

# Práctica de ADEI

## Entregable 2

Alejandro Alarcón

10/19/2021

## Contents

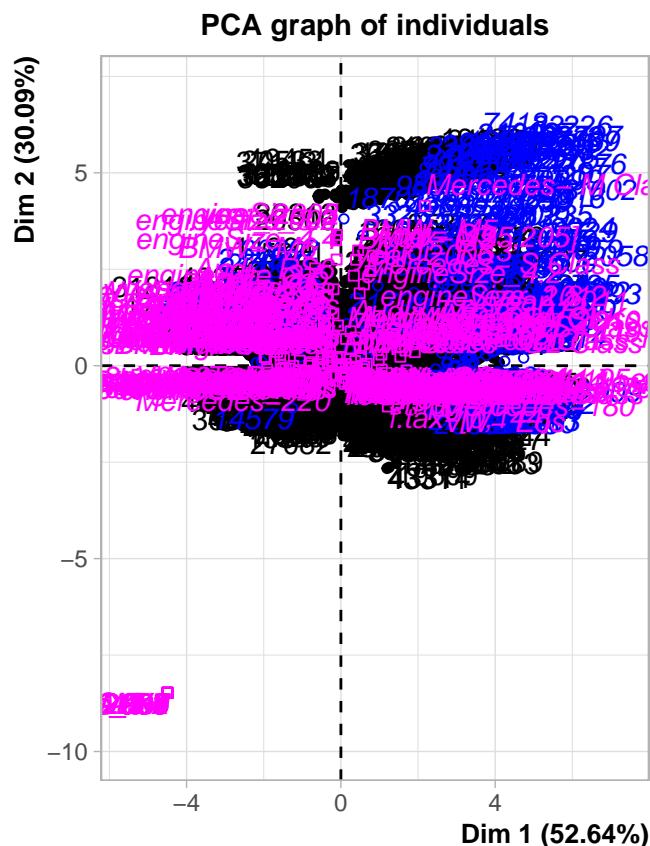
<b>1</b>	<b>Análisis de Componentes Principales</b>	<b>2</b>
1.1	Análisis según los individuos . . . . .	5
1.2	Análisis según las variables . . . . .	7
1.2.1	Variables numéricas . . . . .	7
1.2.2	Targets . . . . .	9
1.2.3	Factores . . . . .	11
<b>2</b>	<b>Hierachical Clustering</b>	<b>12</b>
2.1	Análisis según las variables . . . . .	15
2.1.1	Factores . . . . .	15
2.1.2	Variables numéricas . . . . .	16
2.2	Análisis según los inividuos . . . . .	17
<b>3</b>	<b>Correspondence Analysis</b>	<b>24</b>
<b>4</b>	<b>Multiple Correspondence Analysis</b>	<b>29</b>
4.1	Análisis segun las variables . . . . .	32
4.1.1	Factores . . . . .	32
4.1.2	Variables numéricas . . . . .	34
<b>5</b>	<b>Clustering Jerárquico desde MCA</b>	<b>35</b>
5.1	Análisis según las variables . . . . .	41
5.1.1	Factores . . . . .	41
5.1.2	Variables numéricas . . . . .	41
5.2	Análisis según las componentes del MCA . . . . .	43
5.3	Análisis según individuos . . . . .	46
<b>6</b>	<b>K-Means Clustering desde MCA</b>	<b>48</b>

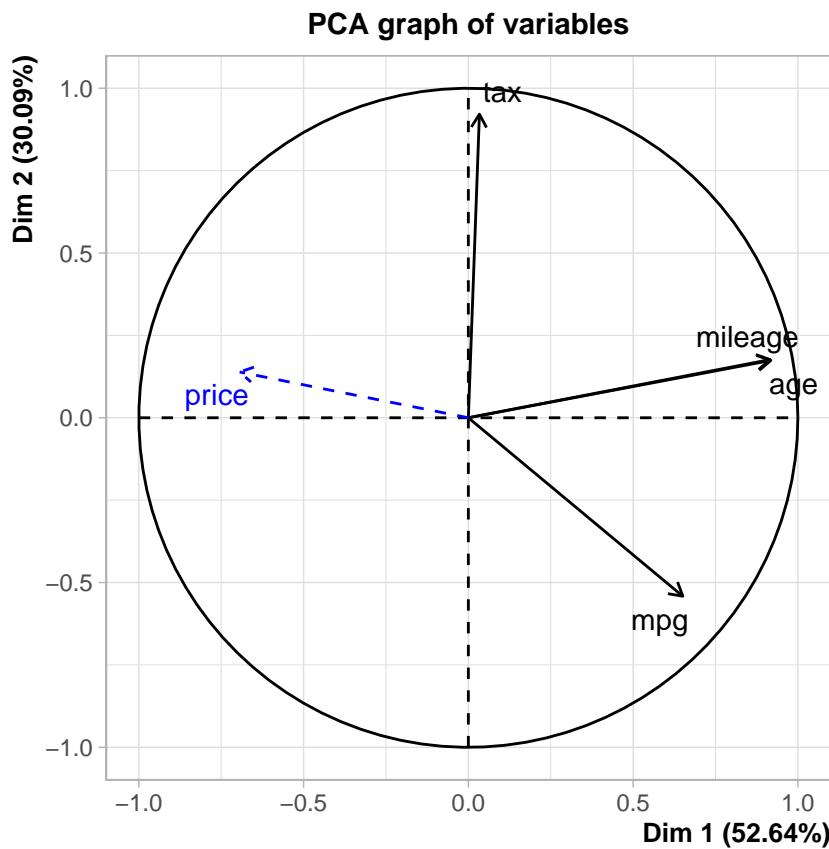
## 1 Análisis de Componentes Principales

En primer lugar, vamos a echar un vistazo a nuestras variables. Las vamos a mostrar en un orden concreto que nos va ayudar más tarde a referirnos a ellas a partir de sus índices.

Realizamos el análisis PCA de nuestro dataset, pasando como variables supplementarias cualitativas todos nuestros factores, y como variable supplementaria cuantitativa nuestro target numérico price. Además, añadimos los individuos que hemos categorizado como multivariant outliers como individuos supplementarios, para que no introduzcan ruido a la hora de calcular el PCA.

```
11 <- which( df$mout == "YesMOut")
res.pca<-PCA(df[,c(vars_res, vars_cat, vars_num)],quali.sup=c(2:13),quanti.sup= c(1), ind.sup = 11 )
```





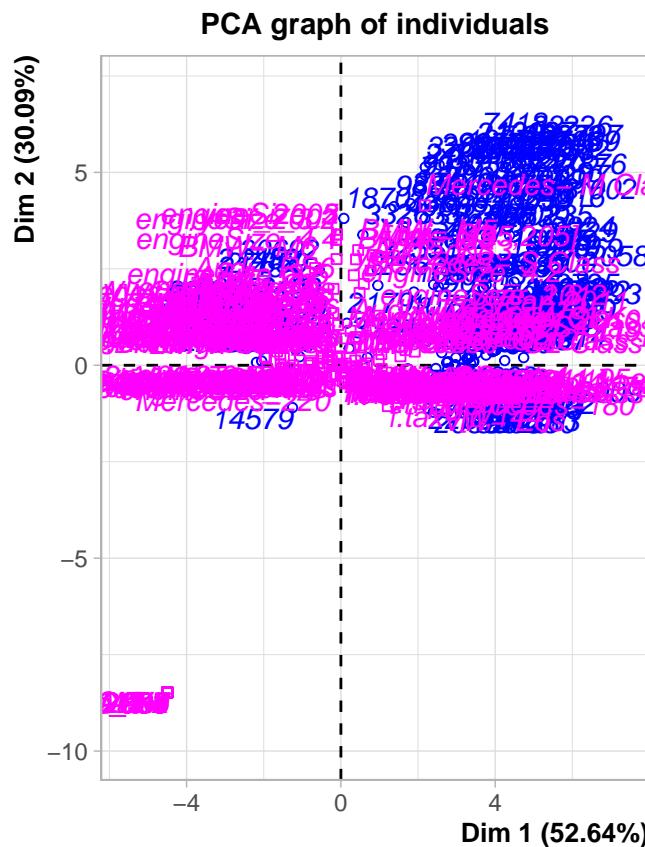
Como podemos apreciar en los gráficos previos, las dos primeras componentes principales aglutinan un 52,64% y un 30,09% de la variabilidad respectivamente.

Podemos apreciar también como la primera componente principal parece aglutinar la variabilidad de las variables mileage y age, mientras que la segunda componente principal aglutina sobre todo la variabilidad de la variable tax. Así mismo, parece que las dos componentes reflejan la variable mpg por igual, sin embargo en el caso de la Dim2 esta relación es negativa.

Además de esto, podemos ver la proyección de nuestro target numérico price en el plano formado por las dos primeras componentes principales. Se puede apreciar que price está más relacionada con la primera componente que con la segunda, pero de manera negativa.

Por último, podemos ver la proyección de los individuos en el plano formado por las dos componentes principales. Cabe destacar un pequeño grupo de individuos en la esquina inferior izquierda del plot.

```
plot.PCA(res.pca, choix=c("ind"), invisible=c("ind"))
```



```
res.pca$eig
```

	eigenvalue	percentage of variance	cumulative percentage of variance
comp 1	2.1057526	52.643814	52.64381
comp 2	1.2034959	30.087397	82.73121
comp 3	0.5101041	12.752601	95.48381
comp 4	0.1806475	4.516188	100.00000

Se puede ver que se han creado 4 componentes diferentes, cuyos Eigenvalues normalizados son 2,106, 1,203, 0,510 y 0,181 respectivamente. Si seguimos el criterio de Kaiser, como estos eigenvalues ya se han normalizado, deberíamos quedarnos con aquellos componentes con eigenvalues superiores a 1, de modo que nos quedaríamos con las dos primeras componentes.

En la siguiente salida, podemos ver que, como ya habíamos mencionado previamente, la variable tax tiene más correlación con el segundo componente que en el primero, mientras que podemos ver que pasa lo contrario con las variables mileage o age. También podemos ver que la variable price se relaciona más con la primera componente aunque lo hace de manera negativa.

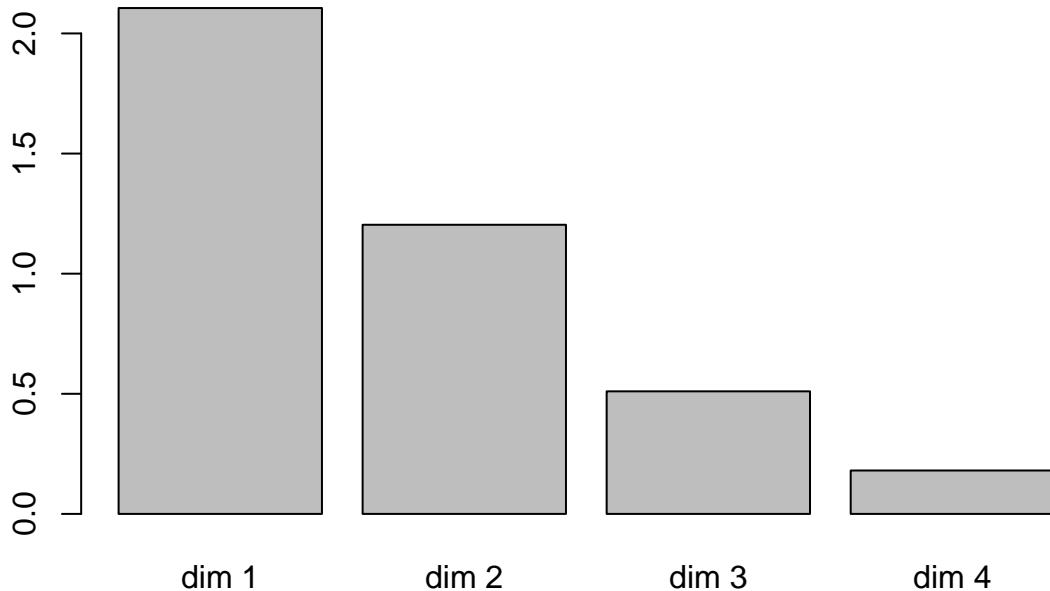
```
res.pca$var$cor
```

	Dim.1	Dim.2	Dim.3	Dim.4
mileage	0.91576938	0.1766153	-0.2012272	0.299467979
tax	0.03361629	0.9213138	0.3873597	0.001799279
mpg	0.65085499	-0.5414809	0.5321388	0.003820229
age	0.91781083	0.1740178	-0.1907681	-0.301576859

En el siguiente plot podemos ver los eigenvalues de las cuatro componentes que se han generado.

```
barplot(res.pca$eig[,1], main="Eigenvalues", names.arg= paste("dim", 1:nrow(res.pca$eig)))
```

## Eigenvalues



### 1.1 Análisis según los individuos

A continuación, vamos a proceder a analizar los individuos.

En la salida a continuación, podemos apreciar: En las dos primeras columnas, las coordenadas que reciben los individuos para las dos primeras componentes principales. En las siguientes dos columnas, los valores correspondientes al cos<sup>2</sup> para las dos dimensiones correspondientes a cada individuo. En las dos últimas columnas, la contribución que tienen los individuos en cada componente principal.

```
head(round(cbind(res.pca$ind$coord[,1:2],res.pca$ind$cos2[,1:2],res.pca$ind$contrib[,1:2]),2))
```

	Dim.1	Dim.2	Dim.1	Dim.2	Dim.1	Dim.2
1	0.01	0.14	0.00	0.08	0.00	0.00
6	1.06	-0.05	0.99	0.00	0.01	0.00
9	1.94	0.28	0.97	0.02	0.04	0.00
23	0.35	-0.32	0.40	0.34	0.00	0.00
25	0.39	4.23	0.01	0.83	0.00	0.31
38	1.08	-1.55	0.26	0.53	0.01	0.04

En la siguiente salida, podemos apreciar, de manera ordenada para la primera dimensión, los individuos que más contribución tienen hacia la primera dimensión y así como sus contribuciones para el resto de dimensiones.

```
inds <- res.pca$ind$coord  
inds <- as.data.frame(inds)  
rang.dim1<-inds[order(inds$Dim.1, decreasing = TRUE),]  
head(rang.dim1)
```

	Dim.1	Dim.2	Dim.3	Dim.4
7901	4.649442	0.7702077	-0.3464954	-0.3581472
8052	4.548182	1.3818932	-0.6501524	0.5637221
44558	4.382013	-0.2958728	-3.0658113	-0.5722813
47421	4.334346	1.5684426	-0.7322448	0.4457186
39177	4.323041	0.9430045	-0.4401120	0.1858499
21391	4.292773	-0.6679742	-2.4828319	-0.9544709

Vamos a proceder a hacer lo mismo para la segunda dimensión.

```
rang.dim2<-inds[order(inds$Dim.2, decreasing = TRUE),]
head(rang.dim2)
```

	Dim.1	Dim.2	Dim.3	Dim.4
19126	2.590791	5.346487	0.7708686	-0.8296873
31585	2.755137	5.318110	0.7948852	0.8235767
32440	3.160048	5.249722	0.1156681	-0.8583244
45549	3.020479	5.220930	0.9387187	0.2231532
33247	2.386721	5.152644	0.2872545	0.6492749
34460	1.309632	5.136760	1.1428969	0.1245982

Como se puede apreciar en las salidas anteriores, podemos ver como parece haber elementos con más contribución en la segunda dimensión que en la primera.

