables.

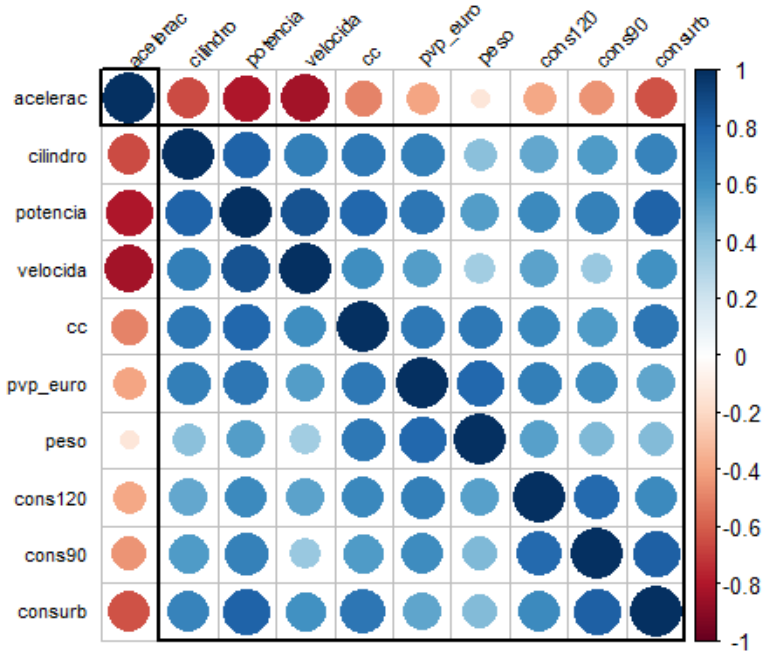
```
##             error_mice2
## error_mice1 0.095
##       0.142     1

##          error_rf2
## error_rf1 0.084
##     0.138     1
```

Como se puede comprobar disminuye las diferencias entre la matriz de correlaciones original (sin NAs) cuando quitamos las variables plazas y rpm para los dos algoritmos (mice y random forest), siendo también el algoritmo de random forest el más preciso y por eso lo utilizaremos.

2.4.2. Matriz de correlación

Representamos de nuevo la matriz de correlaciones entre las variables con el clúster de dos grupos

```
corrplot(cor.mat3, type="full", order="hclust", addrect = 2,
         tl.col="black", tl.cex=0.7, tl.srt=45)
```



Como se puede apreciar en este caso la variable aceleración es la única que tiene relación inversa respecto del resto de variables y por eso se encuentra sola en otro grupo, antes estaba también la variable rpm. Del mismo modo las correlaciones entre las variables ahora son más altas, excepto para la variable peso que se puede apreciar a simple vista que en líneas generales la relación con respecto al resto de variables no es muy fuerte.

2.4.3. KMO

Volvemos a calcular el KMO

```
##       kmo2
## kmo1   0.78
##   0.75    1
```

Como se puede comprobar el KMO ha mejorado al quitar las variables Plazas y RPM del análisis. Lo cual es un indicador de la bondad de la decisión de quitar estas variables.

2.4.4. MSA

Calculamos las diferencias del MSA al quitar las variables plazas y rpm

```
##    Variable Diferencias MSA
## 1  acelerac     -0.01848901
## 2        cc      0.13413659
## 3  cilindro      0.06423565
## 4   cons120     -0.01553232
## 5    cons90     -0.03254410
## 6   consurb     -0.02558765
## 7      peso     -0.04920511
## 9  potencia      0.03176708
## 10 pvp_euro     -0.03823738
## 12 velocida     -0.05195238

## [1] -0.001408629
```

El MSA mejora para las variables: CC, Cilindro y potencia. Sin embargo, empeora para el resto de variables. Aunque como se puede apreciar al sumar las diferencias de todas las variables, la diferencia es muy pequeña.

2.5. Análisis factorial

El análisis factorial (ANFAC) es un método de análisis multivariante que intenta explicar, según un modelo lineal, un conjunto extenso de variables observables mediante un número reducido de variables hipotéticas llamadas factores. Un aspecto esencial del ANFAC es que los factores no sean directamente observables, obedeciendo a conceptos de naturaleza más abstracta que las variables originales (*López Zafra, 2017*).

Un factor no tiene la información que tienen las variables. Tenemos que ponerle un nombre a los factores, que no dejan de ser la esencia de la relación entre variables.

Métodos de extracción de factores. Hay 2:

A) Uno es extracción de con ponentes principales a partir de la matriz de correlaciones. A partir de ahí, se reduce la dimensión. Componentes principales se extraen tantos como variables (en este caso 10). Nos quedamos solo con aquellos componentes principales que sean realmente representativos de la relación entre las variables, que serán los factores.

B) La otra opción es la extracción del factor principal. Es el más "puro". Decimos que es puro ya que el primer método (componentes principales) se olvida del factor único. Pervierte el esquema del análisis factorial. Todas las variables dependen de los factores a la vez que todos los factores dependen de las variables.

2.5.1. Método de las componentes principales (PCA)

En este apartado procederemos a analizar el modelo factorial prescindiendo de las unicidades para expresar las variables empleando exclusivamente factores comunes.

Para ello, determinaremos la primera componente principal o factor F1 de forma que explique la mayor parte de las variables. Una vez se obtenga este, se le resta a las variables y sobre la variabilidad restante se determina la segunda componente principal o F2, siguiendo el mismo criterio.

Mientras que en un análisis clúster agrupamos observaciones, en este caso estamos agrupando variables para ver sus asociaciones.

2.5.1.1. Contribución CCPP en la explicación de las variables

Prestamos atención a los autovalores (*eigenvalue*) y al % de varianza explicada:

```
##         eigenvalue percentage of variance
## comp 1        6.54                  65.44
## comp 2        1.31                  13.09
## comp 3        0.80                   8.01
## comp 4        0.42                   4.18
## comp 5        0.36                   3.63
## comp 6        0.26                   2.57
## comp 7        0.12                   1.22
## comp 8        0.08                   0.84
## comp 9        0.07                   0.70
## comp 10       0.03                   0.31
##         cumulative percentage of variance
## comp 1                              65.44
## comp 2                              78.53
## comp 3                              86.55
## comp 4                              90.73
```

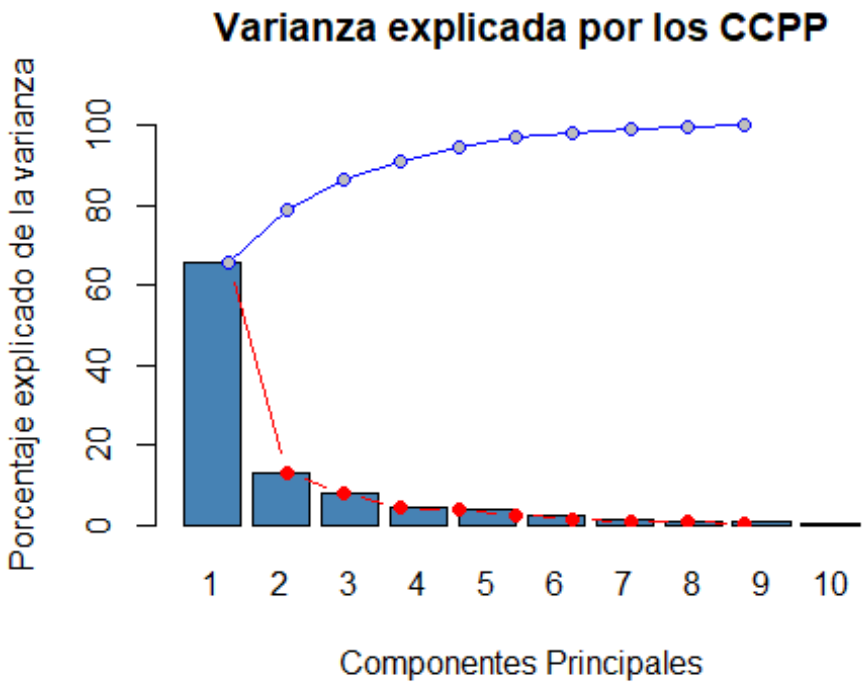
```
## comp 5                                94.36
## comp 6                                96.93
## comp 7                                98.15
## comp 8                                98.99
## comp 9                                99.69
## comp 10                              100.00
```

Los autovalores nos dan una medida de tolerancia para poder decidir con cuanta cantidad de componentes o factores es recomendable quedarnos. Para elegir el número de componentes a emplear podemos utilizar la regla de Kaiser. Los autovalores iguales o mayores a 1 indican que el componente logra explicar más varianza que una variable por sí sola, así la regla de Kaiser establece que solo se escogerán aquellos componentes con autovalores superiores a 1 (*Daróczi, 2015*).

Prestando atención a los autovalores, únicamente los dos primeros componentes tienen un autovalor superior a 1. En este sentido y en línea con el grafico anterior, nos quedaremos con las 2 dimensiones representadas por su capacidad explicativa.

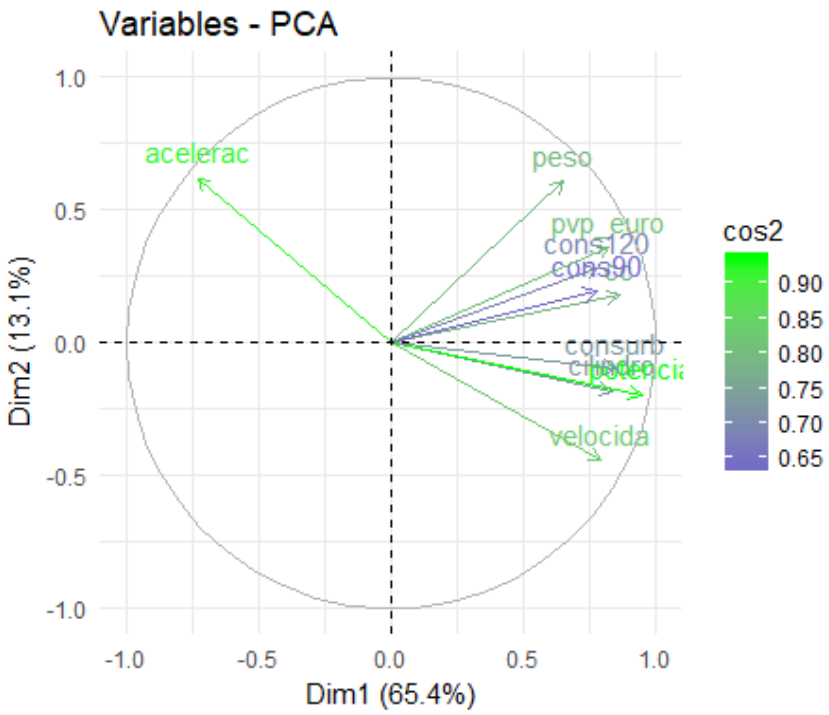
La primera dimensión o primer componente explica el 65.46%, mientras que la segunda explica el 13.08%. Tenemos una explicación conjunta del 78.54% de la varianza.