A continuación, podemos ver todas las variables de los 10 primeros individuos que más aportan a la primera componente principal:

```
df[which(row.names(df) %in% row.names(res.pca$ind$coord[row.names(rang.dim1)[1:10],])),]
```

	model	year	price	transmission	mileage	fuelType	tax
7901	Audi- A3	2011	6500	f.Trans-Manual	74000.00	f.Fuel-Diesel	153.0634
8052	Audi- A4	2012	7990	f.Trans-Manual	88000.00	f.Fuel-Diesel	157.2018
18410	BMW- 3 Series	2012	7490	f.Trans-Manual	81000.00	f.Fuel-Diesel	157.1293
20905	BMW- 3 Series	2011	4990	f.Trans-Manual	67682.53	f.Fuel-Diesel	155.9702
21391	BMW- 1 Series	2010	5990	f.Trans-Manual	69000.00	f.Fuel-Diesel	125.0000
39177	VW- Golf	2012	6499	f.Trans-Manual	78211.00	f.Fuel-Diesel	154.1150
40937	VW- Passat	2010	5995	f.Trans-Manual	75374.00	f.Fuel-Diesel	160.0000
44558	VW- Polo	2010	4250	f.Trans-Manual	79000.00	f.Fuel-Petrol	125.0000
44562	VW- Polo	2011	3980	f.Trans-Manual	86000.00	f.Fuel-Petrol	125.0000
47421	VW- Up	2012	4495	f.Trans-Manual	85000.00	f.Fuel-Petrol	158.4613
	mpg	engineSize	manufacturer	age	outs	f.miles	f.tax
7901	68.9	1.6	Audi	10	2 f.miles-(34.1,119]	f.tax-(145,155]	
8052	62.8	2	Audi	9	2 f.miles-(34.1,119]	f.tax-(155,205]	
18410	61.4	2	BMW	9	2 f.miles-(34.1,119]	f.tax-(155,205]	
20905	62.8	2	BMW	10	3 f.miles-(34.1,119]	f.tax-(155,205]	
21391	57.6	2	BMW	11	1 f.miles-(34.1,119]	f.tax-[0,144]	
39177	65.7	2	VW	9	2 f.miles-(34.1,119]	f.tax-(145,155]	
40937	50.4	2	VW	11	1 f.miles-(34.1,119]	f.tax-(155,205]	
44558	51.4	1.2	VW	11	2 f.miles-(34.1,119]	f.tax-[0,144]	
44562	51.4	1.2	VW	10	2 f.miles-(34.1,119]	f.tax-[0,144]	
47421	60.1	1	VW	9	2 f.miles-(34.1,119]	f.tax-(155,205]	
	f.mpg	f.age	Audi	mout	aux		
7901	f.mpg-alto	f.age-(+4)	Audi	Yes	NoMOut	[899,1.1e+04]	
8052	f.mpg-alto	f.age-(+4)	Audi	Yes	NoMOut	[899,1.1e+04]	
18410	f.mpg-medio	f.age-(+4)	Audi	No	NoMOut	[899,1.1e+04]	
20905	f.mpg-alto	f.age-(+4)	Audi	No	NoMOut	[899,1.1e+04]	
21391	f.mpg-medio	f.age-(+4)	Audi	No	NoMOut	[899,1.1e+04]	
39177	f.mpg-alto	f.age-(+4)	Audi	No	NoMOut	[899,1.1e+04]	
40937	f.mpg-bajo	f.age-(+4)	Audi	No	NoMOut	[899,1.1e+04]	
44558	f.mpg-bajo	f.age-(+4)	Audi	No	NoMOut	[899,1.1e+04]	
44562	f.mpg-bajo	f.age-(+4)	Audi	No	NoMOut	[899,1.1e+04]	
47421	f.mpg-medio	f.age-(+4)	Audi	No	NoMOut	[899,1.1e+04]	
	f.price						
7901	f.price-[899,1.1e+04]						
8052	f.price-[899,1.1e+04]						
18410	f.price-[899,1.1e+04]						
20905	f.price-[899,1.1e+04]						
21391	f.price-[899,1.1e+04]						
39177	f.price-[899,1.1e+04]						
40937	f.price-[899,1.1e+04]						
44558	f.price-[899,1.1e+04]						

```
44562 f.price-[899,1.1e+04]
47421 f.price-[899,1.1e+04]
```

Como se puede apreciar, la mayoría tienen valores altos para las variables age y mileage, que son las que más se relacionan con la primera componente. Contrariamente, por norma general, estos vehículos también tienen precios menores, cosa que era esperable si tenemos en cuenta que la proyección de la variable price en el plano formado por las dos componentes principales tiene sentido negativo.

## 1.2 Análisis según las variables

### 1.2.1 Variables numéricas

Desde el punto de vista de las variables, en la salida a continuación, podemos ver los valores correspondientes al cos2 para cada una de las componentes principales en las dos primeras columnas, así como las contribuciones que tienen hacia estas en las dos últimas. Cabe destacar las contribuciones de las variables age y mileage para la primera dimensión.

```
round(cbind(res.pca$var$cos2[,1:2],res.pca$var$contrib[,1:2]),2)
```

	Dim.1	Dim.2	Dim.1	Dim.2
mileage	0.84	0.03	39.83	2.59
tax	0.00	0.85	0.05	70.53
mpg	0.42	0.29	20.12	24.36
age	0.84	0.03	40.00	2.52

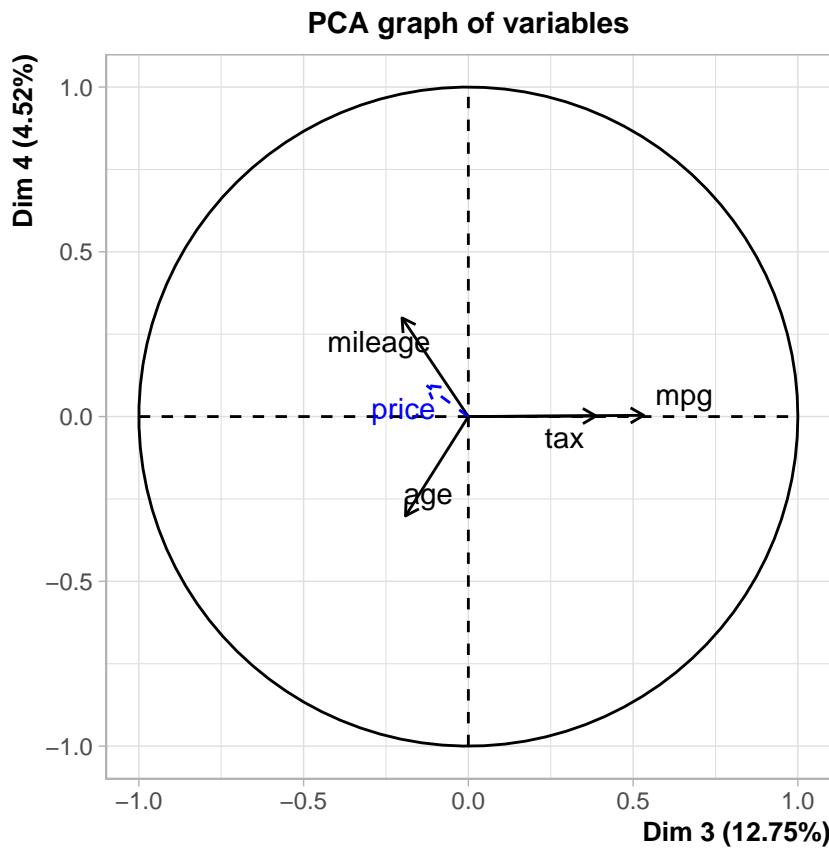
A continuación podemos ver las correlaciones de todas nuestras variables numéricas con la primera componente principal. Cabe destacar las fuertes correlaciones con age y mileage, como habíamos mencionado previamente.

```
res.des<-dimdesc(res.pca)
res.des$Dim.1$quanti
```

	correlation	p.value
age	0.91781083	0.00000000
mileage	0.91576938	0.00000000
mpg	0.65085499	0.00000000
tax	0.03361629	0.02039052
price	-0.69407066	0.00000000

Por último, y aunque hemos afirmado que, según el criterio de Kaiser, solo eran necesarias 2 componentes principales, vamos a analizar graficamente las componentes 3 y 4. Podemos apreciar como las variables tax y mpg tienen gran importancia en la tercera componente, mientras que Para la cuarta tienen más importancia mileage y age.

```
plot.PCA(res.pca, choix=c("var"), axes=c(3,4))
```



En el siguiente gráfico, podemos ver gráficamente y de manera resumida la importancia que tienen cada una de las variables numéricas en las diferentes componentes principales.

```
library("corrplot")
```

```
corrplot 0.90 loaded
```

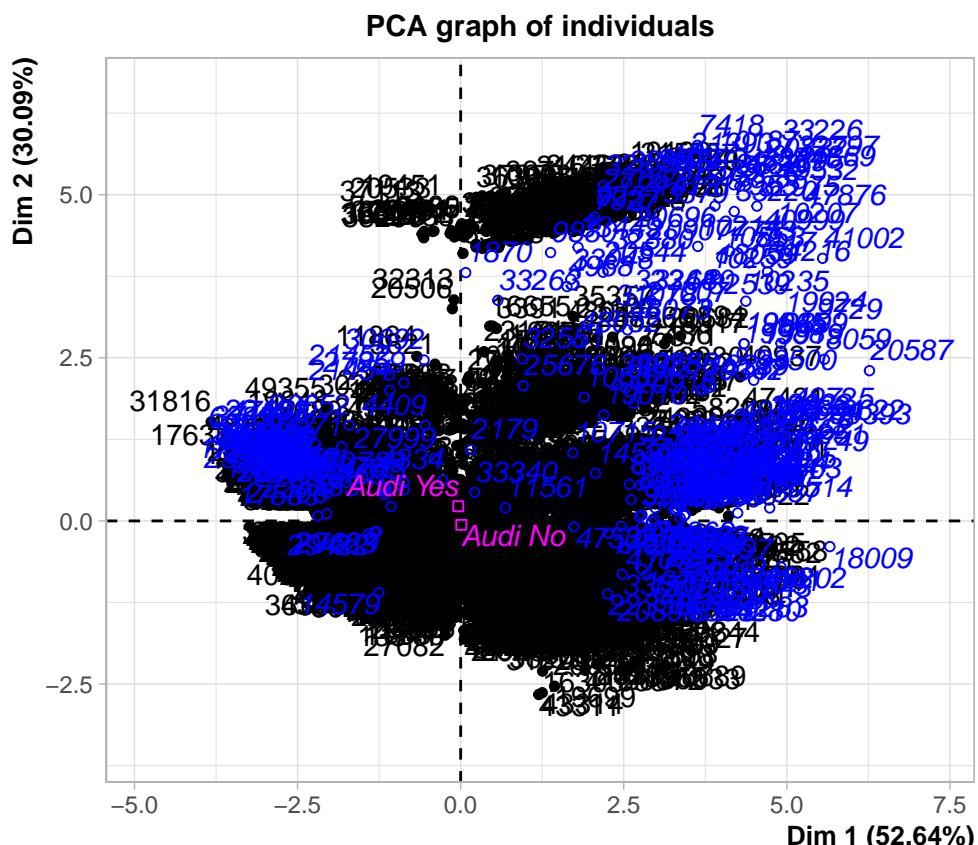
```
corrplot(res.pca$var$cos2, is.corr=FALSE)
```

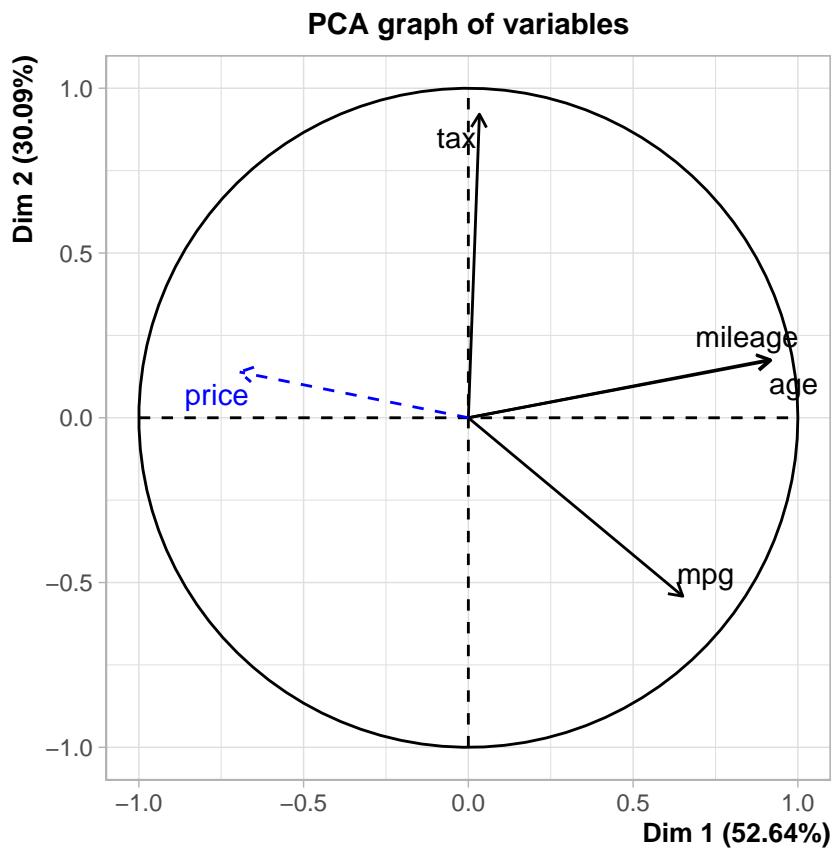


### 1.2.2 Targets

En el siguiente plot, podemos apreciar como nuestro target categórico AUDI, tiene más relación con la segunda dimensión que con la primera.

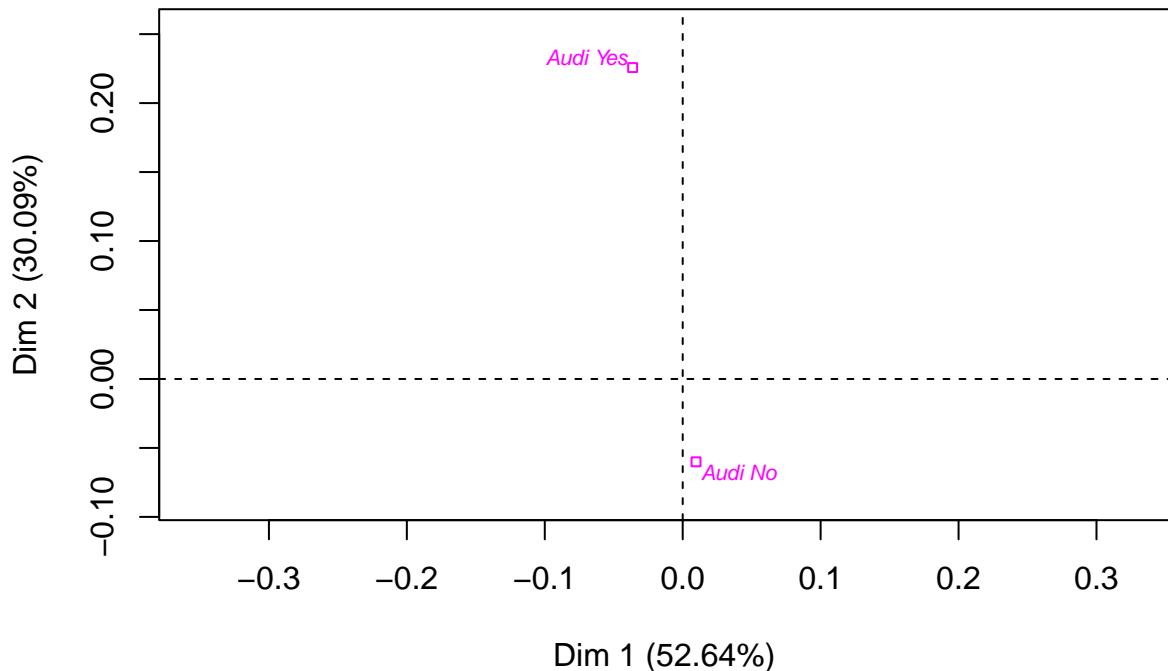
```
11 <- which( df$mout == "YesMOut")
res.pca<-PCA(df[,c(vars_res, vars_num)],quali.sup=c(2),quanti.sup= c(1), ind.sup = 11, ncp=2)
```





```
plot(res.pca, choix="ind", invisible=c("ind", "ind.sup"), cex=0.7, graph.type = "classic")
```

### PCA graph of individuals



Podemos ver como la componente más representativa de este target es la segunda, ya que las coordenadas que aparecen vienen dadas por la correlación.

```
res.pca$quali.sup$coord
```

Dim.1	Dim.2
-------	-------

```
Audi No  0.009658582 -0.06005841
Audi Yes -0.036306609  0.22575955
```

### 1.2.3 Factores

A continuación podemos ver los coeficientes R-squared que aparecen para nuestros factores: Cabe destacar la variabilidad de la componente que viene explicada por la variabilidad de factores como year o f.miles. Si lo analizamos, nos daremos cuenta de que estos factores se crearon en la entrega anterior a partir de las variables mileage y age, de modo que es coherente que los estos factores derivados también tengan una alta explicabilidad de la varianza de la primera componente.

```
res.des$Dim.1$quali
```

	R2	p.value
year	0.85280789	0.000000e+00
f.price	0.53679470	0.000000e+00
f.miles	0.78130507	0.000000e+00
f.mpg	0.39449527	0.000000e+00
f.tax	0.34717517	0.000000e+00
f.age	0.75695089	0.000000e+00
engineSize	0.11832840	1.422077e-114
fuelType	0.08668141	2.289915e-94
model	0.12058632	5.267522e-80
transmission	0.05648609	8.942794e-61

Para la segunda componente, podemos ver como los factores más relevantes son f.tax y f.mpg.

```
res.des$Dim.2$quali
```

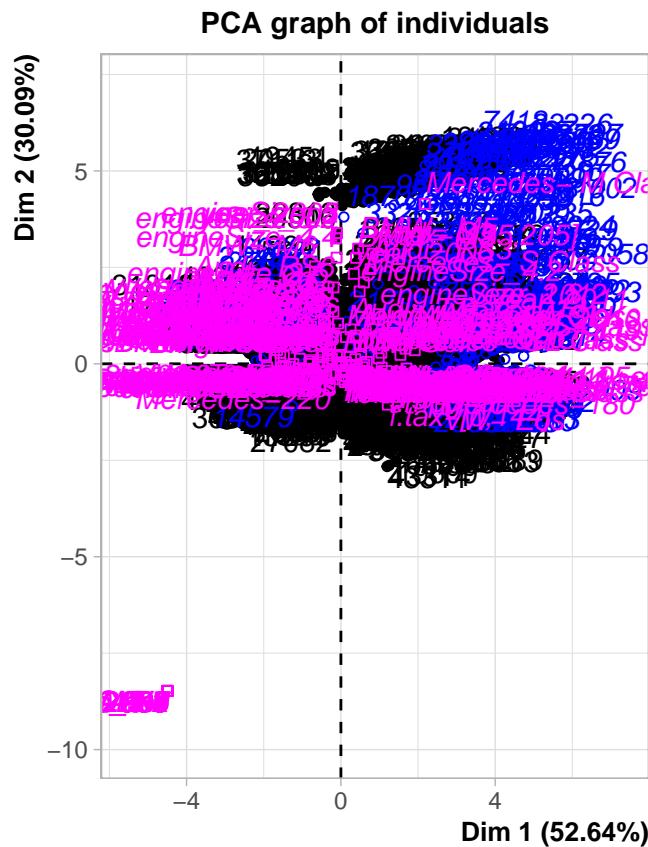
	R2	p.value
f.tax	0.72201517	0.000000e+00
f.mpg	0.26790289	2.924869e-321
engineSize	0.16780404	1.019814e-172
model	0.19181933	1.693427e-157
year	0.07154553	2.153404e-67
fuelType	0.03441978	6.710061e-37
f.miles	0.03029059	1.677088e-31
f.price	0.02895891	5.907027e-27
f.age	0.02364062	1.958276e-25
transmission	0.02351966	2.629124e-25
manufacturer	0.01523235	9.745343e-16
Audi	0.01126614	2.118648e-13

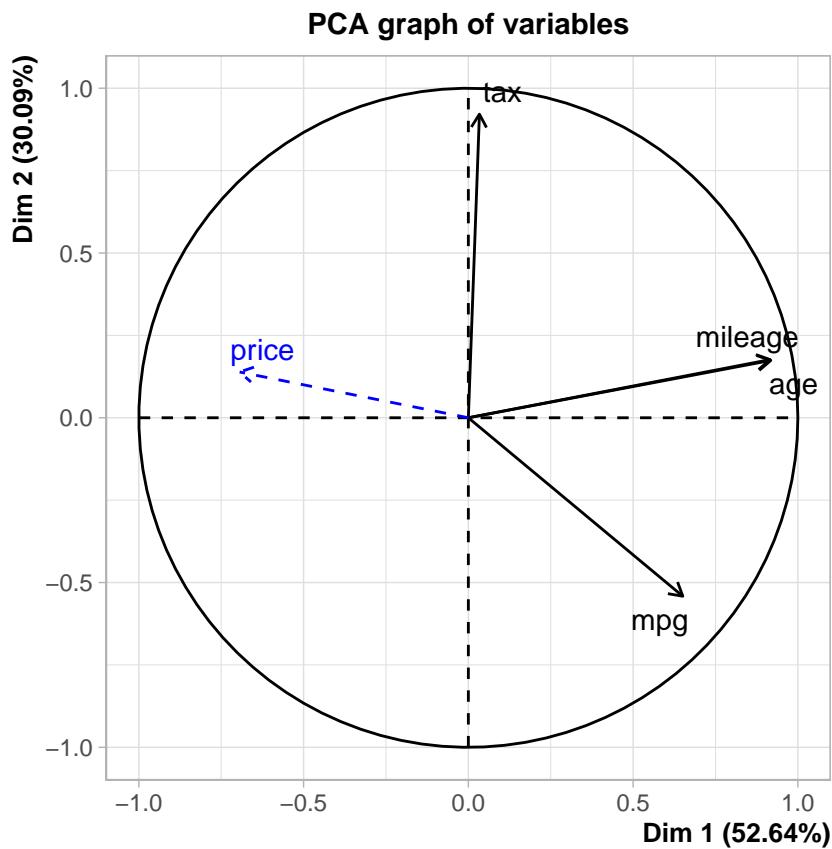
## 2 Hierarchical Clustering

A continuación, y para ser prácticos, vamos a proceder a realizar el proceso de clustering jerárquico, a partir del cual vamos a determinar (o al menos aproximar) el número óptimo de clusters para ejecutar el clustering con K-means.

En primer lugar, vamos a volver a ejecutar el PCA con las variables categoricas como suplementarias y quedándonos solo con las dos primeras componentes principales, como se ha determinado anteriormente con el criterio de Kaiser para el ACP.

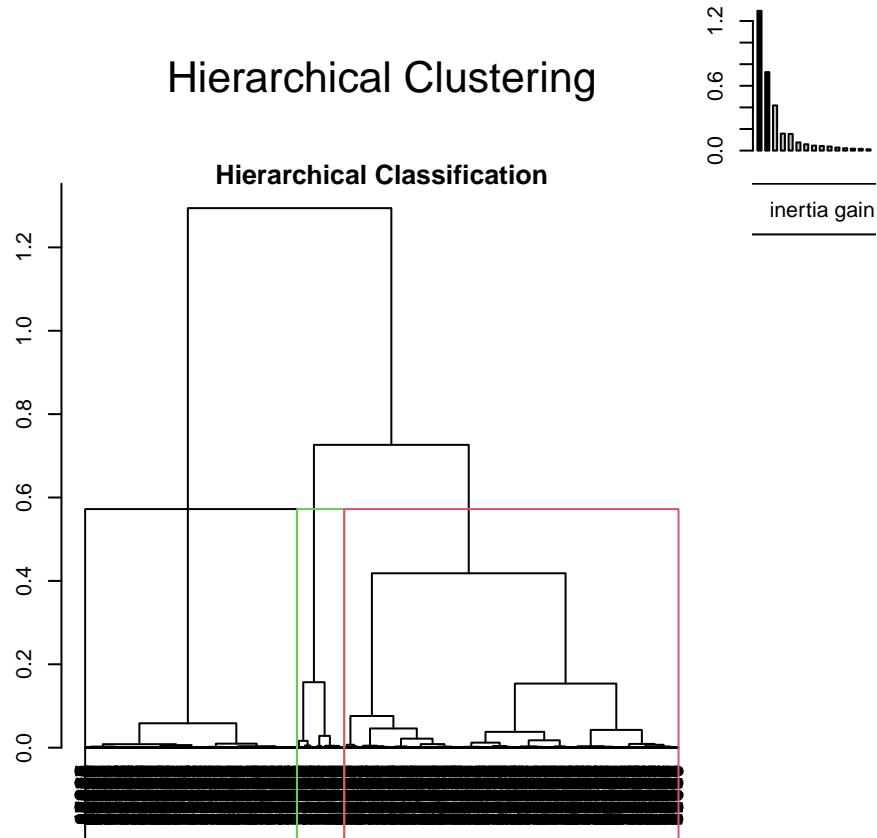
```
res.pca<-PCA(df[,c(vars_res, vars_cat, vars_num)],quali.sup=c(2:13),quanti.sup= c(1), ind.sup = 11, ncp
```



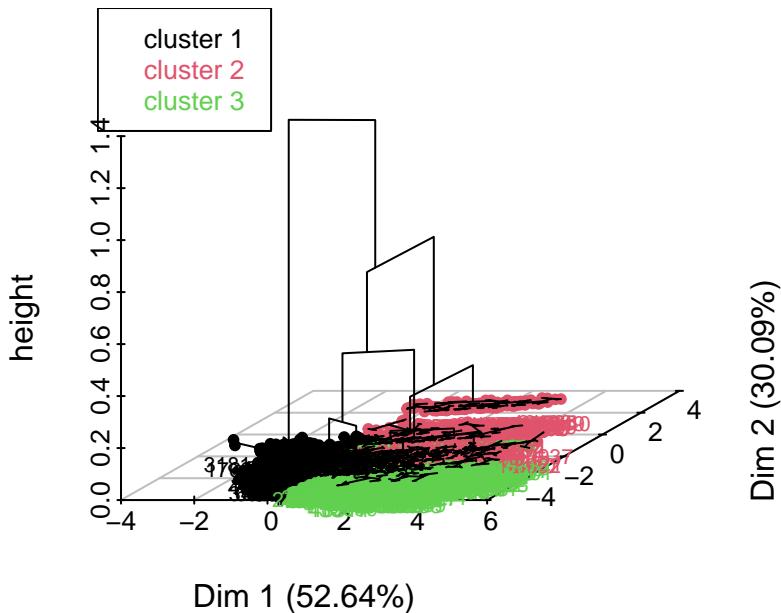


A continuación ejecutaremos el HCPC a partir del ACP anterior. Con el parámetro `nb.clust=-1` indicamos al sistema que tome el número óptimo de clusters según la partición con la que el decreto relativo de inercia es más alto. (Según la documentación -  $(i(\text{clusters } n+1)/i(\text{cluster } n))$  ). Podemos ver que para este caso, se ha seleccionado como óptima una partición de 3 clusters.

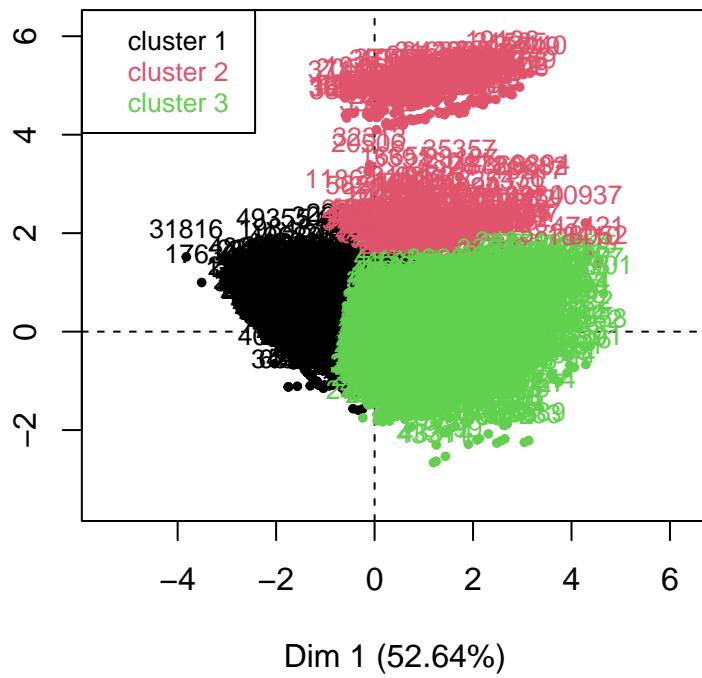
```
res.hcpc<-HCPC(res.pca, order=TRUE, nb.clust=-1)
```



## Hierarchical clustering on the factor map



## Factor map



Si aplicamos el criterio de Kaiser, podemos ver que dos componentes son suficientes.

```
length(which(res.hcpc$call$t$res$eig[,1] > mean(res.hcpc$call$t$res$eig[,1])))
```

[1] 2

Según los dos criterios aplicados (el que aplica el sistema con el parámetro `nb.clust=-1` y el de Kaiser) obtenemos que el número óptimo de componentes es 2 o 3. Procederemos con 3 componentes ya que así aglomeramos mayor variabilidad.

Ejecutando el siguiente comando, podemos ver como se relacionan las dos componentes principales a partir de las cuales hemos generado la clusterización, con los diferentes clusters que se han generado. Se puede apreciar como para el cluster 1, la coordenada de la primera componente principal es significativamente más baja que para el conjunto del datafrfame. En el caso del cluster 2, ambas componentes principales tienen un valor mayor. Por último, para el cluster 3, la primera componente tiene un valor mayor mientras que la segunda tiene un valor menor.

```
res.hcpc$desc.axes

Link between the cluster variable and the quantitative variables
=====
      Eta2 P-value
Dim.1 0.7193237      0
Dim.2 0.6160798      0

Description of each cluster by quantitative variables
=====
$'1'
      v.test Mean in category Overall mean sd in category Overall sd
Dim.1 -58.46739       -1.2215 -1.300956e-14      0.600699  1.451121
      p.value
Dim.1      0

$'2'
      v.test Mean in category Overall mean sd in category Overall sd
Dim.2 52.15160        2.912518 4.835629e-13      1.4447218 1.097040
Dim.1 14.82471        1.095138 -1.300956e-14     0.9535762  1.451121
      p.value
Dim.2 0.000000e+00
Dim.1 1.01412e-49

$'3'
      v.test Mean in category Overall mean sd in category Overall sd
Dim.1 51.29163        1.2640831 -1.300956e-14     0.8996346  1.451121
Dim.2 -26.78506       -0.4990457 4.835629e-13      0.6760169  1.097040
      p.value
Dim.1 0.000000e+00
Dim.2 4.824958e-158
```

## 2.1 Análisis según las variables

A continuación vamos a ver como se relacionan las variables originales del dataframe con los clusters que se han generado.

En primer lugar, podemos ver el número de individuos que se han asignado a cada cluster.

```
summary(res.hcpc$data.clust$clust)
```

```
 1   2   3
2396 357 2006
```

### 2.1.1 Factores

A partir del test de chi2, se puede determinar que factores diferencian los clusters que se han generado.

```
res.hcpc$desc.var$test.chi2
```

	p.value	df
year	0.000000e+00	26
f.price	0.000000e+00	14
f.miles	0.000000e+00	6

```

f.mpg      0.000000e+00  6
f.tax      0.000000e+00  6
f.age      0.000000e+00  4
engineSize 1.826997e-254 38
model     8.503636e-168 168
transmission 1.042687e-68 4
fuelType   2.390256e-50  4
manufacturer 3.494463e-09 6
Audi       1.351598e-04  2

```

Si profundizamos un poco mas en esto, podemos ver como caracterizan los valores de las variables cualitativas los distintos clusters que se han generado.

<He omitido la salida porque ocupa mucho, pero abajo podemos ver algunas de las conclusiones>

```
#res.hcpc$desc.var$category
```

Podemos destacar, por ejemplo la acumulación de coches nuevos y con poco kilometraje en el primer cluster(f.age=f.age-[1,2] -> Mod/Cla 76.0016694)

### 2.1.2 Variables numéricas

Con el test eta-squared, podemos determinar qué variables numéricas han sido influyentes a la hora de generar la clusterización.

```
res.hcpc$desc.var$quanti.var
```

	Eta2	P-value
price	0.3801363	0
mileage	0.5594030	0
tax	0.5785314	0
mpg	0.4666499	0
age	0.6587494	0

Podemos ver que las variables más representativas son age, mileage y tax, que son las mismas que aparecen como más representativas a la hora de realizar el PCA.

Por ultimo, podemos analizar como estas variables numéricas caracterizan los distintos clusters.

```
res.hcpc$desc.var$quanti
```

	v.test	Mean in category	Overall mean	sd in category	Overall sd
price	41.470932	26465.244157	21027.36730	8810.708158	9107.713091
tax	-3.648855	145.772213	146.39236	2.454538	11.804945
mpg	-38.509985	47.337173	53.31998	8.854711	10.790845
mileage	-51.308451	7750.667311	21187.33402	6819.801822	18189.736407
age	-55.506058	2.206459	3.68627	0.787568	1.851781
	p.value				
price	0.0000000000				
tax	0.0002634117				
mpg	0.0000000000				
mileage	0.0000000000				
age	0.0000000000				
	\$'1'				
	v.test	Mean in category	Overall mean	sd in category	Overall sd
tax	51.18692	177.153448	146.39236	19.720384	11.804945
age	22.91810	5.846734	3.68627	1.336169	1.851781
mileage	19.87958	39595.576244	21187.33402	15971.866217	18189.736407
price	-2.86210	19700.361345	21027.36730	6382.111252	9107.713091
mpg	-14.96974	45.096639	53.31998	5.607637	10.790845
	\$'2'				

```

          p.value
tax      0.000000e+00
age     3.066711e-116
mileage 6.114252e-88
price    4.208438e-03
mpg     1.157770e-50

$'3'
          v.test Mean in category Overall mean sd in category Overall sd
mpg      46.97796       61.929422      53.31998      6.934705      10.790845
age      43.97582       5.069292      3.68627      1.310546      1.851781
mileage 41.34652      33960.270695   21187.33402   15649.023378  18189.736407
tax     -23.61090      141.658651     146.39236      7.933143      11.804945
price   -40.46367      14768.438185   21027.36730   4691.847733   9107.713091

          p.value
mpg      0.000000e+00
age      0.000000e+00
mileage 0.000000e+00
tax     2.978511e-123
price   0.000000e+00

```

Podemos destacar, por ejemplo, que para el primer cluster, los vehículos son más nuevos y tienen menor kilometraje y consumo, pero tambien son más caros.