Veamos de forma gráfica el poder explicativo de los componentes principales:



Se observa una gran diferencia entre el primer componente y el resto. La línea azul representa el % de varianza explicada acumulada, mientras que la línea roja marca la varianza explicada por cada componente. Los cambios de pendiente ayudan a observar la capacidad explicativa que va aportando cada componente a medida que se van incorporando al modelo, quedando patente que a partir del tercer componente la contribución marginal es muy pequeña.

El siguiente grafico muestra la contribución de las componentes principales 1 y 2 en la explicación de cada variable (su comunalidad), en escala de colores:



Cuanto más verde sea el vector de la variable, más explicada queda por los componentes principales 1 y 2. Podemos observar como apenas se ha perdido información de las variables potencia y aceleración, siendo las variables referentes al consumo (cons120 y cons90) aquellas sobre las que se pierde más información.

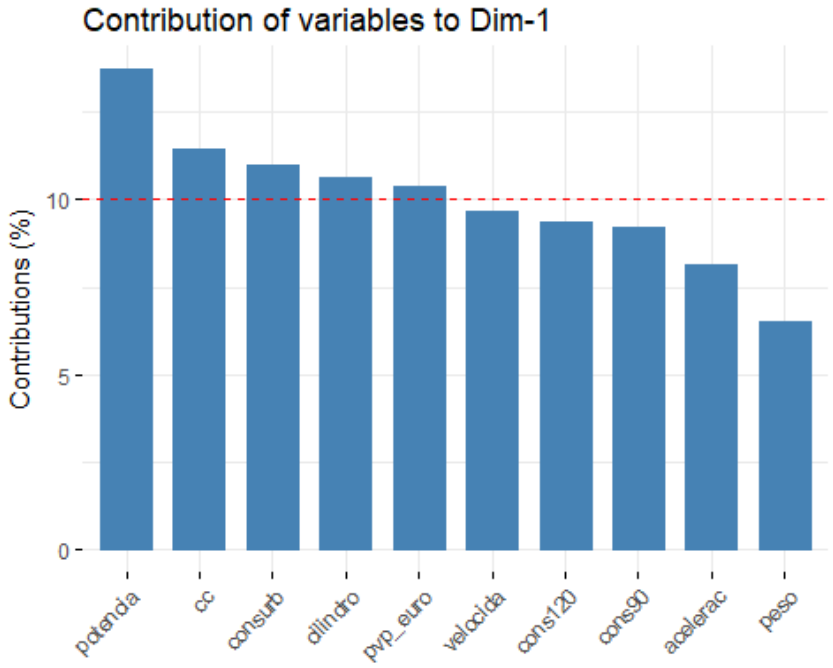
Adicionalmente, la longitud del vector también nos indica el grado de explicación de las variables por los dos componentes. Se observa como las variables con mayor longitud del vector, también tienen un color más verde. En este sentido, conforme más lejos de la circunferencia se encuentran los vectores, menor es la explicación que están dando los 2 componentes principales de dicha variable y, por tanto, son necesarios más de 2 CCPP.

Observamos que las variables cilindro y potencia se comportan prácticamente igual. Cons90 y cc también se comportan de forma muy parecida. Por otro lado, se observa que las variables precio y cc se comportan de forma muy similar. Las variables que más distan en su comportamiento son velocidad y aceleración, pudiendo establecerse la siguiente relación: a mayor aceleración menor velocidad, y viceversa. Esto tiene sentido puesto que según se configuren los desarrollos y la relación de la caja de cambios, más inclinación a la velocidad punta o a la aceleración tendrá un vehículo (*Tipos de desarrollos y de relación en caja de cambios, 2018*). Así, un todoterreno más *Adventure* tendrá más aceleración que uno pensado primordialmente para un uso habitual en carretera y un uso esporádico en caminos.

2.5.1.2. Contribución de las variables a los CCPP

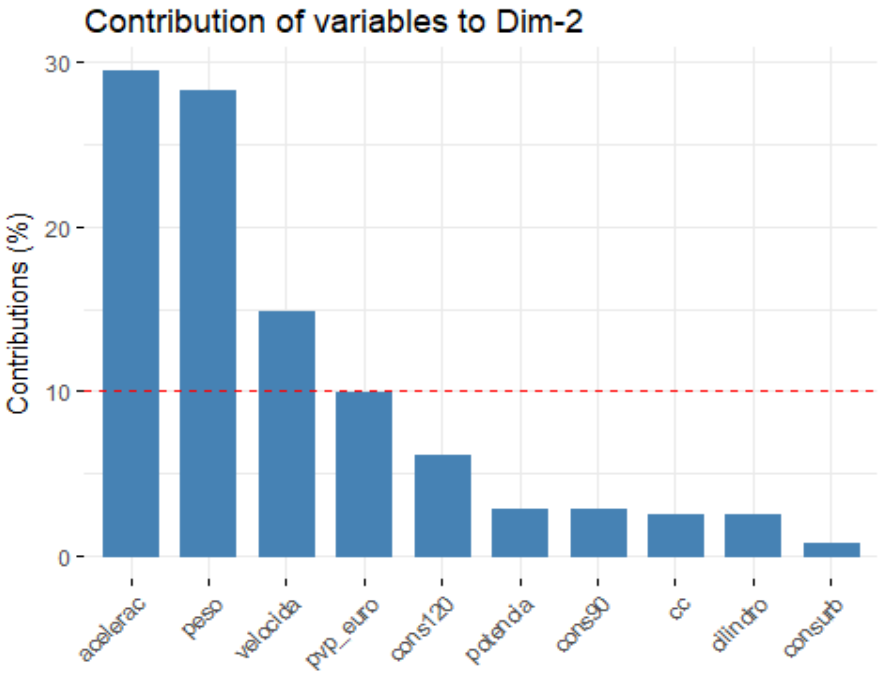
También es interesante conocer la contribución de las variables a los CCPP. A mayor valor de la medida (expresado como % de la relación por cociente entre la comunalidad de la variable respecto del autovalor del CP) mayor contribución de la variable.

El siguiente gráfico muestra la contribución de las variables a la dimensión 1:



Vemos como las variables que más contribuyen a la primera componente principal son precio, cilindro, cc, potencia y consumo urbano, aunque las diferencias con el resto de variables no son muy grandes.

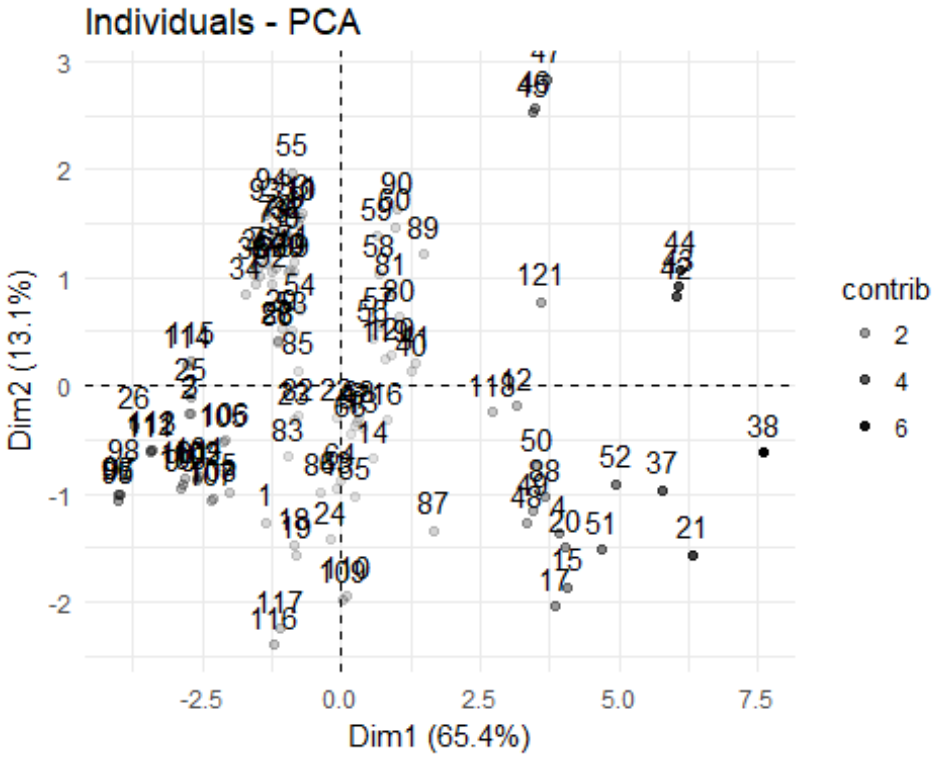
El siguiente gráfico muestra la contribución de las variables a la dimensión 2:



En la segunda componente si se aprecian diferencias más significativas, siendo protagonistas aceleración y peso, seguidas de velocidad y precio. El resto de variables contribuyen conjuntamente en un 17.36%

En ambos gráficos, la línea roja representa la uniformidad en la representación: si cada variable tuviese un poder explicativo uniforme todas quedarían a esa altura. Así, se comprueba que la dimensión 1 está más equilibrada en cuanto a composición, mientras que para la dimensión 2 las variables aceleración y peso son esenciales por su enorme contribución, siendo también muy importante la variable velocidad.

De la misma manera podemos representar las observaciones con su contribución a la explicación de los



PCA:

Aquellos con mayor contribución a las componentes principales 1 y 2 tienen el punto en color negro, mientras que los que tienen una contribución más baja a estos componentes tienen un punto más claro.