## 2.2 Análisis según los individuos

Vamos a analizar los parámetros de cada cluster que se ha derivado a partir del clustering jerárquico.

```
res.hcpc$desc.ind$para
```

```

Cluster: 1
  30871      24055      28264      48543      34122
0.02696516 0.03406107 0.03530562 0.04094043 0.04278051
-----
Cluster: 2
  9872      9145      10617      14937      48041
0.3342379 0.3467858 0.4207337 0.5280356 0.5314977
-----
Cluster: 3
  20977      48828      19474      27898      48747
0.02576784 0.03224691 0.03338568 0.03664690 0.04303244

```

Si analizamos los parámetros del primer cluster, podemos ver como aparecen coches nuevos de gasolina. Todos tienen kilometrajes e impuestos similares.

```
summary(df[c("30871", "24055", "28264", "48543", "34122")])
```

	model	year	price	transmission	
Mercedes- C Class:3	2019 :5	Min. :20149	f.Trans-Manual :0		
Mercedes- B Class:1	2001 :0	1st Qu.:21890	f.Trans-SemiAuto :3		
VW- Arteon	:1 2002 :0	Median :25299	f.Trans-Automatic:2		
Audi- A1	:0 2003 :0	Mean :24357			
Audi- A3	:0 2004 :0	3rd Qu.:26056			
Audi- A4	:0 2005 :0	Max. :28391			
(Other)	:0 (Other):0				
	mileage	fuelType	tax	mpg	engineSize
Min. :10061	f.Fuel-Diesel:0	Min. :145	Min. :45.60	1.5 :3	
1st Qu.:10096	f.Fuel-Petrol:5	1st Qu.:145	1st Qu.:46.30	1.3 :1	
Median :10682	f.Fuel-Hybrid:0	Median :145	Median :46.30	2 :1	
Mean :10883		Mean :145	Mean :46.32	1 :0	
3rd Qu.:11785		3rd Qu.:145	3rd Qu.:46.30	1.2 :0	

```

Max.    :11789                         Max.    :145   Max.    :47.10  1.4    :0
                                         (Other):0

  manufacturer      age       outs          f.miles
Audi      :0     Min.    :2     Min.    :0   f.miles-[0.001,5.81]:0
BMW       :0     1st Qu.:2     1st Qu.:0   f.miles-(5.81,17.7] :5
Mercedes:4     Median :2     Median :0   f.miles-(17.7,34.1] :0
VW        :1     Mean    :2     Mean    :0   f.miles-(34.1,119] :0
                           3rd Qu.:2     3rd Qu.:0
                           Max.    :2     Max.    :0

  f.tax           f.mpg          f.age        Audi
f.tax-[0,144]   :0   f.mpg-muy bajo:1   f.age-[1,2]:5   Audi No :5
f.tax-(144,145]:5   f.mpg-bajo    :4   f.age-(2,4]:0   Audi Yes:0
f.tax-(145,155]:0   f.mpg-medio  :0   f.age-(+4) :0
f.tax-(155,205]:0   f.mpg-alto   :0

  mout          aux          f.price
NoMOut :5   (1.95e+04,2.2e+04]:2   f.price-(1.95e+04,2.2e+04]:2
YesMOut:0   (2.6e+04,3.15e+04]:2   f.price-(2.6e+04,3.15e+04]:2
             (2.2e+04,2.6e+04] :1   f.price-(2.2e+04,2.6e+04] :1
             [899,1.1e+04]   :0   f.price-[899,1.1e+04]   :0
             (1.1e+04,1.4e+04] :0   f.price-(1.1e+04,1.4e+04] :0
             (1.4e+04,1.7e+04] :0   f.price-(1.4e+04,1.7e+04] :0
             (Other)         :0   (Other)         :0

```

Por último, echaremos un vistazo a los individuos más típicos de los clusters:

```
res.hcpc$desc.ind$dist
```

```
Cluster: 1
  31816    17634    17208    48339    24771
5.112322 4.988518 4.128977 4.117106 4.094749
```

```
Cluster: 2
  32440    31585    19126    45549    33247
6.053347 6.005210 5.994198 5.983564 5.762110
```

```
Cluster: 3
  38689    35633    20844    21391    48944
4.876394 4.813790 4.803596 4.800500 4.739948
```

Si analizamos el primer cluster, podemos ver que todos son del 2020 y en su mayoría semi-automáticos. Además, tienen kilómetros y consumos muy bajos. Todos están en el rango más alto de precio.

```
summary(df[c("31816", "17634", "17208", "48339", "24771"),])
```

	model	year	price	transmission	
BMW- X3	:1	2020	:5	Min.   :33900   f.Trans-Manual   :0	
BMW- X4	:1	2001	:0	1st Qu.:43995   f.Trans-SemiAuto :4	
Mercedes- A Class	:1	2002	:0	Median :49980   f.Trans-Automatic:1	
Mercedes- GLC Class	:1	2003	:0	Mean   :47473	
VW- Touareg	:1	2004	:0	3rd Qu.:52991	
Audi- A1	:0	2005	:0	Max.   :56499	
(Other)	:0	(Other):0			
	mileage	fuelType	tax	mpg	engineSize
Min. :	345	f.Fuel-Diesel:0	Min.   :135	Min.   : 1.10	3      :3
1st Qu.:	2000	f.Fuel-Petrol:3	1st Qu.:140	1st Qu.: 5.50	1.3    :1
Median :	3999	f.Fuel-Hybrid:2	Median :145	Median :25.50	2      :1
Mean   :	3141		Mean   :143	Mean   :17.08	1      :0
3rd Qu.:	4360		3rd Qu.:145	3rd Qu.:26.40	1.2    :0

```

Max.    :5000          Max.    :150   Max.    :26.90   1.4    :0
                                         (Other):0
  manufacturer      age       outs           f.miles
Audi      :0     Min.    :1     Min.    :1.0   f.miles-[0.001,5.81]:5
BMW       :2     1st Qu.:1    1st Qu.:1.0   f.miles-(5.81,17.7] :0
Mercedes:2   Median :1    Median :1.0   f.miles-(17.7,34.1] :0
VW        :1     Mean    :1     Mean    :1.2   f.miles-(34.1,119] :0
                           3rd Qu.:1    3rd Qu.:1.0
                           Max.    :1     Max.    :2.0

f.tax           f.mpg       f.age      Audi
f.tax-[0,144] :2  f.mpg-muy bajo:5  f.age-[1,2]:5  Audi No :5
f.tax-(144,145]:2  f.mpg-bajo    :0  f.age-(2,4]:0  Audi Yes:0
f.tax-(145,155]:1  f.mpg-medio  :0  f.age-(+4) :0
f.tax-(155,205]:0  f.mpg-alto   :0

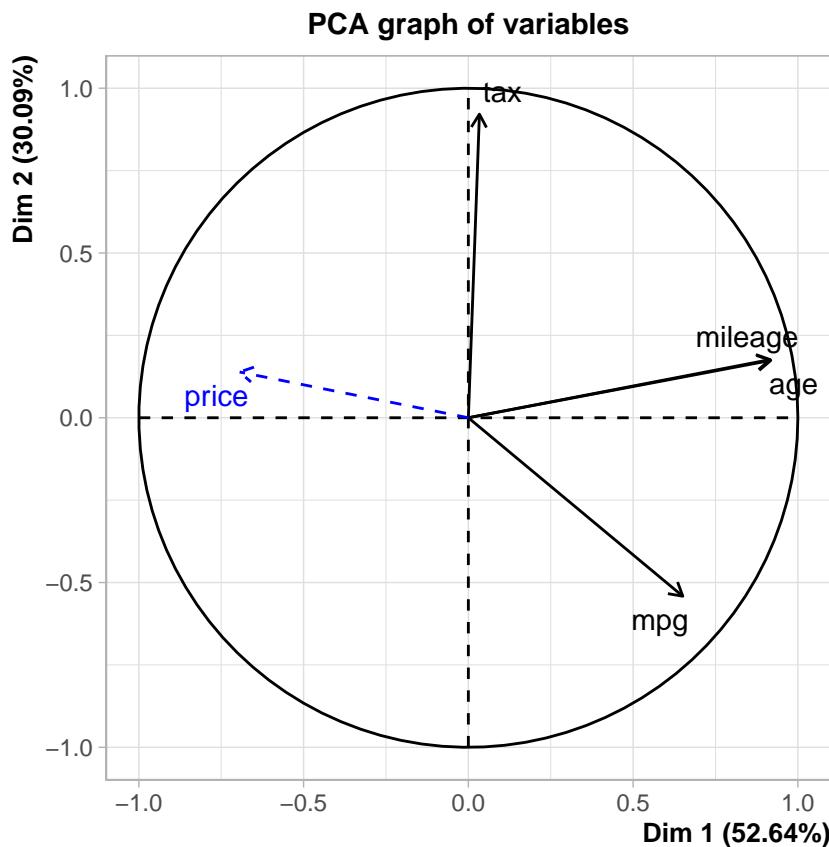
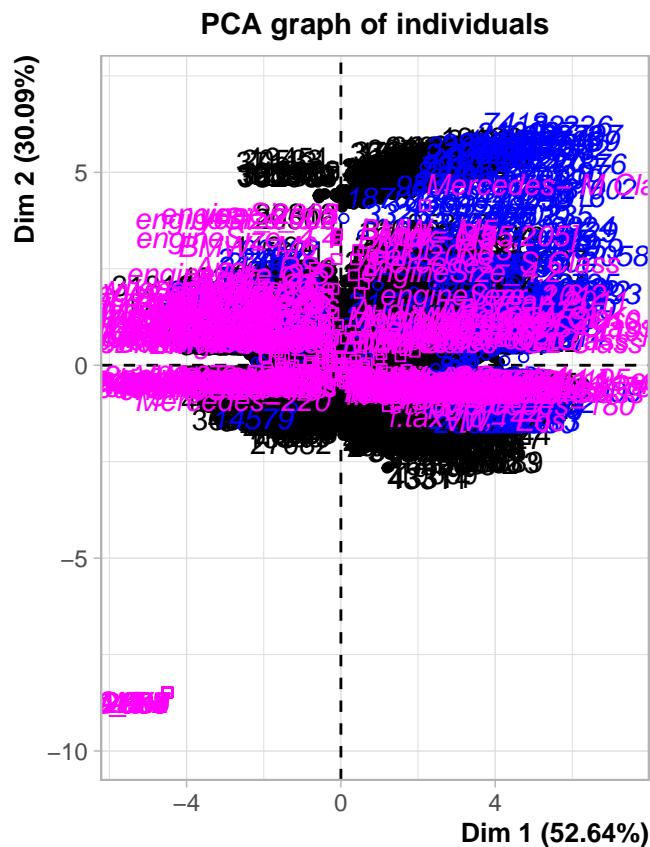
mout          aux           f.price
NoMOut :5   (3.15e+04,1.09e+05]:5  f.price-(3.15e+04,1.09e+05]:5
YesMOut:0   [899,1.1e+04]      :0  f.price-[899,1.1e+04]      :0
            (1.1e+04,1.4e+04]    :0  f.price-(1.1e+04,1.4e+04]    :0
            (1.4e+04,1.7e+04]    :0  f.price-(1.4e+04,1.7e+04]    :0
            (1.7e+04,1.95e+04]   :0  f.price-(1.7e+04,1.95e+04]   :0
            (1.95e+04,2.2e+04]   :0  f.price-(1.95e+04,2.2e+04]   :0
            (Other)             :0  (Other)             :0

```

Podemos ver amplias diferencias entre los individuos típicos de los clusters y los paragons.

# K-means Clustering desde ACP

```
res.pca<-PCA(df[,c(vars_res, vars_cat, vars_num)], quali.sup=c(2:13), quanti.sup= c(1), ind.sup = 11, ncp=
```



```
ppcc <- res.pca$ind$coord
kc <- kmeans (dist(ppcc), 3)
df[-11, "claKMPCA"] <- kc$cluster
```

Podemos ver como dentro de nuestra variable kc, se guardan datos como el cluster al que se asigna cada elemento, las distancias en el interior de los clusters o las distancias entre clusters.

```
summary(kc)
```

	Length	Class	Mode
cluster	4759	-none-	numeric
centers	14277	-none-	numeric
totss	1	-none-	numeric
withinss	3	-none-	numeric
tot.withinss	1	-none-	numeric
betweenss	1	-none-	numeric
size	3	-none-	numeric
iter	1	-none-	numeric
ifault	1	-none-	numeric

Estos valores nos ayudan a determinar la calidad de la cluserización, ya que idealmente, queremos clusters con elementos muy juntos entre si y mucha diferenciación entre clusters. Para var la calidad de la clusterización, vamos a realizar el cociente de las distancias entre clusters entre la suma de todas las distancias.

```
kc$betweenss/kc$totss
```

```
[1] 0.7050496
```

Podemos ver que la suma de las distancias entre los clusters suma un 70% del total.

Por otro lado, si comprobamos la suma de las distancias dentro de los clusters, podemos ver como estas tan solo suman el 30% del total.

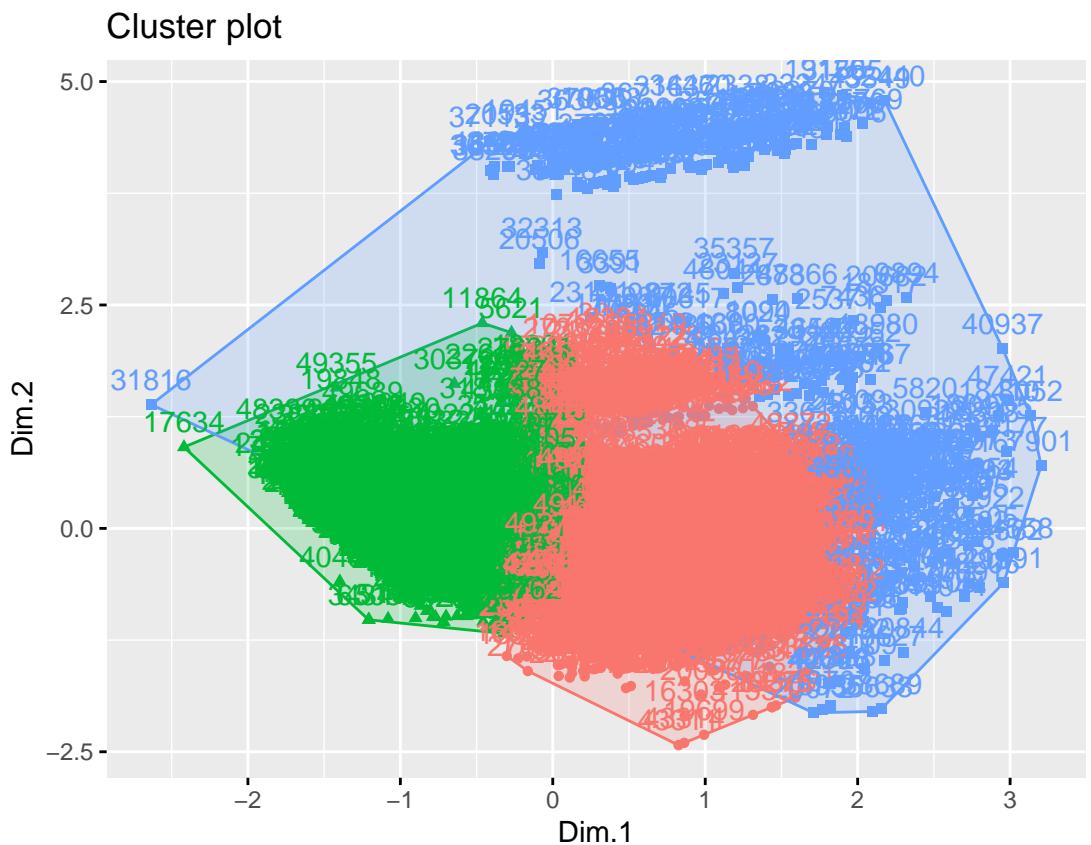
```
kc$tot.withinss/kc$totss
```

```
[1] 0.2949504
```

De este modo, podemos determinar que la calidad de la clusterización es relativamente buena ya que la distancia entre clusters es grande (los clusters están diferenciados entre si), pero las distancias dentro de los clusters son pequeñas (los clusters están formados por elementos muy parecidos).

A continuación, vamos a mostrar un plot donde se muestran claramente los clusters de diferentes colores.

```
fviz_cluster(kc, data=ppcc)
```

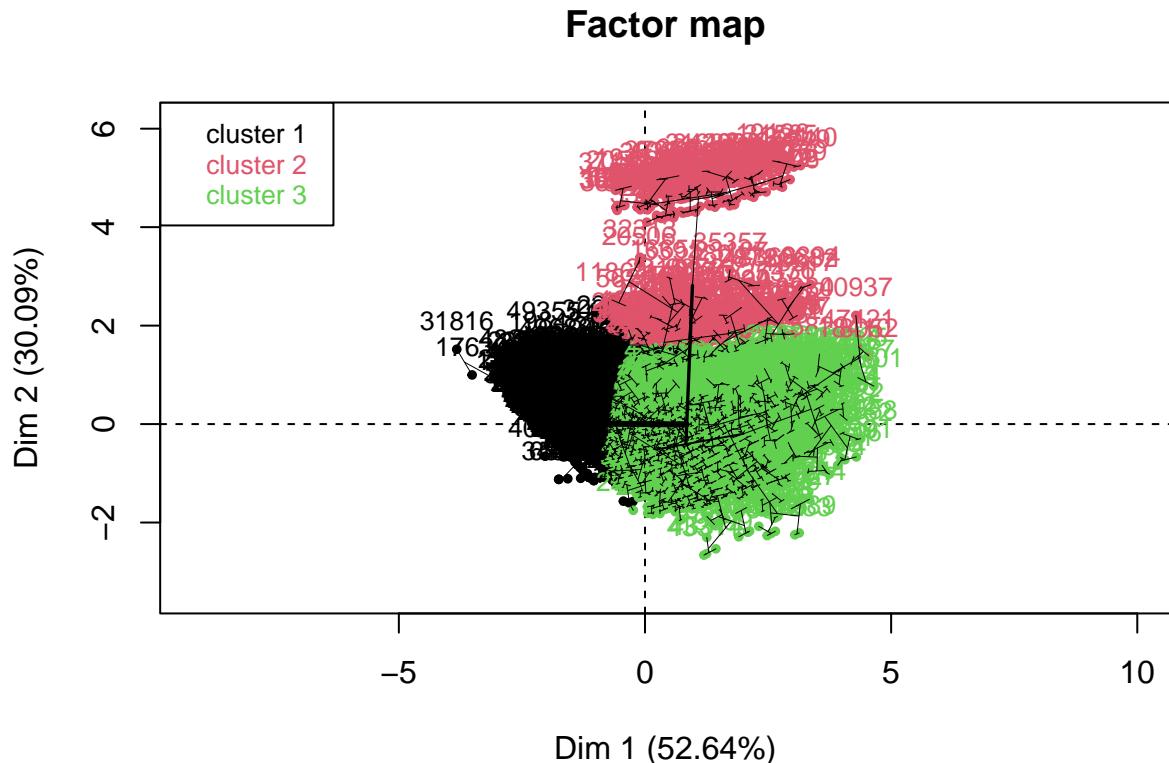


En

este gráfico podemos ver como no existe una definición tan clara como esperábamos en la clusterización.

Por último, vamos a volver a mostrar el gráfico resultante del clustering jerárquico, para poder comparar mejor los resultados:

```
plot.HCPC(res.hcpc, choice="map")
```



Podemos ver como los dos procesos de clusterización han llevado a resultados realmente diferentes. A primera vista, se puede ver como la clusterización jerárquica da un resultado mucho más comprensible en el plano de las dos componentes que se han usado para realizar la clusterización.

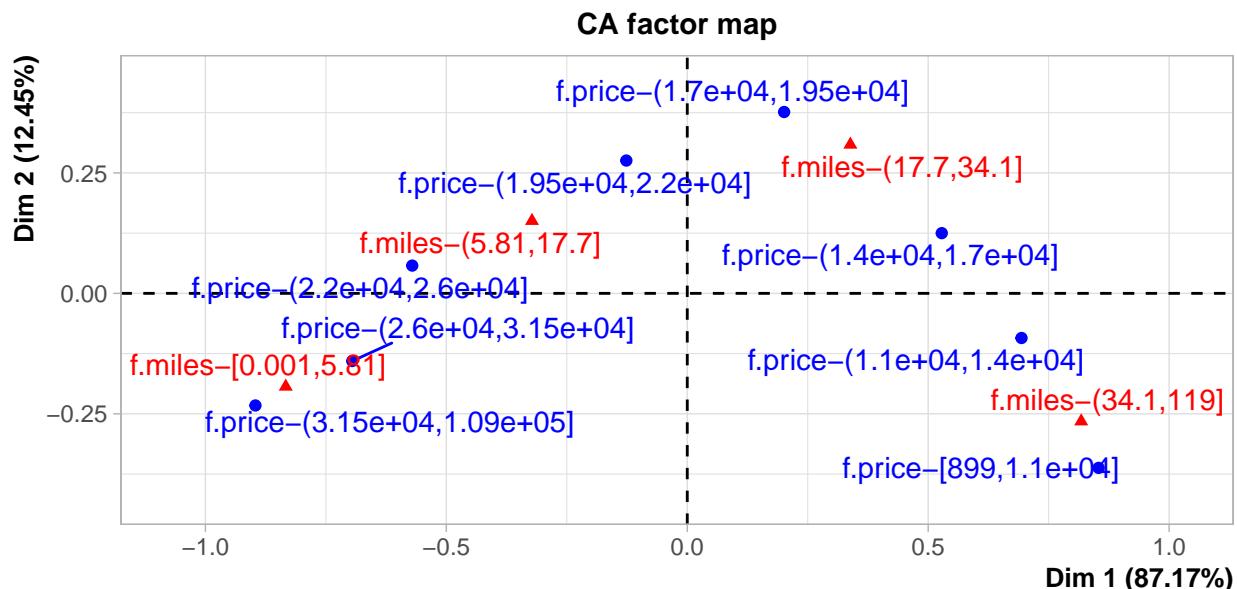
Por otro lado, en los dos procesos se ha definido el primer cluster de manera muy similar.

### 3 Correspondence Analysis

Vamos a proceder a realizar el análisis de correspondencias entre variables.

En primer lugar, vamos a analizar la correspondencia entre f.price y f.miles.

```
tt<-table(df[,c("f.price","f.miles")])  
res.ca<-CA(tt)
```



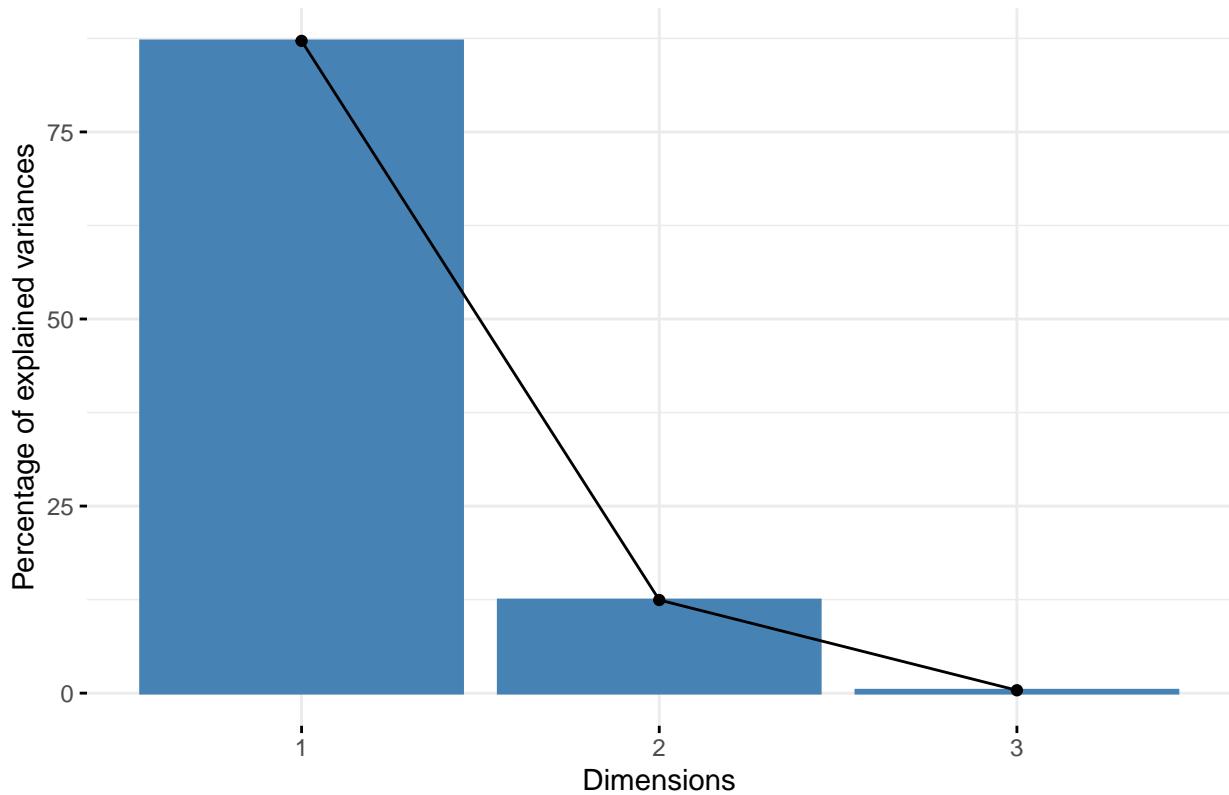
```
chisq.test(tt)
```

```
Pearson's Chi-squared test  
  
data: tt  
X-squared = 2267.6, df = 21, p-value < 2.2e-16
```

Podemos ver que la mayoría de la variabilidad se acumula en la primera componente (87%).

```
fviz_eig(res.ca)
```

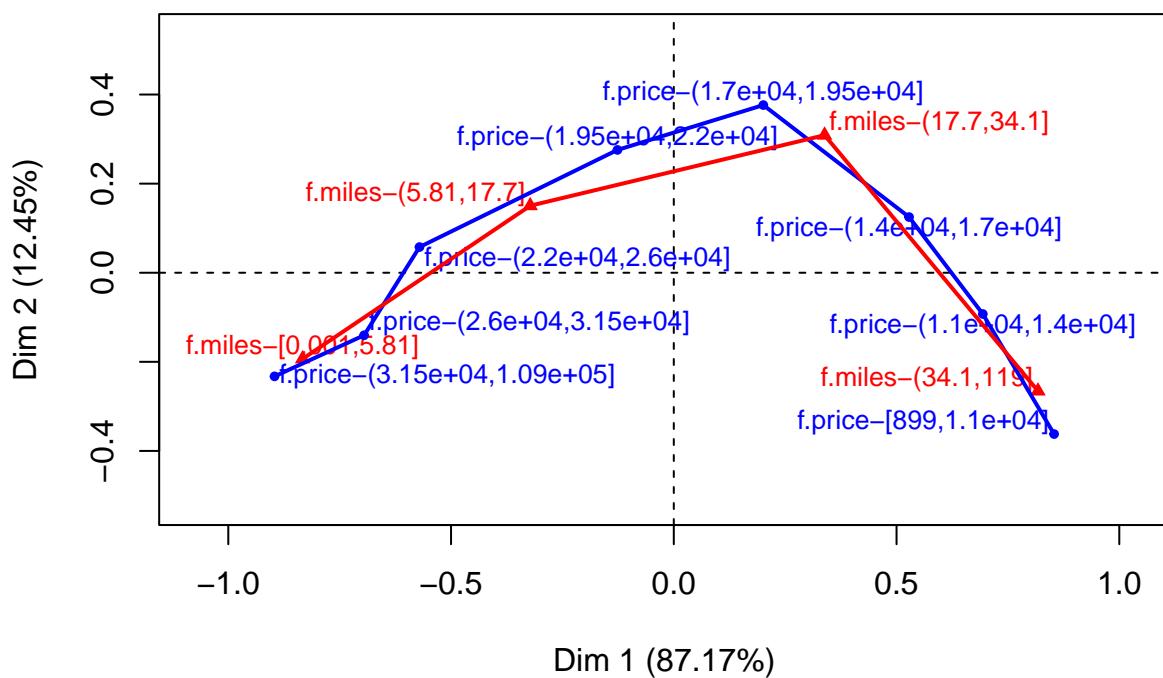
## Scree plot



Si realizamos los plots, podemos ver la presencia del efecto Guttman que nos indica que las variables estan fuertemente relacionadas.

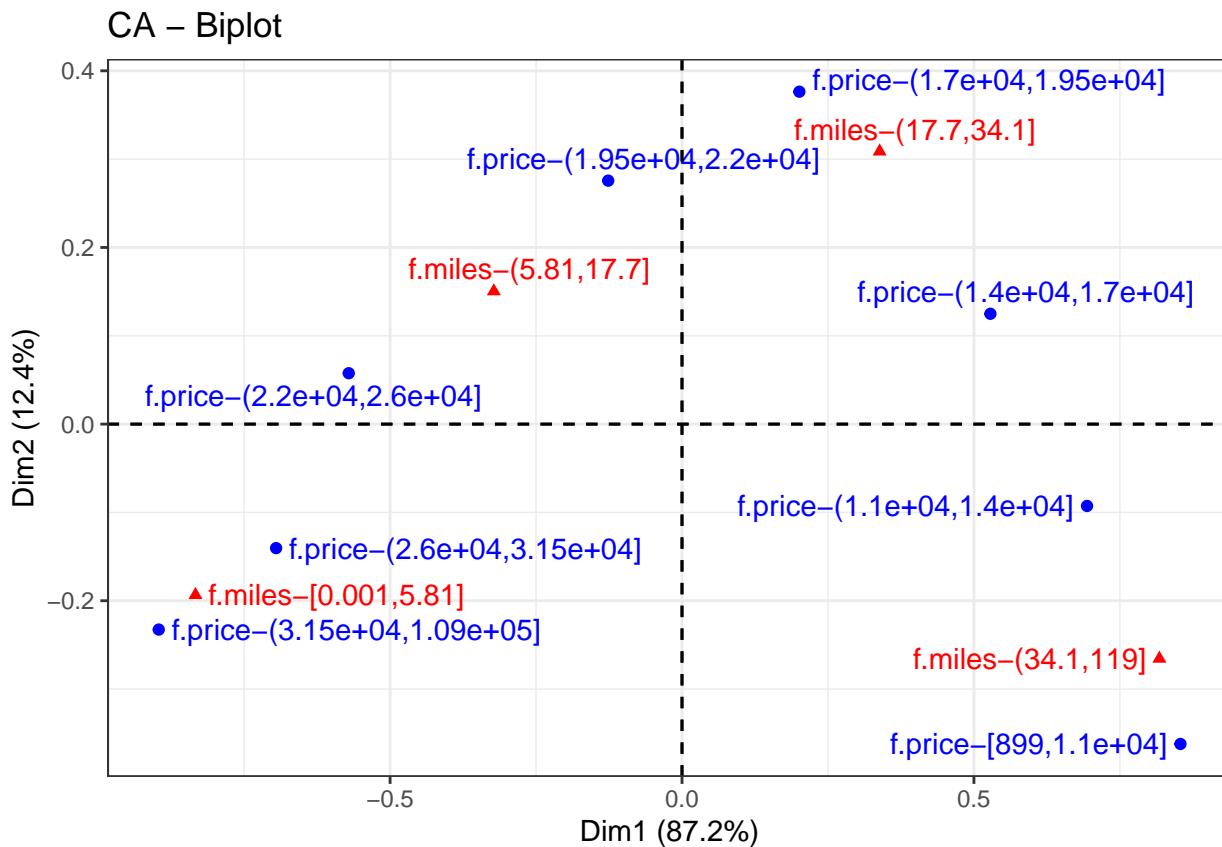
```
plot( res.ca, cex=0.8, graph.type = "classic" )
lines( res.ca$row$coord[,1], res.ca$row$coord[,2], col="blue", lwd = 2 )
lines( res.ca$col$coord[,1], res.ca$col$coord[,2], col="red", lwd = 2 )
```

## CA factor map



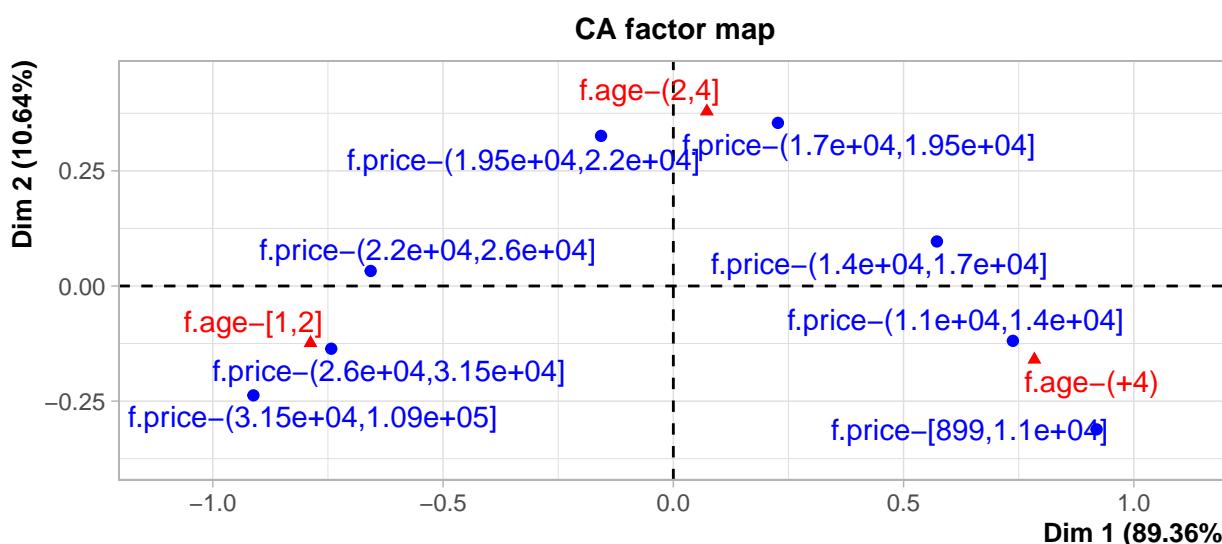
En el siguiente gráfico se puede ver como los coches con menor kilometraje tienen precios más altos y viceversa.

```
fviz_ca_biplot(res.ca, repel=TRUE)+theme_bw()
```