2.5.1.3. Matriz de componentes no rotados

Sabiendo que nuestra intención es reducir la dimensión a dos componentes principales, generamos la matriz de componentes no rotados para observar la carga factorial de los ítems en los componentes con los que nos vamos a quedar:

```
## Principal Components Analysis
## Call: principal(r = completedata_rf, nfactors = 2, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##              PC1   PC2   h2    u2 com
## pvp_euro    0.82  0.36 0.81 0.192 1.4
## cilindro    0.84 -0.18 0.73 0.270 1.1
## cc          0.86  0.18 0.78 0.219 1.1
## potencia    0.95 -0.19 0.94 0.063 1.1
## peso        0.65  0.61 0.80 0.205 2.0
## cons90      0.78  0.19 0.64 0.360 1.1
## cons120     0.78  0.28 0.69 0.310 1.3
## consurb     0.85 -0.10 0.73 0.271 1.0
## velocida    0.80 -0.44 0.83 0.174 1.6
## acelerac -0.73  0.62 0.92 0.083 2.0
##
##                        PC1  PC2
```

```
## SS loadings           6.54 1.31
## Proportion Var        0.65 0.13
## Cumulative Var        0.65 0.79
## Proportion Explained  0.83 0.17
## Cumulative Proportion 0.83 1.00
##
## Mean item complexity =  1.4
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  57.85  with prob <  0.00032
##
## Fit based upon off diagonal values = 0.99
```

Como podemos observar, las cargas factoriales correspondientes a cada variable para cualqueira de los dos factores. Esto dificulta la identificación de a qué factor tiende a asociarse cada variable. Por esta razón procederemos a la rotación factorial.

2.6. Rotaciones factoriales

La matriz de saturaciones factoriales o matriz factorial indica la relacion entre los factores y las variables. Sin embargo, del resultado que finalmente obtenemos puede ser dificil extraer una interpreacion sencilla de los factores. La rotacion factorial pretende seleccionar la solucion mas sencilla e interpretable siguiendo el criterio de parsimonia. El objetivo es girar los ejes de coordenadas, que representan a los factores, de manera que se aproximen al maximo a las variables en que estan saturados.

Existen dos sistemas para rotar la matriz: - Rotaciones ortogonales: conservan la ortogonalidad inicial de los ejes, lo que supone que los factores seguiran incorrelados dos a dos. -Rotaciones oblicuas: no conservan la ortogonalidad de los ejes.

2.6.1. Rotacion Varimax

Conociendo que queremos reducir la dimensión a 2, realizaremos una solucion rotada mediante el procedimiento de varianza maxima (*Varimax*) con el objetivo de explicar con 2 factores las 10 variables iniciales. Varimax es un ajuste de rotación (rotacion ortogonal) de los componentes, que minimiza el número de variables que tienen saturaciones altas en cada factor. Con este método se simplifica la interpretación de los factores. (*IBM Knowledge Center, n.d.*)

Para aplicar la rotacion Varimax normalizaremos los datos y aplicaremos componentes principales con rotacion Varimax, obteniendo los siguientes resultados:

```
TT.rotP = principal(tterreno.norm, nfactors=2, rotate ="none")
TT.rotP
```

```
## Principal Components Analysis
## Call: principal(r = tterreno.norm, nfactors = 2, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##             PC1   PC2   h2    u2 com
## pvp_euro   0.82  0.36 0.81 0.192 1.4
## cilindro   0.84 -0.18 0.73 0.270 1.1
## cc         0.86  0.18 0.78 0.219 1.1
## potencia   0.95 -0.19 0.94 0.063 1.1
## peso       0.65  0.61 0.80 0.205 2.0
## cons90     0.78  0.19 0.64 0.360 1.1
## cons120    0.78  0.28 0.69 0.310 1.3
## consurb    0.85 -0.10 0.73 0.271 1.0
## velocida   0.80 -0.44 0.83 0.174 1.6
## acelerac -0.73  0.62 0.92 0.083 2.0
##
```

```
##                         PC1  PC2
## SS loadings           6.54 1.31
## Proportion Var        0.65 0.13
## Cumulative Var        0.65 0.79
## Proportion Explained  0.83 0.17
## Cumulative Proportion 0.83 1.00
##
## Mean item complexity =  1.4
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  57.85  with prob <  0.00032
##
## Fit based upon off diagonal values = 0.99
```

Realizamos la rotacion Varimax y obtenemos los siguientes resultados:

```
## Principal Components Analysis
## Call: principal(r = tterreno.norm, nfactors = 2, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##            RC1   RC2   h2    u2 com
## pvp_euro  0.84  0.32 0.81 0.192 1.3
## cilindro  0.47  0.72 0.73 0.270 1.7
## cc        0.74  0.48 0.78 0.219 1.7
## potencia  0.54  0.81 0.94 0.063 1.7
## peso      0.89  0.03 0.80 0.205 1.0
## cons90    0.69  0.41 0.64 0.360 1.6
## cons120   0.75  0.35 0.69 0.310 1.4
## consurb   0.53  0.67 0.73 0.271 1.9
## velocida  0.25  0.87 0.83 0.174 1.2
## acelerac -0.08 -0.95 0.92 0.083 1.0
##
##                         RC1  RC2
## SS loadings           3.95 3.91
## Proportion Var        0.39 0.39
## Cumulative Var        0.39 0.79
## Proportion Explained  0.50 0.50
## Cumulative Proportion 0.50 1.00
##
## Mean item complexity =  1.5
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  57.85  with prob <  0.00032
##
## Fit based upon off diagonal values = 0.99
```

Podemos observar como la rotacion da lugar a una explicacion idéntica de la varianza. Se observa también que los factores explican una cantidad de varianza similar, a diferencia de lo que ocurría en la solucion no rotada.

Hemos obtenido un primer factor asociado especialmente al precio, al peso y al consumo en menor medida. Por otro lado, el segundo factor explica las características asociadas al rendimiento del motor, destacando en carga factorial la aceleración, la velocidad, la potencia y, en menor medida, el numero de cilindros. En este sentido, por resumir podriamos denominar "*Valores de rendimiento del motor*" (sin ser exactamente así) al segundo factor, puesto que se centra en las macros cuantitativas de motor dejando un poco de lado el consumo, mientras que el primer factor podría denominarse "*precio, peso y consumo*" (también sin ser exacto).

Suma de las cargas factoriales al cuadrado es igual al "autovalor rotado"

2.6.2. Rotacion Oblimin

Existen otros procedimientos de rotación, como por ejemplo "Oblimin", que es un tipo de rotación oblicua, que se utiliza cuando se considera que los componentes/factores a extraer no son completamente independientes entre sí, debido a que se entienden como pertenecientes a un mismo concepto latente general. Vamos a probar los resultados que obtendriamos en caso de rotar la matriz con este metodo:

```
## Loading required package: GPArotation

## Principal Components Analysis
## Call: principal(r = tterreno.norm, nfactors = 2, rotate = "oblimin")
##
##  Warning: A Heywood case was detected.
## Standardized loadings (pattern matrix) based upon correlation matrix
##           TC1   TC2   h2    u2 com
## pvp_euro 0.84  0.11 0.81 0.192 1.0
## cilindro 0.32  0.65 0.73 0.270 1.4
## cc       0.69  0.31 0.78 0.219 1.4
## potencia 0.37  0.73 0.94 0.063 1.5
## peso     0.98 -0.23 0.80 0.205 1.1
## cons90   0.65  0.25 0.64 0.360 1.3
## cons120  0.74  0.16 0.69 0.310 1.1
## consurb  0.41  0.58 0.73 0.271 1.8
## velocida 0.04  0.89 0.83 0.174 1.0
## acelerac 0.18 -1.03 0.92 0.083 1.1
##
##                        TC1  TC2
## SS loadings           4.01 3.85
## Proportion Var        0.40 0.38
## Cumulative Var        0.40 0.79
## Proportion Explained  0.51 0.49
## Cumulative Proportion 0.51 1.00
##
##  With component correlations of
##      TC1  TC2
## TC1 1.00 0.49
## TC2 0.49 1.00
##
## Mean item complexity =  1.3
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  57.85  with prob <  0.00032
##
## Fit based upon off diagonal values = 0.99
```

A modo de observación, se ha obtenido un *Heywood Case*. Dado que las comunidades son correlaciones al cuadrado, se esperaría que siempre estén entre 0 y 1. Sin embargo, una peculiaridad matemática del modelo de factor común es que las estimaciones finales de comunalidad podrían exceder 1. Si una comunidad es igual a 1, se hace referencia a la situación como un caso Heywood, y si una comunalidad excede 1, es un caso ultra-Heywood. Un caso ultra-Heywood implica que algún factor único tiene una varianza negativa, una clara indicación de que algo está mal. Los expertos no están de acuerdo sobre si una solución de factores con un caso de Heywood puede considerarse legítima o no (*SAS/STAT(R) 9.2 User's Guide, 2018*).

Al margen de esta observacion, se obtiene una explicacion de la varianza idéntica al resto de soluciones. Sin embargo, teniendo en cuenta que se ha obtenido un Heywood Case y que los factores de la solución rotada por el método varimax resultan más fácilmente interpretables, se descarta esta solución.

2.7 Método del factor principal

En este apartado procedemos a analizar el modelo factorial lineal completo, con factores comunes y unicos. Es decir, en este modelo se tendrá en cuenta tanto la varianza de la variable explicada por los factores comunes o comunalidad (columna h2), como la varianza de la variable explicada por el factor único de cada variable o unicidad (columna u2).