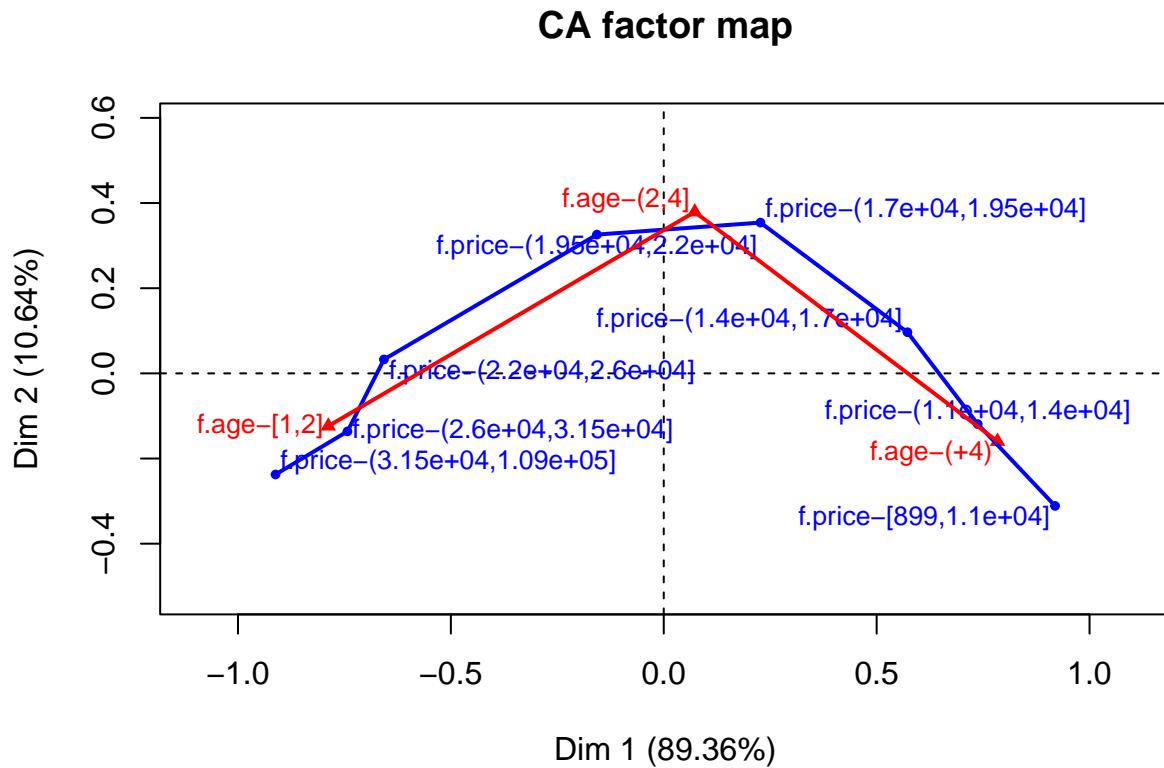
A continuación, vamos a realizar el mismo proceso para las variables f.price y f.age.

```
tt<-table(df[,c("f.price","f.age")])
res.ca<-CA(tt)
```



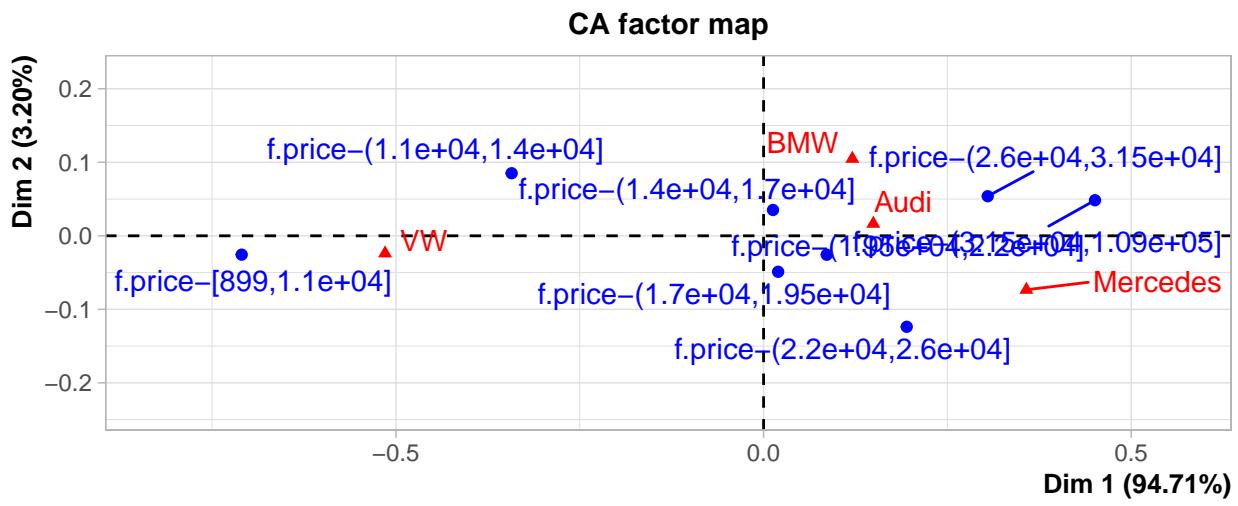
En este caso, también se puede ver la clara relación que hay entre las variables, donde los coches más nuevos son más caros y los viejos más baratos.

```
plot( res.ca, cex=0.8, graph.type = "classic" )
lines( res.ca$row$coord[,1], res.ca$row$coord[,2], col="blue", lwd = 2 )
lines( res.ca$col$coord[,1], res.ca$col$coord[,2], col="red", lwd = 2 )
```



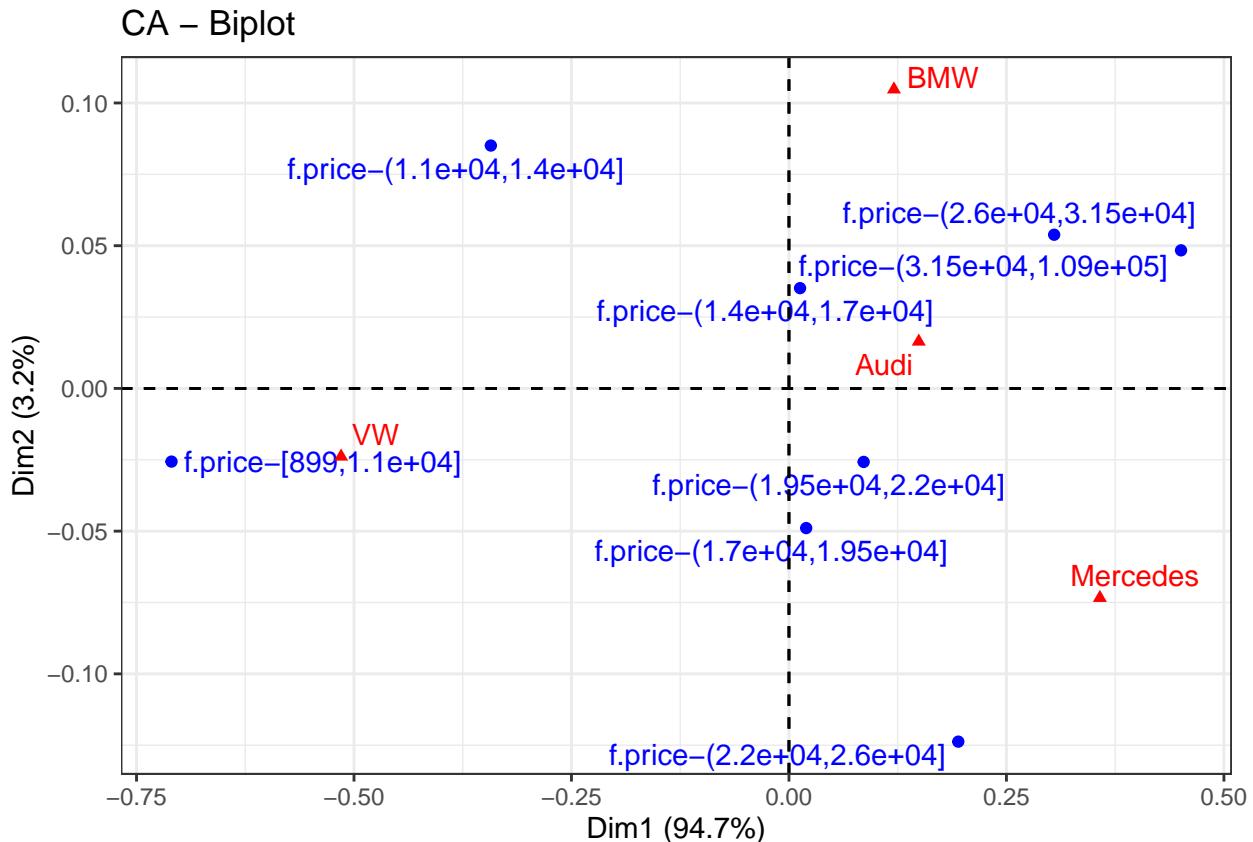
Por último, vamos a analizar la relación entre f.price y manufacturer. En este caso, se puede ver claramente los coches VW tienen precios más baratos que los BMW, Audi o Mercedes.

```
tt<-table(df[,c("f.price","manufacturer")])
res.ca<-CA(tt)
```



En este caso, en el siguiente gráfico se puede ver claramente los coches VW tienen precios más baratos que los BMW, Audi o Mercedes.

```
fviz_ca_biplot(res.ca,repel=TRUE)+theme_bw()
```



Es por este motivo, que factores como pueden ser manufacturer o engineSize pueden no resultar determinantes a la hora de explicar nuestro target numérico price.

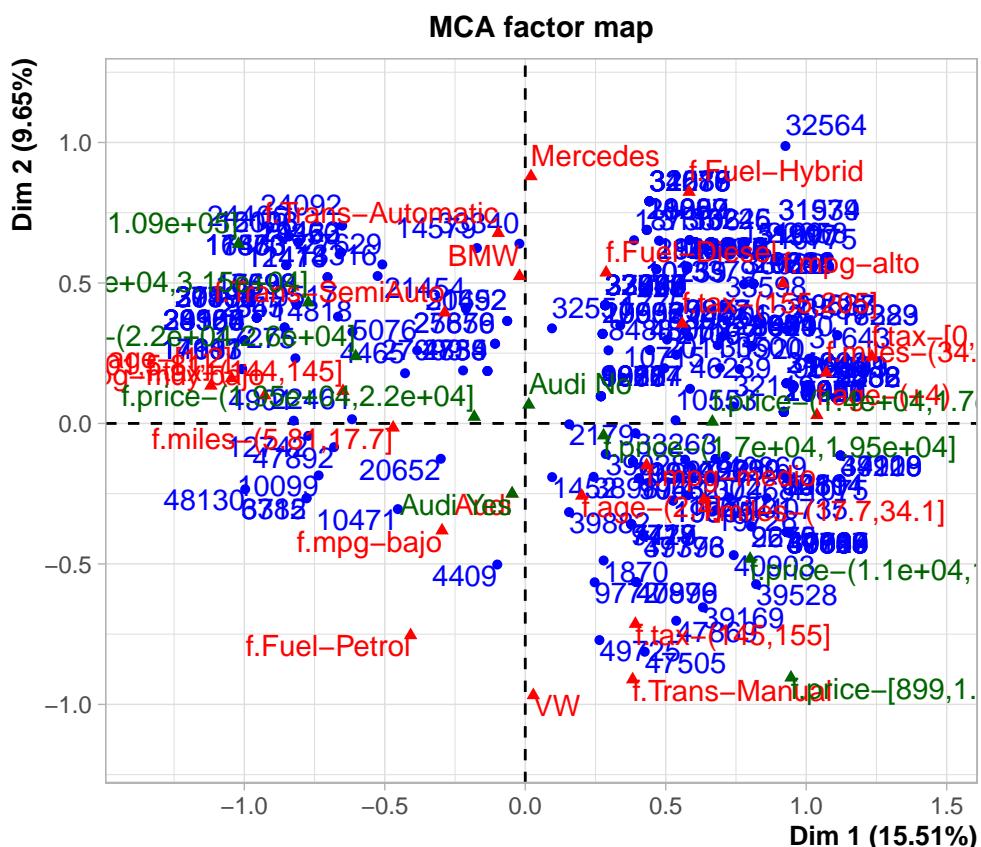
## 4 Multiple Correspondence Analysis

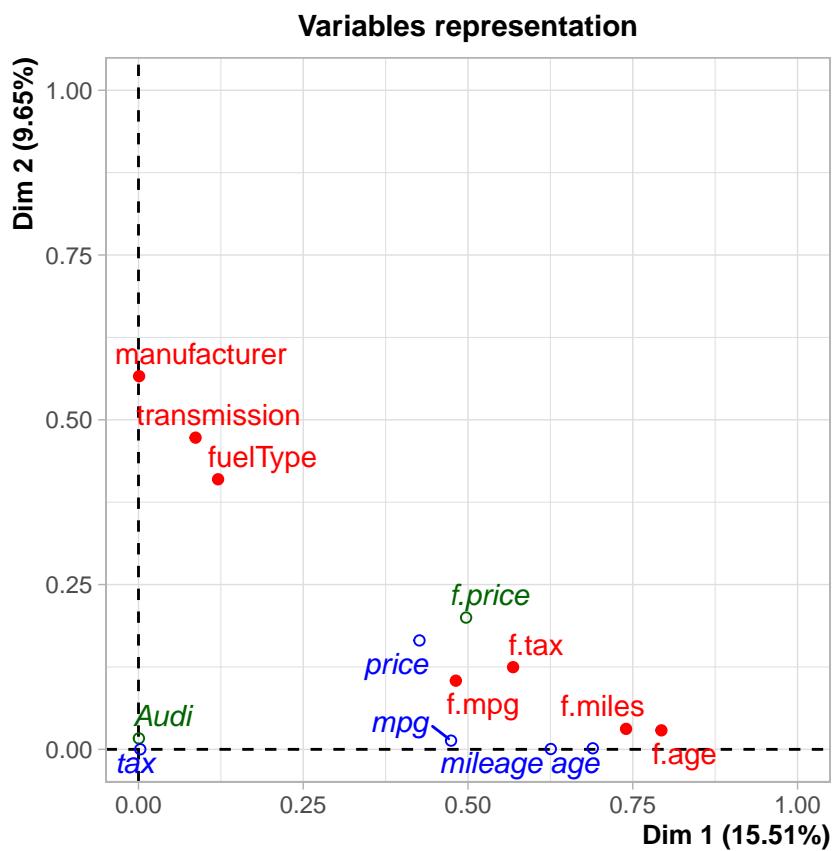
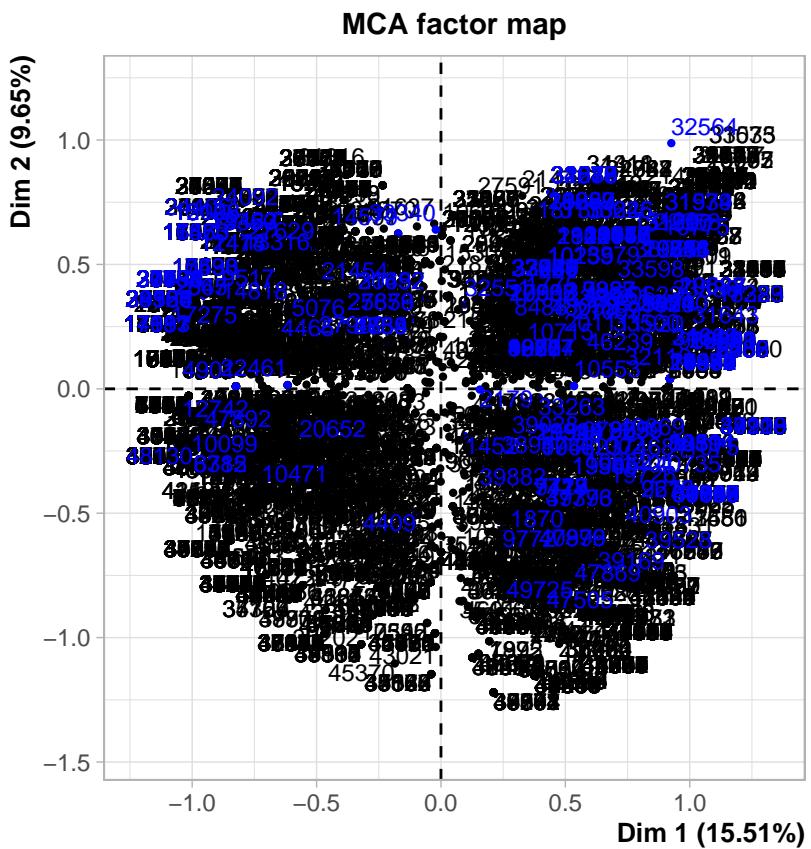
A continuación vamos a proceder a realizar el MCA. Para ello, en primer lugar vamos a seleccionar los individuos que hemos considerado como multivariant outliers para añadirlos como suplementarios. También se ha considerado la eliminación de la variable engineSize ya que si se incluye genera errores. También se han suprimido los factores model y year. Model se ha quitado ya que tiene demasiados valores y acaba no siendo explicativo. Year se ha eliminado porque guarda demasiada relación con f.age. Se añaden como variables suplementarias los factores f.price y Audi, además de la variable price, ya que son nuestros targets.

```
llvout<-which(df$mout=="YesMOut");length(llvout)
```

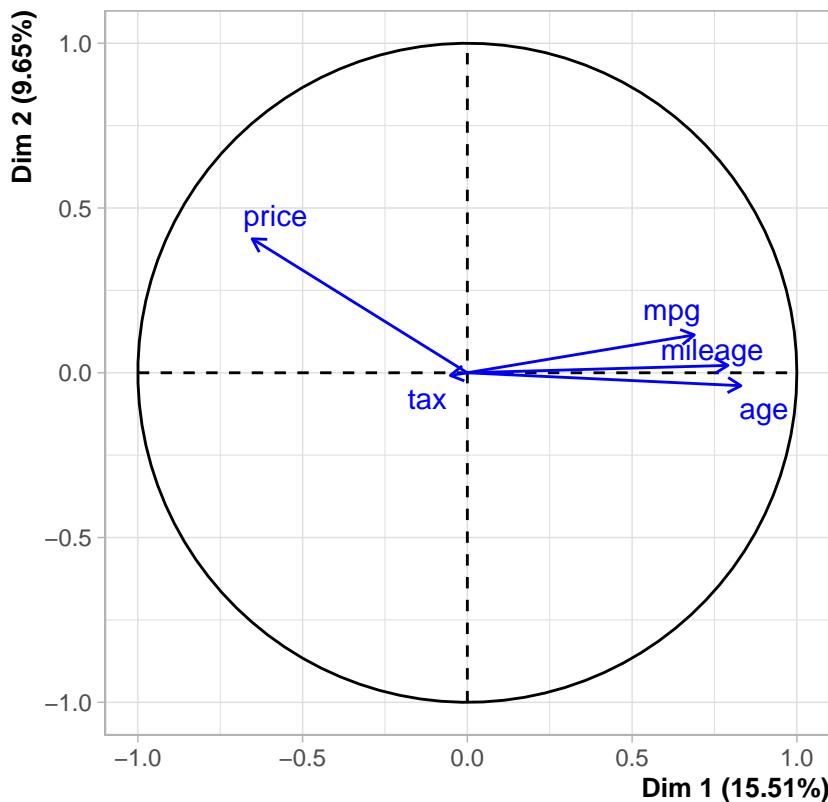
[1] 241

```
res.mca<-MCA(df[,c("f.price","Audi",vars_cat[c(3:4,6,8:11)],"price", vars_num) ], quali.sup=c(1,2),quant=1)
```





## Supplementary quantitative variables



Vamos a aplicar el criterio de Kaiser para determinar el número de dimensiones relevantes para continuar el análisis MCA. En este caso, como los eigenvalues no están normalizados, nos quedaremos con todas las dimensiones que tengan un eigenvalue mayor que la media de todos los eigenvalues.

```
length(which(res.mca$eig[,1] > mean(res.mca$eig[,1])))
```

```
[1] 7
```

A continuación podemos ver los eigenvalues de las dimensiones seleccionadas, los porcentajes de varianza que acumulan y las varianza que hay acumulada hasta esa dimensión. Podemos ver que en el caso de la dimensión 7, que es la que nos ha indicado el criterio de Kaiser, se ha acumulado cerca de un 60% de la varianza.