A diferencia del ACP, no pedimos modelos con tantos factores como variables para una primera exploración, ya que dicha solución en el caso de factorial no converge. Esto se debe a que a diferencia del ACP, AFC trabaja sólo con la covarianza, y no con la varianza total.

```
## Factor Analysis using method =  pa
## Call: fa(r = completedata_rf, nfactors = 2, rotate = "none", max.iter = 500,
##     fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##            PA1   PA2   h2    u2 com
## pvp_euro  0.81  0.35 0.78 0.222 1.4
## cilindro  0.81 -0.12 0.67 0.330 1.0
## cc        0.85  0.19 0.76 0.245 1.1
## potencia  0.96 -0.18 0.95 0.045 1.1
## peso      0.63  0.52 0.67 0.332 1.9
## cons90    0.74  0.16 0.57 0.429 1.1
## cons120   0.75  0.24 0.62 0.382 1.2
## consurb   0.82 -0.06 0.68 0.320 1.0
## velocida  0.78 -0.37 0.75 0.250 1.4
## acelerac -0.74  0.63 0.94 0.057 1.9
##
##                        PA1  PA2
## SS loadings           6.29 1.09
## Proportion Var        0.63 0.11
## Cumulative Var        0.63 0.74
## Proportion Explained  0.85 0.15
## Cumulative Proportion 0.85 1.00
##
## Mean item complexity =  1.3
## Test of the hypothesis that 2 factors are sufficient.
##
## The degrees of freedom for the null model are  45  and the objective function was  12.
03 with Chi Square of  1321.6
## The degrees of freedom for the model are 26  and the objective function was  2.74
##
## The root mean square of the residuals (RMSR) is  0.06
## The df corrected root mean square of the residuals is  0.09
##
## The harmonic number of observations is  115 with the empirical chi square  43.3  with
prob <  0.018
## The total number of observations was  115  with Likelihood Chi Square =  297.19  with
prob <  7.7e-48
##
## Tucker Lewis Index of factoring reliability =  0.628
## RMSEA index =  0.311  and the 90 % confidence intervals are  0.272 0.334
## BIC =  173.82
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy
##                                                 PA1  PA2
## Correlation of (regression) scores with factors   0.99 0.95
## Multiple R square of scores with factors          0.99 0.91
## Minimum correlation of possible factor scores     0.97 0.81
```

A partir de los autovalores, podemos pensar que sería razonable trabajar con una solución de 2 factores atendiendo a la regla de Kaiser (ya que sólo 2 factores logran autovalor > 1).

Existe, sin embargo, y a diferencia del ACP, otro indicador que ayuda a tomar la decisión a nivel estadístico. Este indicador es el promedio de las comunalidades individuales (promedio de la columna h2, que indica para cada variable la comunalidad de cada una de estas con el conjunto de variables restantes). El promedio de las comunalidades individuales funciona como límite que indica con cuántos factores es razonable quedarse. *(Zamora, Esnaola and Boccardo, 2015)*.

En este caso, al resultar 0.73 resulta razonable la solucion de 2 factores. Este resultado supone una perdida de varianza total explicada. En este sentido, cabe mencionar que en el AFC las comunalidades individuales muestran cómo covaría la variable individual con el resto de las variables. En caso de que alguna variable tuviese comunalidad < 0.3, existiría problemas para la buena convergencia, parsimonia e interpretabilidad de los modelos. En este sentido, cuando la covarianza (comunalidad) es similar a la varianza total (cercana a 1) se espera que los modelos AFC y ACP sean similares en sus resultados. En casos en que son muy distintas, por ejemplo comunalidades de 0,7 o menos, probablemente ambos modelos darán resultados diferentes *(Zamora, Esnaola and Boccardo, 2015)*.

Así, podemos comprobar que algunas comunalidades de nuestro modelo AFC son inferiores a 0.7, lo que puede dar pie a las diferencias existentes entre ambos modelos.

Dado que no creamos que exista una realidad subyacente e indetectable a simple vista para explicar las características de los todoterreno (como sería el caso del papel que juga la inteligencia en la obtención de buenas notas por parte de un grupo de alumnos) y sabiendo que nuestro objetivo es explicar el máximo de varianza posible de cara a conocer las carácterísticas fundamentales de los todotererno, descartamos esta solución frente a la solución rotada de 2 componentes anteriormente descrita.

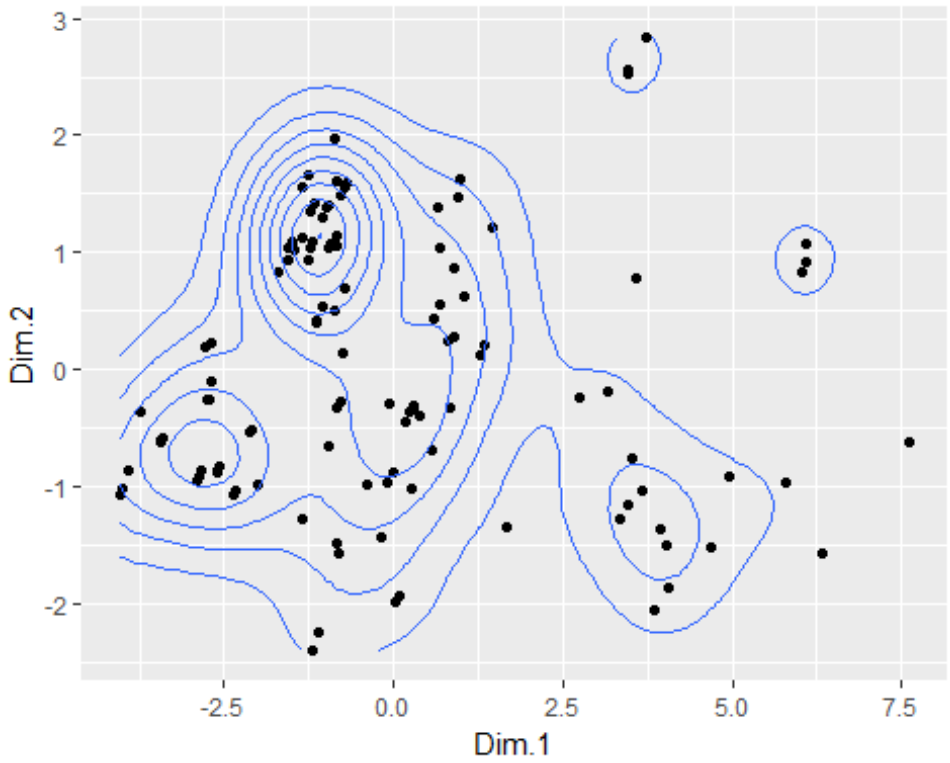
3. Análisis Clúster:

Con el objetivo de obtener más información sobre los todos terrenos, vamos a realizar un análisis clúster para comprobar si podemos agrupar los vehículos en función de las variables explicativas.

3.1. ¿Tiene sentido realizar el análisis clúster?

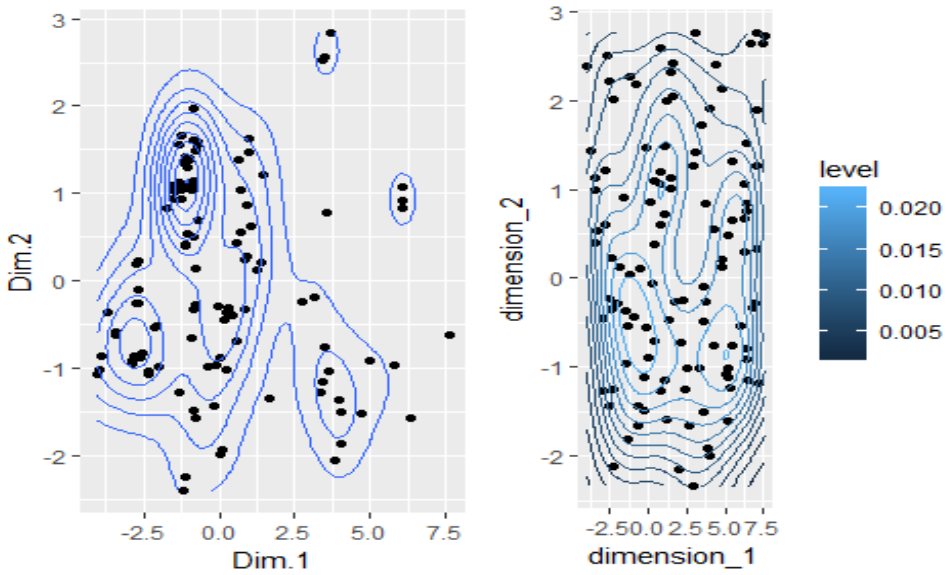
Para realizar el análisis clúster vamos a utilizar solo los dos factores obtenidos mediante el método de los componentes principales antes de su rotación. Recordemos que con estos dos factores se explica el 80% de la variabilidad de las observaciones. Podemos observar cómo podemos utilizar el método de los componentes principales como técnica de reducción de la dimensión.

Vamos a visualizar un gráfico de densidad para tener una idea de la distribución de las observaciones en las componentes principales 1 y 2 y ver si hay indicios de asociación:



Observamos que hay cierta asociación en función de la posición de las observaciones, distinguiendo un sector más cercano al eje 2 y otro más alejado.

Vamos a comparar este gráfico de densidad con el pertinente a un grupo de datos generados aleatoriamente para ambas dimensiones con el objetivo de probar que ciertamente existen patrones de asociación:



Podemos apreciar las diferencias en las densidades de ambos gráficos. El patrón de asociación de los datos generados aleatoriamente es distinto al de la muestra. En la muestra podríamos distinguir 2 grupos en función de su posición en respecto de la dimensión 1 (grupo 1 < 2.5 < grupo 2), además de subgrupos

claros por la densidad. En el gráfico de datos generados aleatoriamente, aunque existe cierta asociación, cuesta más distinguir grupos concretos.

3.1.1. Evaluación de la bondad del análisis clúster

Vamos a comprobar si efectivamente existen argumentos para proceder a realizar un análisis clúster. Para ello utilizaremos el **Estadístico de Hopkins**.

El estadístico de Hopkins se trata de un contraste frente a la estructura aleatoria a través de una distribución uniforme del espacio de datos; la idea es contrastar una hipótesis de distribución uniforme / aleatoria de los datos frente a su alternativa (que no lo sea); de aceptarse la hipótesis nula, no existirían grupos de observaciones interesantes en el conjunto analizado. Valores próximos a 0.5 señalan promedios de distancias entre vecinos los más próximos muy similares, haciendo irreal e inoperante el agrupamiento; por el contrario, valores próximos a 0 permitirían rechazar (*López Zafra, 2017*).

Calculamos el estadístico para la muestra de todoterrenos:

```
## $H
## [1] 0.2362346
```

El estadístico es bastante inferior a 0.5, permitiendo rechazar la hipótesis nula de que la distribución de los datos es aleatoria y, por tanto, tiene sentido llevar a cabo un análisis clúster.

Calculamos el estadístico para el conjunto de datos aleatorios:

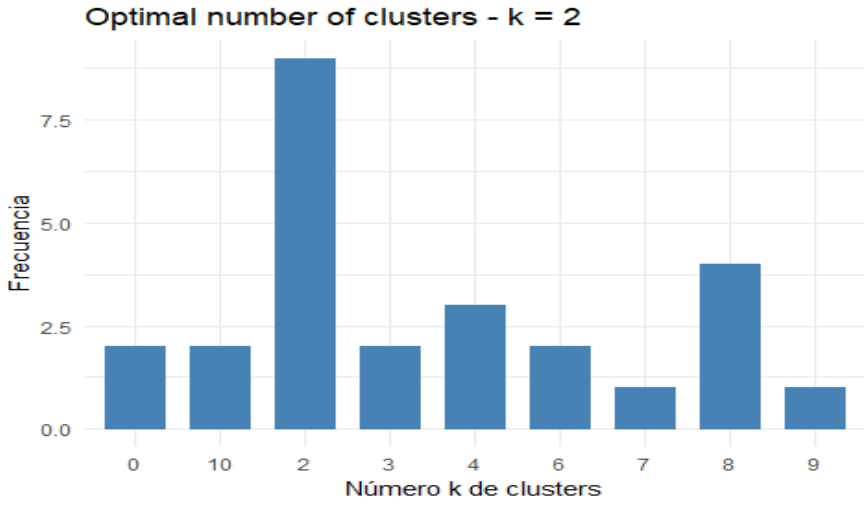
```
## $H
## [1] 0.5542418
```

Como se puede apreciar, con los datos aleatorios el estadístico de Hopkins tiene un valor muy próximo a 0.5 y, por tanto, indica que no hay posibilidad de realizar un análisis clúster.

3.2. Identificación del número de grupos.

Una vez aceptada la conveniencia de llevar a cabo el análisis clúster, hemos de determinar el número de clústeres a utilizar.

Para ello utilizaremos el paquete de R NbClust, que proporciona 30 índices para determinar el número de clústeres y propone al usuario el mejor esquema de agrupamiento a partir de los diferentes resultados obtenidos al variar todas las combinaciones de número de clústeres, medidas de distancia y métodos de agrupamiento (*Charrad et al., 2015*)



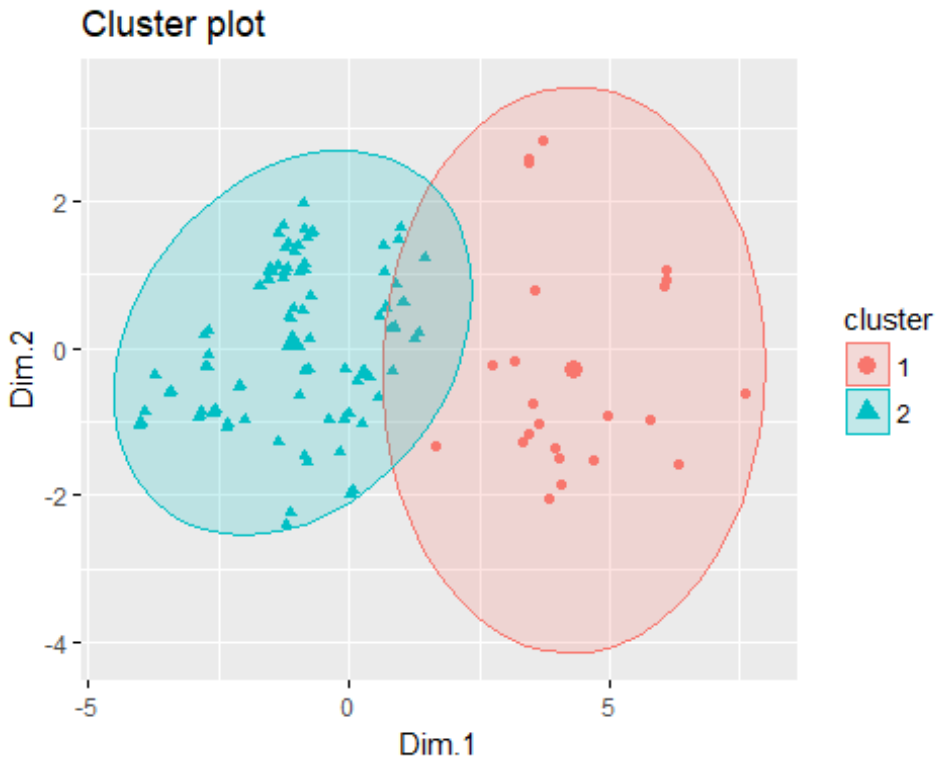
Tal y como podemos observar, 9 de los 30 Índices indican que el número óptimo de clusters es 2. Aplicando la regla de la mayoría este será, en principio, el número de clusters que utilizaremos.

3.3. Aplicación del algoritmo K-means

Tras el estudio de la muestra, viendo que las variables explicativas son de carácter cuantitativo y habiendo hecho las pruebas que permiten establecer el número adecuado de grupos antes de proceder a la segmentación, procederemos a aplicar un método no jerárquico. Concretamente, aplicaremos el algoritmo *K-means*.

Podríamos definir el algoritmo K-means como un algoritmo de clasificación no supervisada (es decir, los datos no tienen etiquetas y se clasifican a partir de su estructura interna (propiedades y características) que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o clúster (*unioviedo.es, n.d.*). No resulta invariante ante cambios de escala (*López Zafra, 2017*), por eso hemos escalado las variables.

Observamos los 2 grupos resultantes:



Observamos que existen dos grupos bien definidos. Aunque existe cierto solapamiento en algunas observaciones del clúster 2 que se encuentran clasificadas dentro del cluster1. Vamos a observar su composición.

3.4. Interpretación de los grupos

Clúster 1:

```
##     pvp_euro        cilindro           cc          potencia
##  Min.   :21672   Min.   :6.000   Min.   :2959   Min.   :136.0
##  1st Qu.:31676   1st Qu.:6.000   1st Qu.:3182   1st Qu.:173.5
##  Median :39657   Median :6.000   Median :3497   Median :181.0
##  Mean   :44436   Mean   :6.261   Mean   :3618   Mean   :182.2
##  3rd Qu.:62362   3rd Qu.:6.000   3rd Qu.:3960   3rd Qu.:208.0
##  Max.   :69461   Max.   :8.000   Max.   :5216   Max.   :225.0
##
##       peso          cons90         cons120         consurb
##  Min.   :1455   Min.   : 7.40   Min.   :11.00   Min.   : 9.80
##  1st Qu.:1778   1st Qu.:10.30   1st Qu.:13.90   1st Qu.:15.75
##  Median :1925   Median :10.80   Median :14.60   Median :17.30
##  Mean   :1944   Mean   :10.92   Mean   :14.94   Mean   :16.57
##  3rd Qu.:2130   3rd Qu.:11.60   3rd Qu.:16.20   3rd Qu.:18.30
##  Max.   :2320   Max.   :13.70   Max.   :18.50   Max.   :22.10
##
##    velocida      acelerac     cluster_kmeans  dataset_mod$marca
##  Min.   :145   Min.   : 9.40   Min.   :1      MERCEDES  :6
##  1st Qu.:170   1st Qu.:10.55   1st Qu.:1      MITSUBISHI:5
##  Median :175   Median :11.50   Median :1      JEEP      :4
##  Mean   :173   Mean   :11.83   Mean   :1      LAND ROVER:2
##  3rd Qu.:180   3rd Qu.:12.33   3rd Qu.:1      OPEL      :2
##  Max.   :196   Max.   :16.00   Max.   :1      TOYOTA    :2
##                                               (Other)   :2
##                                                     dataset_mod$modelo
##  Monterey 3.2i V6 24V                                          : 2
##  Montero Corto 3.0 GL                                          : 2
##  4 Runner V6                                                   : 1
##  Blazer Aut.                                                   : 1
##  Cherokee 4.0 Jambore                                          : 1
##  Explorer 4.0 V6 XLT                                           : 1
##  (Other)                                                       :15
```

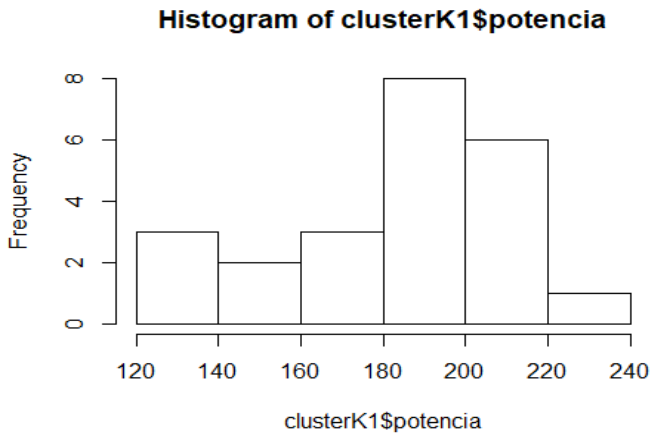
El clúster 1 está compuesto por 23 todoterrenos. Este grupo está caracterizado por una elevada potencia de motor (el mínimo es el máximo del resto de todoterrenos, 136CV) y mayor cilindrada (el mínimo es 6 cilindros, que también es el valor medio, aunque llegan hasta 8). También se caracterizan por un peso medio elevado, una aceleración media más baja respecto al resto de observaciones y una velocidad máxima media superior. Por lo general son todoterrenos con un consumo relativamente alto, tanto urbano como a 90 y 120 km/h. El precio medio de estos vehículos es el doble que el del resto de todoterrenos.