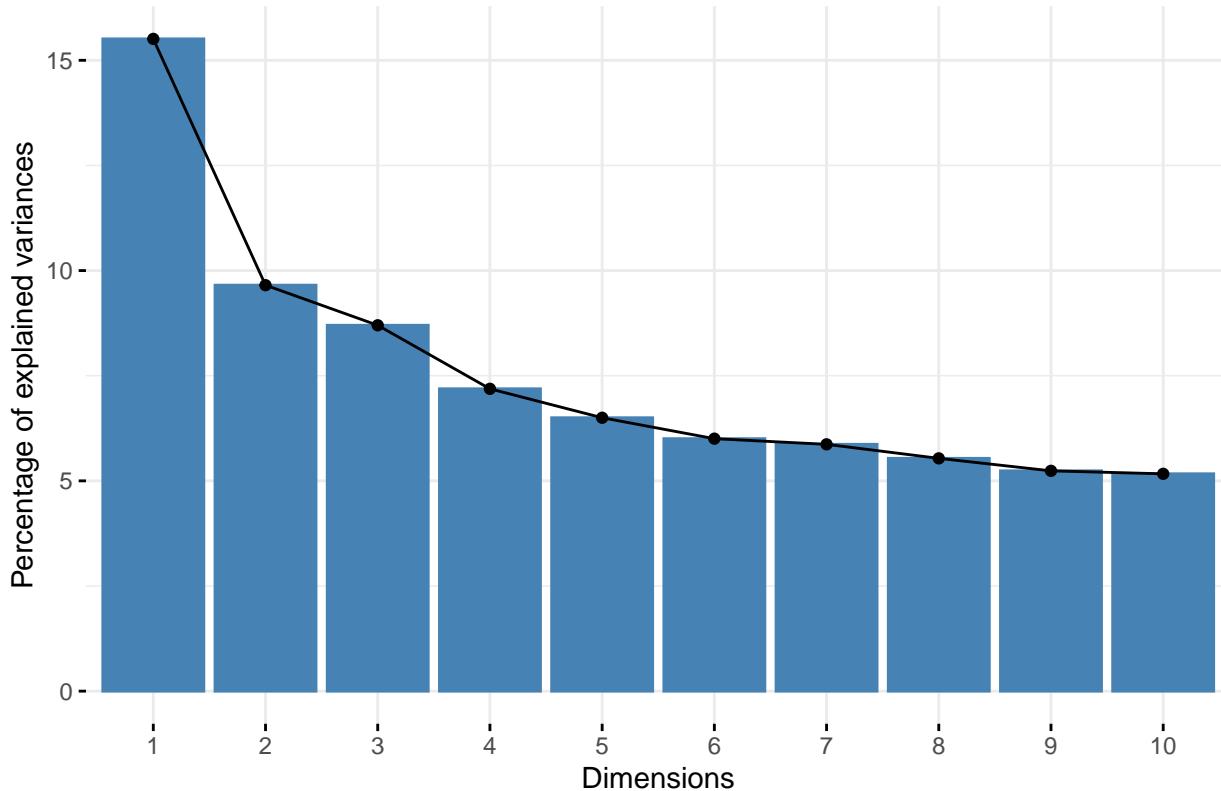
```
res.mca$eig[1:7,]
```

	eigenvalue	percentage of variance	cumulative percentage of variance
dim 1	0.3987936	15.508640	15.50864
dim 2	0.2482132	9.652737	25.16138
dim 3	0.2236947	8.699237	33.86061
dim 4	0.1848499	7.188607	41.04922
dim 5	0.1671620	6.500745	47.54997
dim 6	0.1543653	6.003097	53.55306
dim 7	0.1509051	5.868533	59.42160

En el siguiente gráfico podemos ver de una manera más visual la varianza acumulada para cada dimensión:

```
fviz_eig(res.mca)
```

Scree plot



## 4.1 Análisis segun las variables

### 4.1.1 Factores

A continuación podemos ver el peso que adquieren los distintos factores para cada dimensión. Podemos ver que para la dimensión 1 destacan los vehículos más nuevos y con menos kilometraje, mientras que para la dimensión 2 se acumulan muchos vehículos de la marca VM y de transmisión Manual.

```
head(res.mca$var$contrib)
```

	Dim 1	Dim 2	Dim 3	Dim 4	Dim 5
f.Trans-Manual	1.86642658	17.0519591	0.07469764	4.9175161	4.4279298
f.Trans-SemiAuto	1.15169203	3.4996681	0.33077117	2.2507154	1.9816458
f.Trans-Automatic	0.08346797	6.6664716	1.08306670	0.5925208	0.5628084
f.Fuel-Diesel	1.68958227	9.4465219	0.52101597	1.3584794	2.0289889
f.Fuel-Petrol	2.49542678	13.6859587	0.76172684	2.3491720	0.4722622
f.Fuel-Hybrid	0.14079989	0.4514899	0.03297777	1.0727704	34.4510988

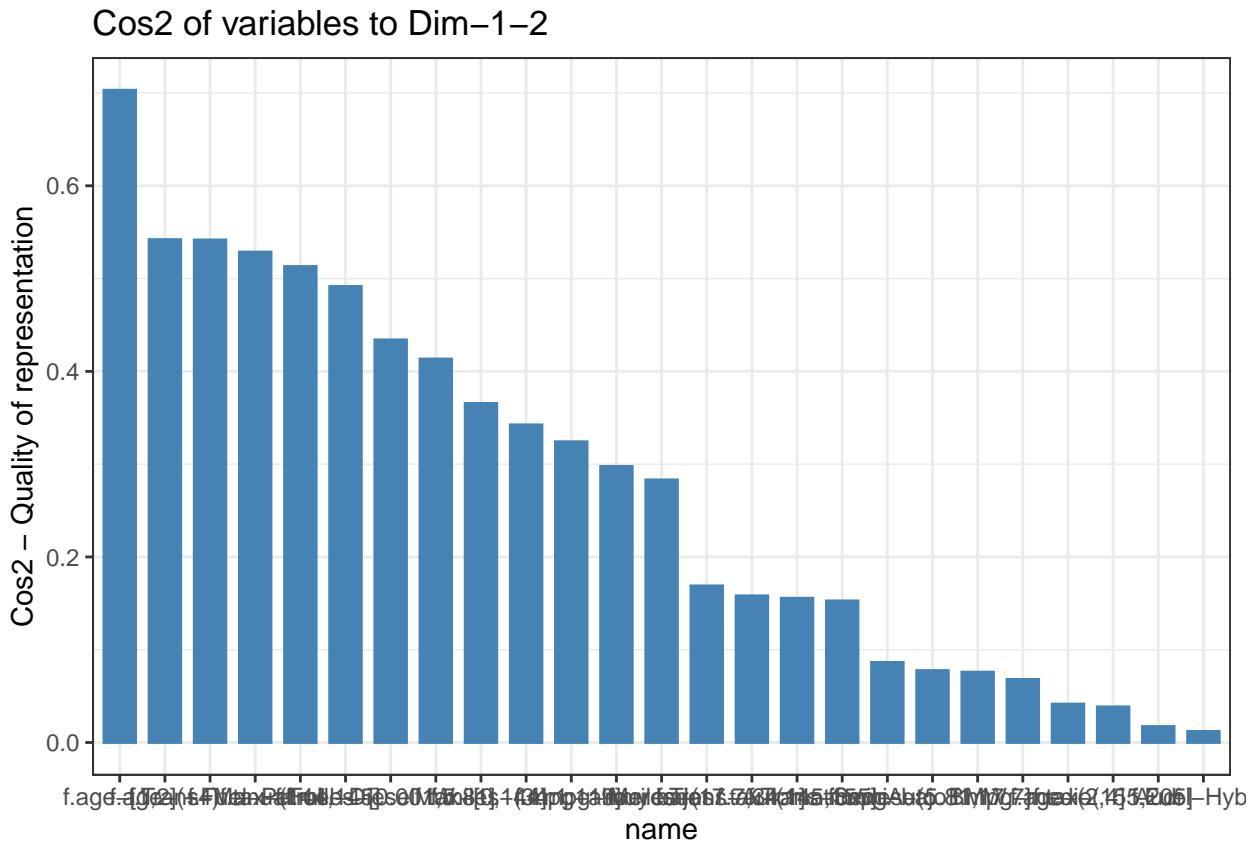
Si analizamos el estadístico cos2, podemos llegar a una conclusión similar que con la contribución. Podemos ver que para la Dim 1, el valor más relevante es f.age-[1,2] y para la segunda dimensión estaría f.Trans-Manual.

```
head(res.mca$var$cos2)
```

	Dim 1	Dim 2	Dim 3	Dim 4	Dim 5
f.Trans-Manual	0.081004566	0.460627238	0.0018184982	0.09892713	0.080554306
f.Trans-SemiAuto	0.052777652	0.099819940	0.0085025421	0.04780852	0.038065294
f.Trans-Automatic	0.003116562	0.154927586	0.0226839466	0.01025488	0.008808582
f.Fuel-Diesel	0.109707327	0.381772789	0.0189764408	0.04088654	0.055223688
f.Fuel-Petrol	0.119767963	0.408834751	0.0205070187	0.05226148	0.009500981
f.Fuel-Hybrid	0.003976463	0.007936325	0.0005224243	0.01404341	0.407837474

En el siguiente gráfico se pueden apreciar los valores de cos2 de un modo más visual:

```
fviz_cos2(res.mca, choice = "var", axes = 1:2)+theme_bw()
```



Por último, si echamos un vistazo a eta2, podemos ver el peso que tienen los factores, no solo los valores que estos toman.

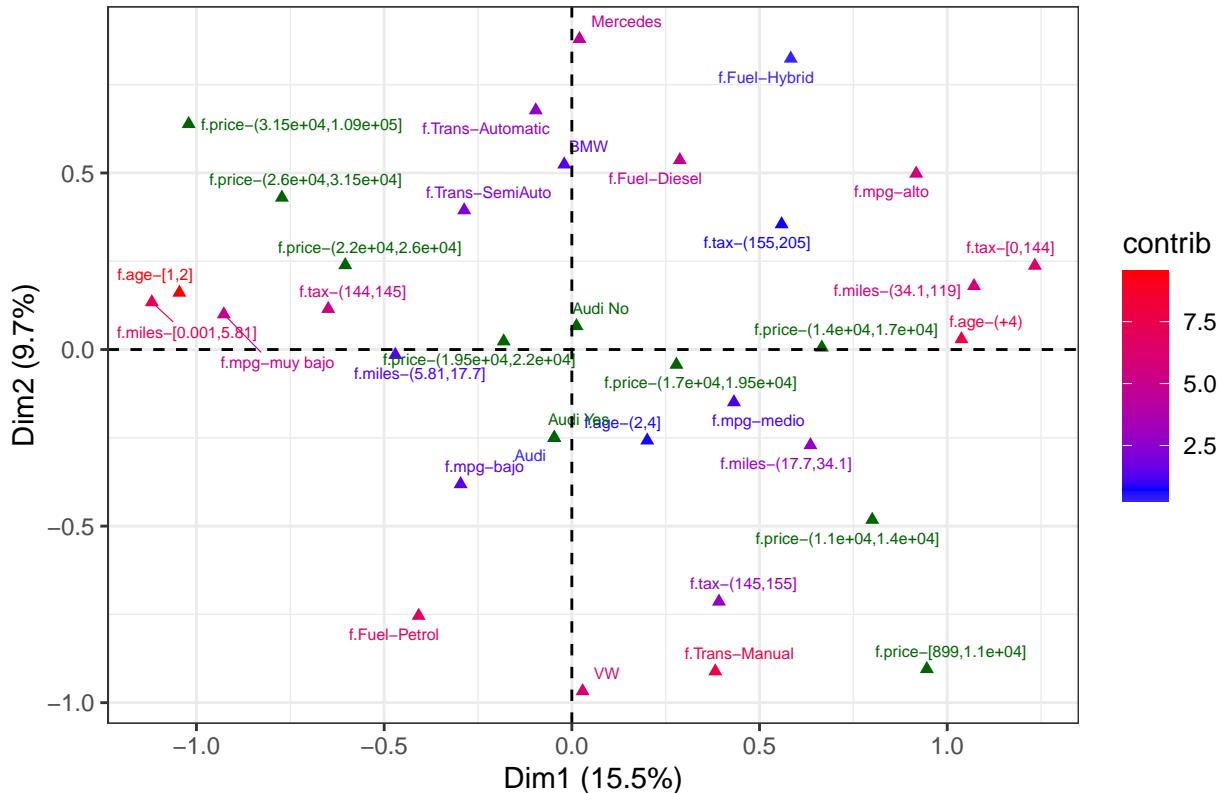
res.mca\$var\$eta2

	Dim 1	Dim 2	Dim 3	Dim 4	Dim 5
transmission	0.0865825002	0.47291245	0.02330842	0.10042020	0.081586248
fuelType	0.1207573427	0.40976974	0.02060238	0.06185623	0.432392070
manufacturer	0.0009114589	0.56611264	0.14378407	0.10776968	0.028222856
f.miles	0.7398617465	0.03104883	0.32222216	0.35701789	0.003286308
f.mpg	0.4815674962	0.10414582	0.21354811	0.10311629	0.456666486
f.tax	0.5684324235	0.12467461	0.45581714	0.29060566	0.166884121
f.age	0.7934421913	0.02882849	0.38658032	0.27316328	0.001096075

En el siguiente gráfico, podemos ver la contribución que tienen las variables para las dos primeras dimensiones generadas por el MCA, y que recordemos, acumulan cerca del 25% de la variabilidad.

```
fviz_mca_var(res.mca, col.var="contrib", repel=TRUE, labelsize = 2)+  
  scale_color_gradient2(low="green", mid="blue",  
  high="red", midpoint=0.75)+theme_bw()
```

## Variable categories – MCA



### 4.1.2 Variables numéricas

Vamos a echar un vistazo a las variables numéricas. En la siguiente salida podemos observar la correlación que existe entre las variables numéricas originales y las componentes que se han obtenido con el MCA.

```
res.mca$quanti
```