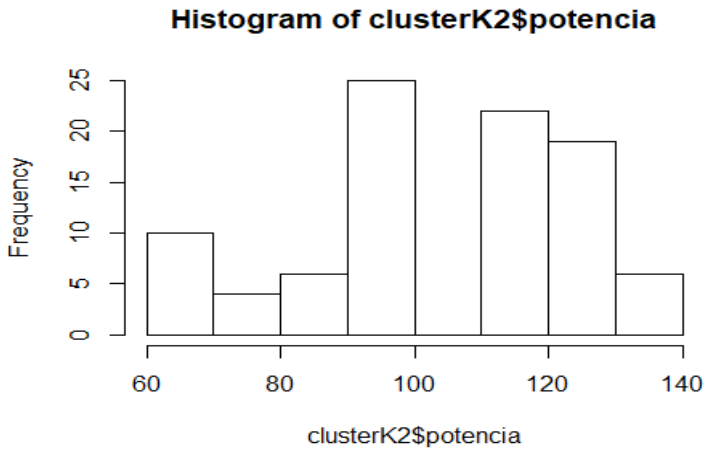
Cluster 2:

```
##     pvp_euro        cilindro          cc          potencia
##  Min.   : 9113   Min.   :4.00   Min.   :1298   Min.   : 64.0
##  1st Qu.:16906   1st Qu.:4.00   1st Qu.:1986   1st Qu.: 95.0
##  Median :22968   Median :4.00   Median :2477   Median :112.0
##  Mean   :23180   Mean   :4.13   Mean   :2301   Mean   :104.3
##  3rd Qu.:28437   3rd Qu.:4.00   3rd Qu.:2663   3rd Qu.:121.2
##  Max.   :52880   Max.   :6.00   Max.   :3059   Max.   :136.0
##
##       peso          cons90          cons120         consurb
##  Min.   : 930   Min.   : 6.600   Min.   : 8.40   Min.   : 8.10
##  1st Qu.:1277   1st Qu.: 7.600   1st Qu.:10.00   1st Qu.:10.20
##  Median :1696   Median : 8.400   Median :11.45   Median :11.60
##  Mean   :1600   Mean   : 8.391   Mean   :11.44   Mean   :11.59
##  3rd Qu.:1850   3rd Qu.: 9.100   3rd Qu.:12.88   3rd Qu.:12.93
##  Max.   :2115   Max.   :10.600   Max.   :16.20   Max.   :18.10
##
##     velocida        acelerac     cluster_kmeans  dataset_mod$marca
##  Min.   :125.0   Min.   :12.30   Min.   :2      SUZUKI    :19
##  1st Qu.:140.0   1st Qu.:14.85   1st Qu.:2      NISSAN    :14
##  Median :145.0   Median :17.21   Median :2      LAND ROVER:13
##  Mean   :147.0   Mean   :16.72   Mean   :2      MITSUBISHI:10
##  3rd Qu.:155.8   3rd Qu.:17.88   3rd Qu.:2      OPEL      : 7
##  Max.   :170.0   Max.   :22.00   Max.   :2      FORD      : 6
##                                                 (Other)   :23
##                                                     dataset_mod$modelo
##  Montero La. TDI 2.8                                            : 3
##  Maverick 2.7 TD GLS                                            : 2
##  Montero Co. TDI 2.5                                            : 2
##  Montero Co. TDI 2.8                                            : 2
##  Rocsta 2.2 DX techo                                            : 2
##  Terrano II 2.7 TD LX                                           : 2
##  (Other)                                                        :79
```

Este segundo grupo está compuesto por aquellos todoterrenos con un motor menos potente por lo general y de menor rendimiento en velocidad, aunque con más aceleración de media. Son vehículos con un consumo medio inferior al del clúster 1 y tienen un precio medio más bajo. El número de cilindros suele ser 4, aunque algunos todoterrenos de este grupo tienen 6 cilindros.

Para observar mejor las diferencias, mostramos el histograma respecto a potencia:





Observamos unas diferencias notables puesto que la mayoría de todoterrenos del clúster 1 se sitúa entre los 160 y los 220 CV, mientras que la mayor parte de los miembros del clúster 2 están entre 90 y 130 CV.

En este sentido vamos estudiar las gamas de modelos de las distintas marcas en función de su pertenencia al clúster 1 (todoterrenos de elevada potencia, cilindrada, velocidad y alto consumo) o al clúster 2 (todoterrenos de potencia, velocidad y consumos medios).

```
##
##                1  2
##   ASIA MOTORS  0  3
##   CHEVROLET    1  0
##   DAIHATSU     0  1
##   FORD         1  6
##   JEEP         4  6
##   KIA          0  2
##   LADA         0  2
##   LAND ROVER   2 13
##   MERCEDES     6  0
##   MITSUBISHI   5 10
##   NISSAN       0 14
##   OPEL         2  7
##   SSANGYONG    0  3
##   SUZUKI       0 19
##   TATA         0  2
##   TOYOTA       2  4
##   UAZ          0  0
```

Como se puede apreciar en la tabla hay marcas como Suzuki y Nissan con un gran número de modelos y todos ellos pertenecen al clúster 2. Podemos por tanto concluir que son marcas de consumo medio (para este tipo de coches), cilindrada y potencia media baja. Hay marcas con pocos coches todo terreno especializados también en el segmento de potencia media. baja como son Asia Motors, Daihatsu, Kia, Lada, Ssanyong y Tata.

Por otro lado, hay marcas como Landrover, Mitsubishi, Jeep, Toyota, Opel y Ford que tienen dentro de sus catálogos coches de las 2 categorías, lo que indica que son marcas que tienen un segmento de clientes más amplio.

Por último, destaca la marca Mercedes pues es la única que tiene todos sus coches dentro del segmento de alta potencia. Lo cual no es algo que nos extraña dado el perfil de clientes, la calidad de los motores y los precios de los todoterrenos de esta marca.

4. Conclusiones

Hemos llevado a cabo un proceso de reducción de la dimensión para pasar de 15 variables, 12 numéricas y 3 de tipo nominal, a dos componentes principales resultantes de la combinación lineal de 10 de las variables numéricas, todo plenamente justificado y siguiendo el criterio de parsimonia.

De todos los métodos empleados para acometer dicho fin, el más explicativo ha sido el método de componentes principales, que ha permitido explicar el 78.54%% de la varianza inicial tras la reducción de la dimensión mencionada. Por tanto, se ha logrado explicar, con éxito, las características fundamentales de los vehículos todoterreno reduciendo el número de variables explicativas.

A pesar de esto, los componentes generados han resultado de difícil interpretación. Por esta razón y a fin de facilitar la interpretación de los componentes, se ha propuesto una solución rotada por el método *varimax* de 2 componentes que, explicando la misma proporción de varianza total, permite una interpretación más sencilla de los factores gracias a una mejor identificación de a qué variable latente (componente o factor) tiende a asociarse cada variable observada. En este sentido uno de los factores centra más su atención en las propiedades más intrínsecas al rendimiento del motor, como son la potencia, la velocidad y la aceleración, mientras que el otro factor hace más hincapié en otras características como el peso, el precio y el consumo.

A partir de los dos componentes principales originados en la solución no rotada, se ha realizado un análisis clúster para agrupar los vehículos todoterreno en función de sus características. Se han identificado, con éxito, dos grupos claramente diferenciados que corresponderían a los vehículos de más potencia y rendimiento (con un precio medio muy superior) y a los vehículos más comunes en cuanto a prestaciones de motor (y precio medio inferior). Esta clasificación nos ha permitido distinguir las compañías que solo lanzan al mercado todoterrenos de potencia, velocidad y consumos medios, de las que lanzan este tipo de todoterreno, pero también otros modelos de elevada potencia, cilindrada, velocidad y alto consumo, así como de una compañía que se posiciona en el mercado por vender exclusivamente todoterrenos del segundo tipo.

Por estas razones, consideramos que se han alcanzado con éxito los objetivos propuestos al inicio del proyecto, facilitando la comprensión tanto de las características fundamentales de los vehículos todoterreno como de los segmentos en los que estos vehículos pueden agruparse.

5. Bibliografia

Charrad, M., Ghazzali, N., Boiteau, V. and Niknafs, A. (2015). *Package 'NbClust'*. [pdf] cran.r-project. Available at: https://cran.r-project.org/web/packages/NbClust/NbClust.pdf [Accessed 4 Jan. 2018].

Daróczi, G. (2015). *Mastering Data Analysis with R*. Birmingham: PACKT Publishing, pp.193-235.

Ibm.com. (n.d.). *IBM Knowledge Center*. [online] Available at: https://www.ibm.com/support/knowledgecenter/es/SSLVMB_22.0.0/com.ibm.spss.statistics.help/spss/base/idh_fact_rot.htm [Accessed 3 Jan. 2018].

López Zafra, J. (2017). *El Análisis Clúster*. Madrid: Máster en Data Science para Finanzas - CUNEF.

López Zafra, J. (2017). *El análisis clúster. Determinación del número de grupos*.

Support.sas.com. (n.d.). *SAS/STAT(R) 9.2 User's Guide, Second Edition*. [online] Available at: https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_factor_sect022.htm [Accessed 3 Jan. 2018].

Tipos de desarrollos y de relación en caja de cambios. (2018). [Blog] *Tecnologia del automovil*. Available at: http://autastec.com/blog/tecnologias-limpias/desarrollos-caja-cambios/ [Accessed 2 Jan. 2018].

Tutorial on 5 Powerful R Packages used for imputing missing values. (2016). [Blog] *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/ [Accessed 28 Dec. 2017].

unioviedo.es. (2018). *El algoritmo k-means aplicado a clasificación y procesamiento de imágenes*. [online] Available at: https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html [Accessed 4 Jan. 2018].

Zamora, R., Esnaola, J. and Boccardo, G. (2015). Guía de trabajo en "R": ANÁLISIS FACTORIAL Y ANÁLISIS DE COMPONENTES PRINCIPALES. Departamento de Sociología - Universidad de Chile.

6. Anexo: Código

Metodología empleada

Análisis exploratorio

```
library(foreign) #Cargamos la libreria foreign para importar el fichero de tipo sav
dataset = read.spss("C:/Users/valen/Desktop/Master Datascience/Tecnicas de reduccion y
agrupacion/Practica todoterrenos/tterreno_euro.sav", to.data.frame=TRUE)
dataset$plazas <- as.numeric(paste(dataset$plazas))
dataset$cilindro <- as.numeric(paste(dataset$cilindro))
if(nrow(unique(dataset))== nrow(dataset)){
  print("No hay duplicados")
}
str(dataset)
```

Variables numéricas

```
#Resumen pvp euro
summary(dataset$pvp_euro)
#Resumen numero cilindros
library(plyr)
numero_cilindros <- count(dataset$cilindro)
numero_cilindros[,"freq relativa"] <- numero_cilindros$freq/sum(numero_cilindros$freq)
frecuencias_acumuladas <- NULL
for (i in 1:nrow(numero_cilindros)){
  frecuencias_acumuladas <- c(frecuencias_acumuladas,sum(numero_cilindros$`freq relativa`
[1:i]))
}
numero_cilindros[,"frecuencia_acumulada"] <- frecuencias_acumuladas
numero_cilindros
#Resumen cilindrada
summary(dataset$cc)
#Resumen potencia
summary(dataset$potencia)
#Resumen RPM
intervalos <- seq(from=3.6, to=6.8, by=0.4)
rpm_cut <- cut(dataset$rpm/1000,intervalos, right=FALSE)
rpm_freq <- table(rpm_cut)
rpm_freq <- as.data.frame(rpm_freq)
rpm_freq[,"frecuencia_relativa"] <- rpm_freq$Freq/sum(rpm_freq$Freq)
frecuencias_acumuladas <- NULL
for (i in 1:nrow(rpm_freq)){
  frecuencias_acumuladas <- c(frecuencias_acumuladas,sum(rpm_freq$frecuencia_relativa[1:i
]))
}
rpm_freq[,"frecuencia_acumulada"] <- frecuencias_acumuladas
print(rpm_freq,row.names = FALSE)
#Resumen peso
summary(dataset$peso)
#Resumen numero plazas
library(plyr)
numero_plazas <- count(dataset$plazas)
numero_plazas[,"freq relativa"] <- numero_plazas$freq/sum(numero_plazas$freq)
frecuencias_acumuladas <- NULL
for (i in 1:nrow(numero_plazas)){
  frecuencias_acumuladas <- c(frecuencias_acumuladas,sum(numero_plazas$`freq relativa`[1:
i]))
}
numero_plazas[,"frecuencia_acumulada"] <- frecuencias_acumuladas
numero_plazas
```

```
#Consumo 90
breaksc90 <- seq(from=6.60,to= 14.10,by = 1.5)
consumo90_cut <- cut(dataset$cons90,breaksc90, right=FALSE)
consumo90_freq <- table(consumo90_cut)
consumo90_freq <- as.data.frame(consumo90_freq)
consumo90_freq[,"frecuencia_relativa"] <- consumo90_freq$Freq/sum(consumo90_freq$Freq)
frecuencias_acumuladas <- NULL
for (i in 1:nrow(consumo90_freq)){
  frecuencias_acumuladas <- c(frecuencias_acumuladas,sum(consumo90_freq$frecuencia_relati
va[1:i]))
}
consumo90_freq[,"frecuencia_acumulada"] <- frecuencias_acumuladas
print(consumo90_freq,row.names = FALSE)
#Consumo 120
breaksc120 <- seq(from=8.40,to= 18.40,by = 2)
consumo120_cut <- cut(dataset$cons120,breaksc120, right=FALSE)
consumo120_freq <- table(consumo120_cut)
consumo120_freq <- as.data.frame(consumo120_freq)
consumo120_freq[,"frecuencia_relativa"] <- consumo120_freq$Freq/sum(consumo120_freq$Freq)
frecuencias_acumuladas <- NULL
for (i in 1:nrow(consumo120_freq)){
  frecuencias_acumuladas <- c(frecuencias_acumuladas,sum(consumo120_freq$frecuencia_relat
iva[1:i]))
}
consumo120_freq[,"frecuencia_acumulada"] <- frecuencias_acumuladas
print(consumo120_freq,row.names = FALSE)
#Consumo urbano
z <- na.omit(dataset$consurb)
hist(z,freq=FALSE,breaks=8,
     xlab="Litros",
     main="Distribucion Consumo urbano")
curve(dnorm(x, mean=mean(z), sd=sd(z)),
        add=TRUE, col="blue", lwd=2)
lines(density(z)$x, density(z)$y,
        col="red", lwd=2, lty=2)
#Velocidad
breaksvelocidad <- seq(from=120,to= 200,by = 10)
velocidad_cut <- cut(dataset$velocida,breaksvelocidad, right=FALSE)
velocidad_freq <- table(velocidad_cut)
velocidad_freq <- as.data.frame(velocidad_freq)
velocidad_freq[,"frecuencia_relativa"] <- velocidad_freq$Freq/sum(velocidad_freq$Freq)
frecuencias_acumuladas <- NULL
for (i in 1:nrow(velocidad_freq)){
  frecuencias_acumuladas <- c(frecuencias_acumuladas,sum(velocidad_freq$frecuencia_relati
va[1:i]))
}
velocidad_freq[,"frecuencia_acumulada"] <- frecuencias_acumuladas
print(velocidad_freq,row.names = FALSE)
#Aceleracion
z <- na.omit(dataset$acelerac)
hist(z,freq=FALSE,breaks=8,
     xlab="Segundos",
     main="Distribucion Aceleración")
lines(density(z)$x, density(z)$y,
        col="red", lwd=2, lty=2)
```

Variables nominales

```
#marca
Marca <- summary.factor(dataset$marca)
Marca <- as.matrix(Marca,ncol=2)
```

```
Marca <- as.data.frame(Marca)
Marca[,"Nombre"] <- rownames(Marca)
rownames(Marca) <- NULL
Marca <- Marca[order(-Marca$V1),]
Marca2 <- as.data.frame(cbind(Marca$Nombre,Marca$V1))
head(Marca2)
tail(Marca2)
#modelos
numero_modelos <- count(dataset$modelo)
count_modelos <- NULL
for (i in 1:nrow(dataset)){
      count_modelos <- c(count_modelos,numero_modelos[dataset$modelo[i],2])
}
dataset[,"Numero_modelos"] <- count_modelos
modelos_repetidos <- data.frame(dataset$modelo)
modelos_repetidos <- cbind(modelos_repetidos,count_modelos)
colnames(modelos_repetidos) <- c("modelo","numero_de_veces")
unique(modelos_repetidos[modelos_repetidos$numero_de_veces>1,])
#modelos duplicados
dataset_duplicados <- dataset[dataset$Numero_modelos>1,]

tabla_diferencias <- dataset_duplicados %>%
  group_by(modelo) %>%
  summarise(Diferencias_precio= round(max(pvp_euro)-min(pvp_euro),2),
            Diferencias_plazas= round(max(plazas)-min(plazas),2),
            Diferencias_consumo_120 = round(max(cons120)-min(cons120),2)) %>%
  arrange(desc(Diferencias_precio))
tablas_diferencias <- as.data.frame(tabla_diferencias)
print(tabla_diferencias)
```

Imputacion NAs

```
require(mice)
library(VIM)
mice_plot <- aggr(dataset[,-16], col=c('navyblue','yellow'),
                  numbers=TRUE, sortVars=TRUE,
                  labels=names(dataset), cex.axis=.7,
                  gap=3, ylab=c("Missing data","Pattern"))
dataset_mod <- dataset[!(is.na(dataset$acelerac)&is.na(dataset$cons120)&is.na(dataset$con
s90)),]
#Mice
imputed_Data <- mice(dataset_mod[,c(-1,-2,-15,-16)], m=5, maxit = 50, method = 'pmm', see
d = 123)
require(lattice)
imp_tot2 <- complete(imputed_Data, "long", inc = TRUE)
head(imp_tot2)
col <- rep(c("blue", "red")[1 + as.numeric(is.na(imputed_Data$data$acelerac))], 6)
stripplot(acelerac ~ .imp, data = imp_tot2, jit = TRUE, col = col, xlab = "imputation Num
ber")
completeData1 <- complete(imputed_Data,1)
completeData2 <- complete(imputed_Data,2)
completeData3 <- complete(imputed_Data,3)
completeData4 <- complete(imputed_Data,4)
completeData5 <- complete(imputed_Data,5)
CompleteData <- (completeData1 + completeData2 + completeData3 + completeData4 + complete
Data5)/5
#Missforest
library(missForest)
imputed_data_rf <- missForest(dataset_mod[,c(-1,-2,-15,-16)])
imputed_data_rf$OOBerror
completedata_rf <- imputed_data_rf$ximp
cor.mat3 = round(cor(completedata_rf),2)
```

```
#Eleccion método
diferencia_mice <- cor.mat - cor.mat2
flattenCorrMatrix <- function(cormat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut]
    )
}
diferencia_mice_flat <- flattenCorrMatrix(diferencia_mice)
error_mice1 <- round(sum(abs(diferencia_mice_flat$cor))/nrow(diferencia_mice_flat),3)
error_mice1
diferencia_rf <- cor.mat- cor.mat3
diferencia_rf_flat <- flattenCorrMatrix(diferencia_rf)
error_rf1 <- round(sum(abs(diferencia_rf_flat$cor))/nrow(diferencia_rf_flat),3)
error_rf1
```

Pertinencia realizar análisis factorial

```
#Analisis matriz de correlaciones
require(Hmisc)
cor.mat.nds= rcorr(as.matrix(completedata_rf))
require(corrplot)
corrplot(cor.mat3, type="lower", order="original", tl.col="black", tl.cex=0.7, tl.srt=45)
corrplot(cor.mat3, type="full", order="hclust", addrect = 2,
         tl.col="black", tl.cex=0.7, tl.srt=45)
#Determinante de la matriz de correlaciones
det(cor.mat3)
#Test de Barlett
library(psych)
print(cortest.bartlett(cor.mat3, n=nrow(completedata_rf)))
#Kmo global
require(ppcor)
p.cor.mat=pcor(completedata_rf)
p.cor.mat2=as.matrix(p.cor.mat$estimate)
kmo.num1 = sum(cor.mat3^2) - sum(diag(cor.mat3^2)) #numerador
kmo.denom1 = kmo.num1 + (sum(p.cor.mat2^2) - sum(diag(p.cor.mat2^2))) #denominador
kmo1 = round(kmo.num1/kmo.denom1,2)
kmo1
#Kmo parcial
p.cor.mat2=data.frame(p.cor.mat2)
rownames(p.cor.mat2) = c(rownames(cor.mat))
colnames(p.cor.mat2)=c(colnames(cor.mat))
p.cor.mat2
variable1 <- NULL
valores1  <- NULL
for (j in 1:ncol(completedata_rf)){
        kmo_j.num <- sum(cor.mat[,j]^2) - cor.mat[j,j]^2  #cojo todos los elementos de la
fila 1 menos el 1 al cuadrado
        kmo_j.denom <- kmo_j.num + (sum(p.cor.mat2[,j]^2) - p.cor.mat2[j,j]^2) #lo mismo
pero con los cuadrados
        kmo_j <- kmo_j.num/kmo_j.denom
        variable1 <- c(variable1,colnames(completedata_rf)[j])
        valores1 <- c(valores1,kmo_j)
        df_msa1 <- data.frame(variable1,valores1)
        print(paste(colnames(completedata_rf)[j],"=",kmo_j))
}
#Sin RPM y Plazas
#Errores imputacion
#Mice
imputed_Data <- mice(dataset_mod[,c(-1,-2,-7,-9,-15,-16)], m=5, maxit = 50, method = 'pmm
```

```
', seed = 123)
completeData1 <- complete(imputed_Data,1)
completeData2 <- complete(imputed_Data,2)
completeData3 <- complete(imputed_Data,3)
completeData4 <- complete(imputed_Data,4)
completeData5 <- complete(imputed_Data,5)
CompleteData <- (completeData1 + completeData2 + completeData3 + completeData4 + complete
Data5)/5
cor.mat = round(cor(dataset[,c(-1,-2,-7,-9,-15,-16)], use="complete.obs"),2)
cor.mat2 = round(cor(CompleteData),2)
diferencia_mice <- cor.mat - cor.mat2
flattenCorrMatrix <- function(cormat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut]
    )
}
diferencia_mice_flat <- flattenCorrMatrix(diferencia_mice)
error_mice2 <- round(sum(abs(diferencia_mice_flat$cor))/nrow(diferencia_mice_flat),3)
error_mice2
#RF
imputed_data_rf2 <- missForest(dataset_mod[,c(-1,-2,-7,-9,-15,-16)])
completedata_rf <- imputed_data_rf2$ximp
cor.mat3 = round(cor(completedata_rf),2)
diferencia_rf <- cor.mat- cor.mat3
diferencia_rf_flat <- flattenCorrMatrix(diferencia_rf)
error_rf2 <- round(sum(abs(diferencia_rf_flat$cor))/nrow(diferencia_rf_flat),3)
error_rf2
#Matriz de correlacion
corrplot(cor.mat3, type="full", order="hclust", addrect = 2,
         tl.col="black", tl.cex=0.7, tl.srt=45)
#Kmo
p.cor.mat=pcor(completedata_rf)
p.cor.mat2=as.matrix(p.cor.mat$estimate)
kmo.num2 = sum(cor.mat3^2) - sum(diag(cor.mat3^2)) #numerador
kmo.denom2 = kmo.num2 + (sum(p.cor.mat2^2) - sum(diag(p.cor.mat2^2))) #denominador
kmo2 = round(kmo.num2/kmo.denom2,2)
table(kmo1,kmo2)
#Msa
p.cor.mat2=data.frame(p.cor.mat2)
#Ponemos como nombre de filas y columnas el nombre de las filas y columnas de la matriz d
e correlaciones
rownames(p.cor.mat2) = c(rownames(cor.mat))
colnames(p.cor.mat2)=c(colnames(cor.mat))
variable2 <- NULL
valores2 <- NULL
for (j in 1:ncol(completedata_rf)){
        kmo_j.num <- sum(cor.mat[,j]^2) - cor.mat[j,j]^2  #cojo todos los elementos de la
fila 1 menos el 1 al cuadrado
        kmo_j.denom <- kmo_j.num + (sum(p.cor.mat2[,j]^2) - p.cor.mat2[j,j]^2) #lo mismo
pero con los cuadrados
        kmo_j <- kmo_j.num/kmo_j.denom
        variable2 <- c(variable2,colnames(completedata_rf)[j])
        valores2 <- c(valores2,kmo_j)
        df_msa2 <- data.frame(variable2,valores2)
}
variable3 <- c("rpm","plazas")
valores3 <- c(0,0)
df_msa3 <- data.frame(variable3,valores3)
```

```
colnames(df_msa3) <- c("variable2","valores2")
df_msa2 <- df_msa2[order(df_msa2$variable2),]
df_msa2 <- rbind(df_msa2[1:7,],df_msa3[2,],df_msa2[8:9,],df_msa3[1,],df_msa2[10,])
df_msa1 <- df_msa1[order(df_msa1$variabl),]
df_msa <- data.frame(df_msa1$variable1,df_msa1$valores1,df_msa2$valores2)
df_msa[,"Diferencia_msa"] <- df_msa$df_msa2.valores2-df_msa$df_msa1.valores1
colnames(df_msa) <- c("Variable","MSA1","MSA2","Diferencias MSA")
df_msa[c(-8,-11),c(1,4)]
sum(df_msa[c(-8,-11),4])
```

Análisis factorial

```
library(FactoMineR)
library(factoextra)
tterreno.acp = PCA(completedata_rf, scale.unit = TRUE, ncp = ncol(completedata_rf), graph
= TRUE)
autoval= tterreno.acp$eig
round(autoval, 2)
barplot(autoval[, 2], names.arg=1:nrow(autoval),
        main = "Varianza explicada por los CCPP",
        xlab = "Componentes Principales",
        ylab = "Porcentaje explicado de la varianza",
        col ="steelblue",
        ylim=c(0,105))
lines(x = 1:nrow(autoval), autoval[, 2],
      type="b", pch=19, col = "red")
lines(x = 1:nrow(autoval), autoval[, 3],
      type="o", pch=21, col = "blue", bg="grey")
tterreno.acp$var$coord #coordenadas de las observaciones
#si miramos las coordenadas de la dimension 1 y 2, podemos situar en el grafico circular
las variables.
#Comprobamos que el autovalor de cada factor j es la suma de los cuadrados de los a_ij
CP1=tterreno.acp$var$coord[,1]
CCP2=CP1^2
sum(CCP2)
#La suma conforma el autovalor del primer componente.
posiciones <- tterreno.acp$ind$coord[,1:2]
#creamos el objeto posiciones que nos servira para representar las observaciones y para a
gruparlas en analisis cluster
fviz_pca_var(tterreno.acp, col.var="cos2") +
        scale_color_gradient2(low="white", mid="blue",
                              high="green", midpoint=0.5) + theme_minimal()
A=as.matrix(tterreno.acp$var$contrib[,1:2])/100
B=as.matrix(tterreno.acp$eig[1:2,1])
A%*%B
fviz_contrib(tterreno.acp, choice = "var", axes = 1)
fviz_contrib(tterreno.acp, choice = "var", axes = 2)

apply(as.matrix(tterreno.acp$var$contrib), 2, sum)
#Comprobamos que suma 100 por columnas

fviz_pca_ind(tterreno.acp, alpha.ind="contrib") +
        theme_minimal()

#Matriz de componentes no rotados
TT.rotPrueba = principal(completedata_rf, nfactors=2, rotate ="none")
TT.rotPrueba
```

Rotaciones factoriales

```
#Varimax
tterreno.norm = data.frame (scale (completedata_rf)) # normalizamos
```

```
summary(tterreno.norm)
TT.rot = principal(tterreno.norm, nfactors=2, rotate ="varimax")
TT.rot
round(sapply(tterreno.norm, mean, na.rm = T), 2)
#Oblimin
require(GPArotation)
TT.rot.ob = principal(tterreno.norm, nfactors=2, rotate ="oblimin")
TT.rot.ob
```

Análisis cluster

```
#Cargamos las librerías.
library(cluster)
library(dendextend)
library(fpc)
library(factoextra)
library(NbClust)
df_cluster <- as.data.frame(posiciones)
library("ggplot2")
graf.datos <- ggplot(df_cluster, aes(x=Dim.1, y=Dim.2)) +
geom_point()+
geom_density_2d() # Estimación bidimensional de la densidad
graf.datos
#Generamos un conjunto aleatorio de datos para las dos variables
set.seed(123)
n = nrow(df_cluster)
random_df = data.frame(
x = runif(nrow(df_cluster), min(df_cluster$Dim.1), max(df_cluster$Dim.1)),
y = runif(nrow(df_cluster), min(df_cluster$Dim.2), max(df_cluster$Dim.2)))
#Colocamos en objeto para representación posterior
graf.aleat=ggplot(random_df, aes(x, y)) +
geom_point() +
labs(x="dimension_1",y="dimension_2") +
stat_density2d(aes(color = ..level..))
require(gridExtra)
grid.arrange(graf.datos, graf.aleat, nrow=1, ncol=2)
#Exige haber empaquetado los objetos, como hemos hecho; equivale a par(mfrow=c(f, c))
require(clustertend)
set.seed(123)
hopkins(df_cluster, n = nrow(df_cluster)-1)
#y ahora sobre los datos aleatorios
set.seed(123)
hopkins(random_df, n = nrow(random_df)-1)
nb.todos = NbClust(df_cluster, distance = "euclidean", min.nc = 2,
max.nc = 10, method = "complete", index ="all")
fviz_nbclust(nb.todos) + theme_minimal() +
labs(x="Número k de clusters", y="Frecuencia")
require(factoextra)
set.seed(123)
prueba1 = kmeans(df_cluster, 2)
cluster_kmeans <- prueba1$cluster
fviz_cluster(list(data = df_cluster, cluster = prueba1$cluster),
frame.type = "norm", geom = "point", stand = FALSE)
#representacion grafica
clasificacion2 <- data.frame(cluster_kmeans)
#Seguidamente introducimos los resultados del análisis clúster Kmeans.
df_completoK <- data.frame(completedata_rf, clasificacion2)
df_completoK <- cbind(df_completoK,dataset_mod$marca,dataset_mod$modelo)
clusterK1<-df_completoK[df_completoK$cluster_kmeans==1,]
clusterK2<-df_completoK[df_completoK$cluster_kmeans==2,]
summary(clusterK1)
summary(clusterK2)
```

```
hist(clusterK1$potencia)
hist(clusterK2$potencia)
tabla1<-table(df_completoK$`dataset_mod$marca`, df_completoK$cluster_kmeans)
tabla1
```