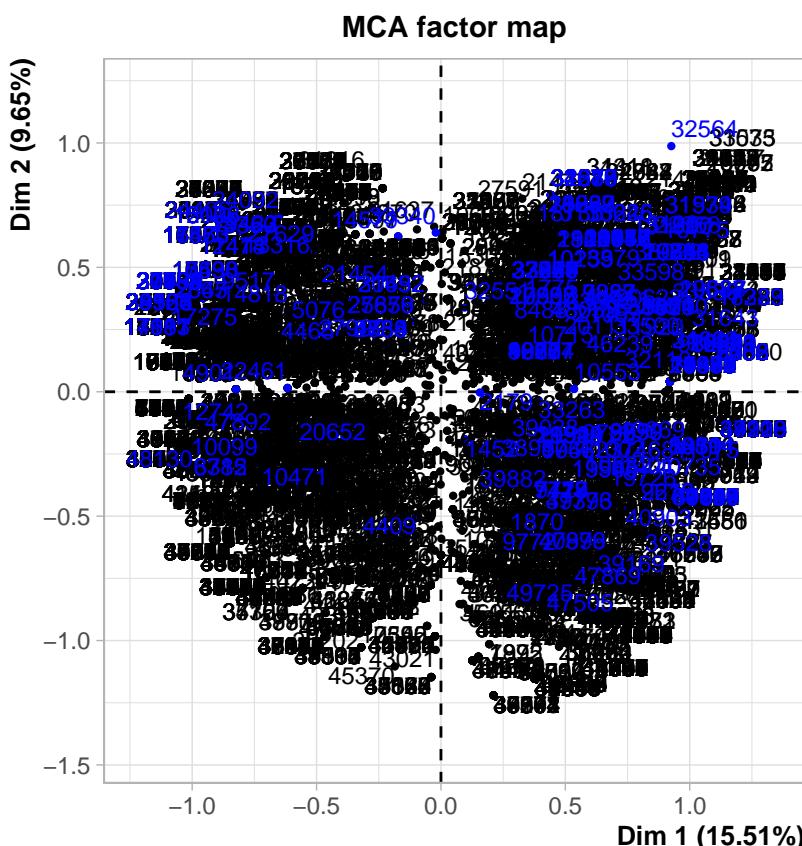
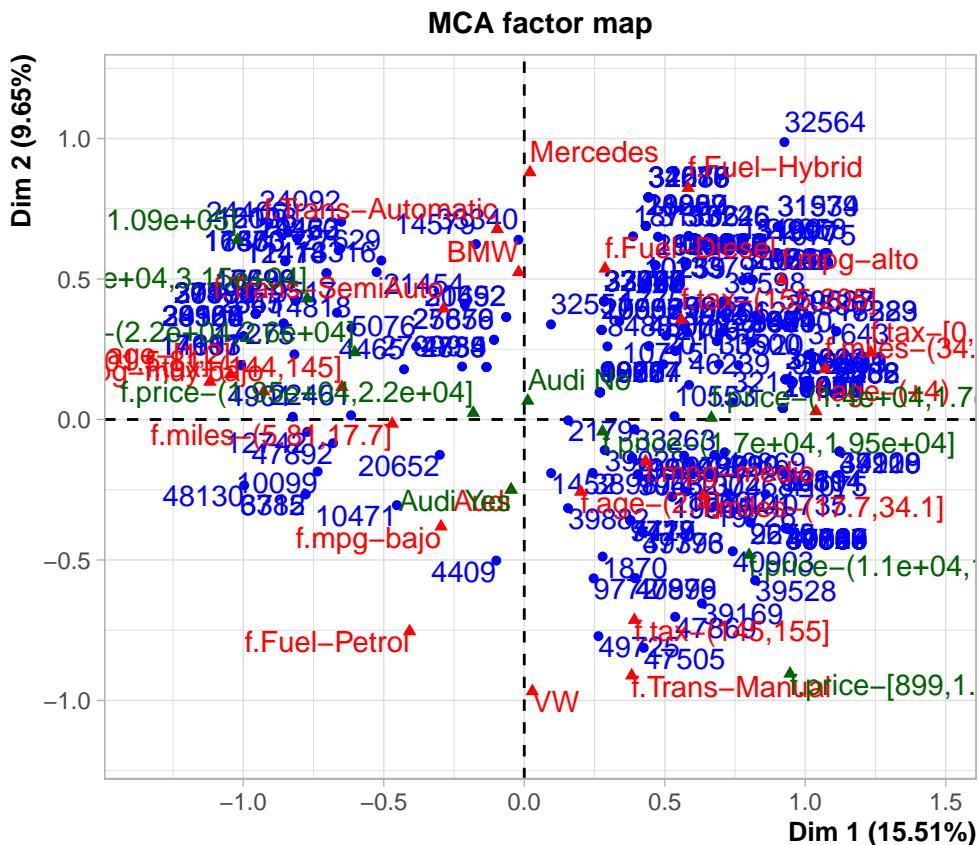
```
$coord
      Dim 1       Dim 2       Dim 3       Dim 4       Dim 5
price -0.65294608  0.40647213  0.01852461  0.10983643  0.10298148
mileage  0.79095768  0.02241196  0.28993254  0.05398482  0.02082680
tax     -0.05171566 -0.00805522  0.51151048  0.40339669 -0.02175406
mpg      0.68868164  0.11496164 -0.40072956 -0.31001209 -0.15611429
age      0.83025290 -0.03928698  0.27147016  0.13365610  0.01868976
```

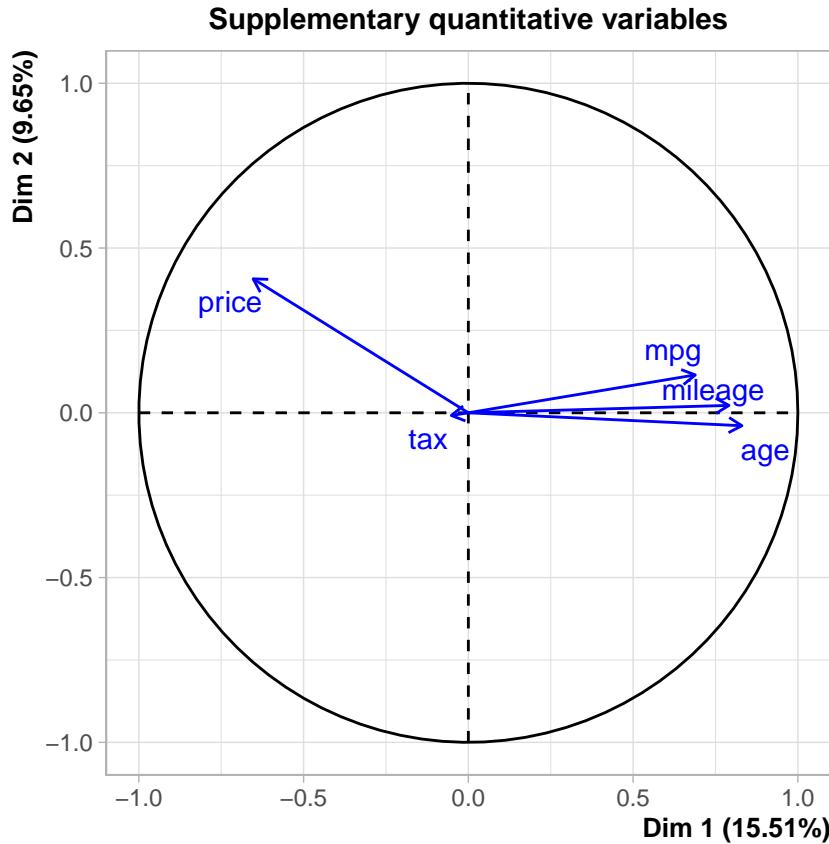
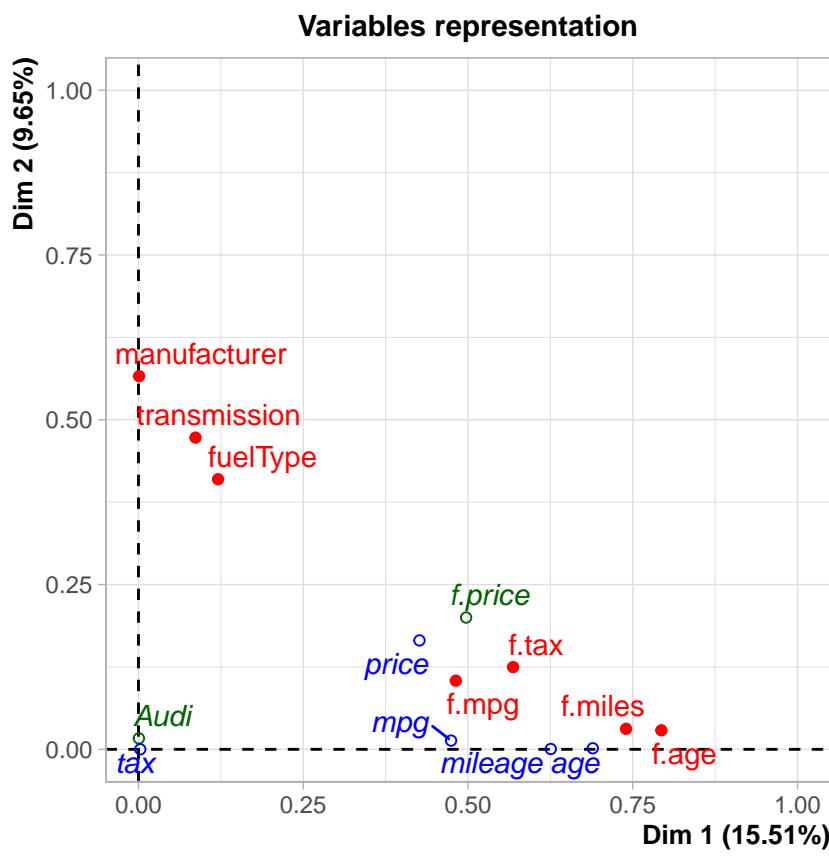
Podemos destacar la fuerte correlación de las variables age y mileage con la primera componente, y de price con la segunda.

## 5 Clustering Jerárquico desde MCA

Vamos a proceder a volver a realizar el clustering jerárquico pero esta vez lo vamos a lanzar desde el MCA en lugar del PCA. Vamos a añadir el número de componentes que hemos determinado durante el análisis MCA a partir del criterio de Kaiser.

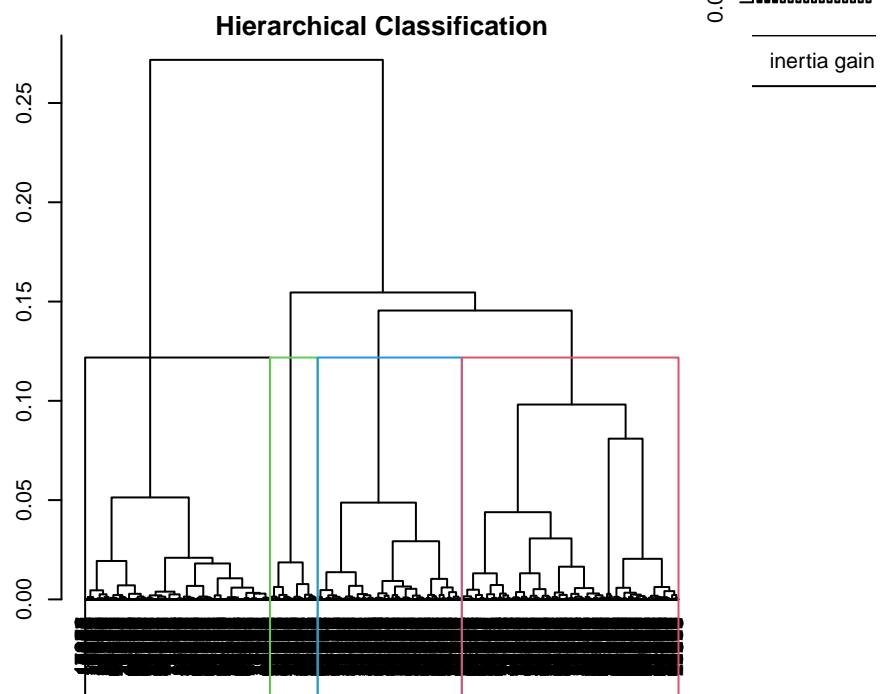
```
res.mca<-MCA(df[,c("f.price","Audi",vars_cat[c(3:4,6,8:11)],"price",vars_num) ],quali.sup=c(1,2),quanti
```



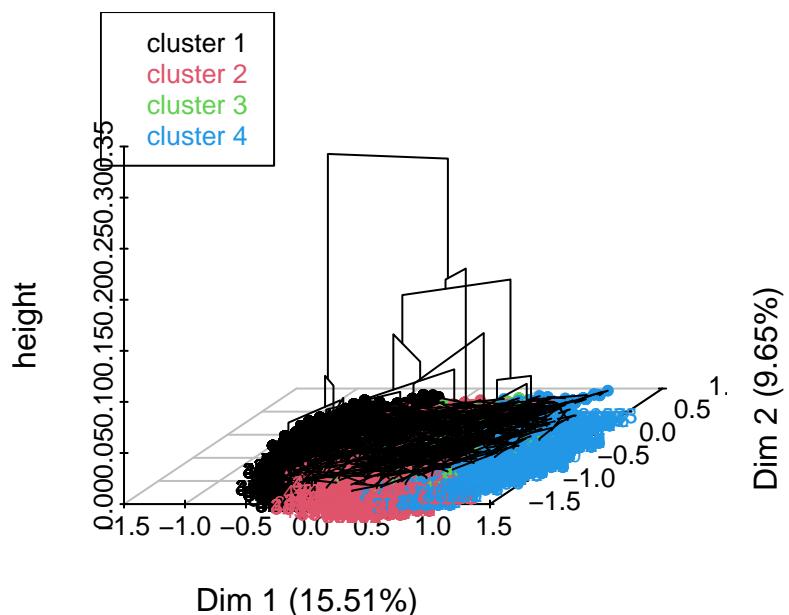


```
res.hcmc<-HCPC(res.mca, nb.clust=-1, order=TRUE)
```

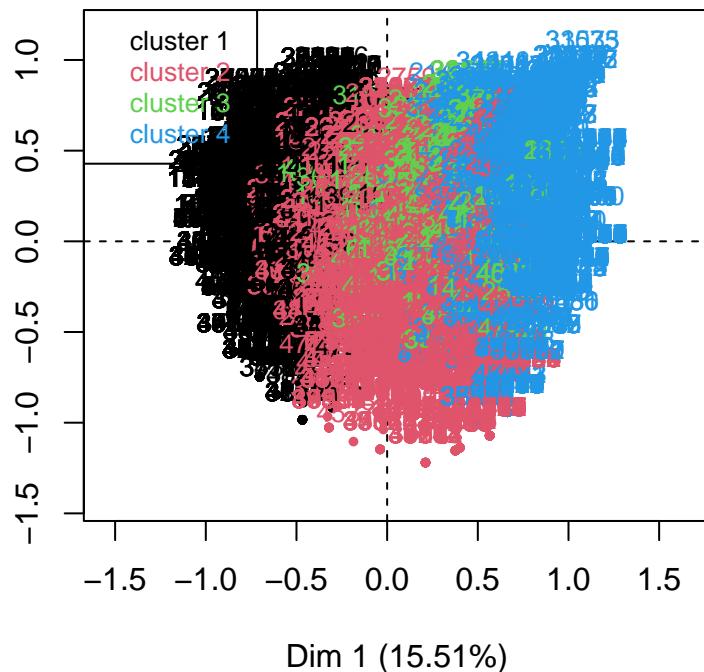
## Hierarchical Clustering



## Hierarchical clustering on the factor map



## Factor map



Como podemos ver en la salida, en este caso, a partir del parámetro `nb.clust=-1` que como hemos mencionado anteriormente, automatizaba la selección del número óptimo de clusters, se han generado 4 clusters.

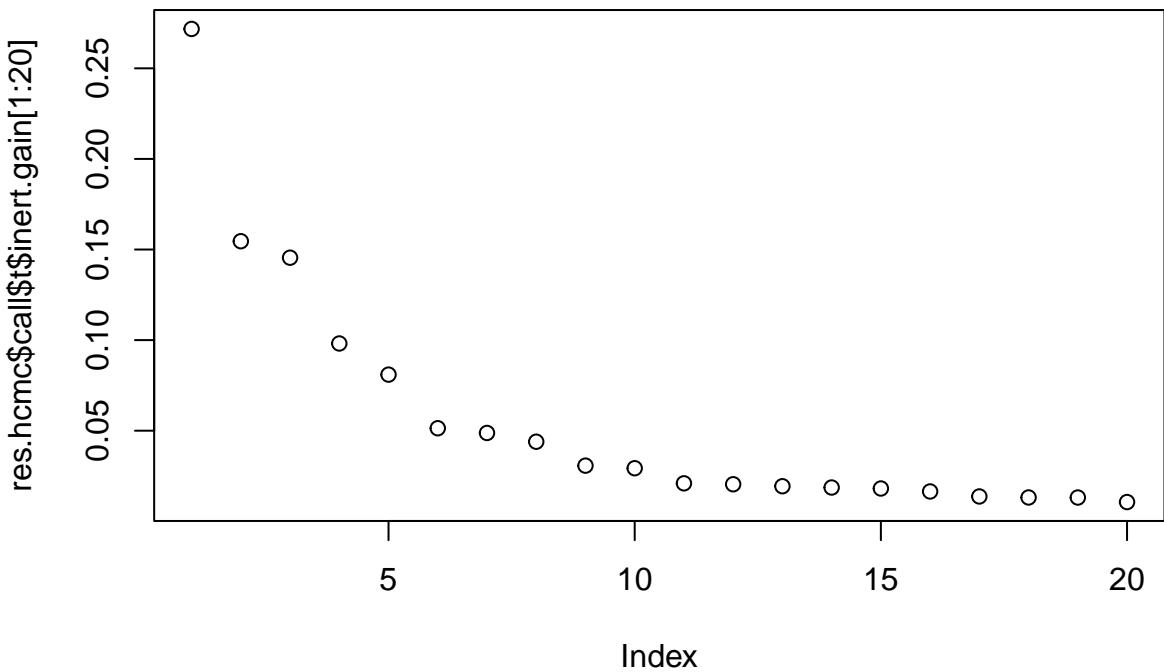
Sin embargo, si tenemos en cuenta el criterio de Kaiser, podemos ver que, en este caso, este número de componentes no son suficientes. Segundo Kaiser, deberíamos tomar 7 componentes.

```
length(which(res.hcmc$call$t$res$eig[,1] > mean(res.hcmc$call$t$res$eig[,1])))
```

```
[1] 7
```

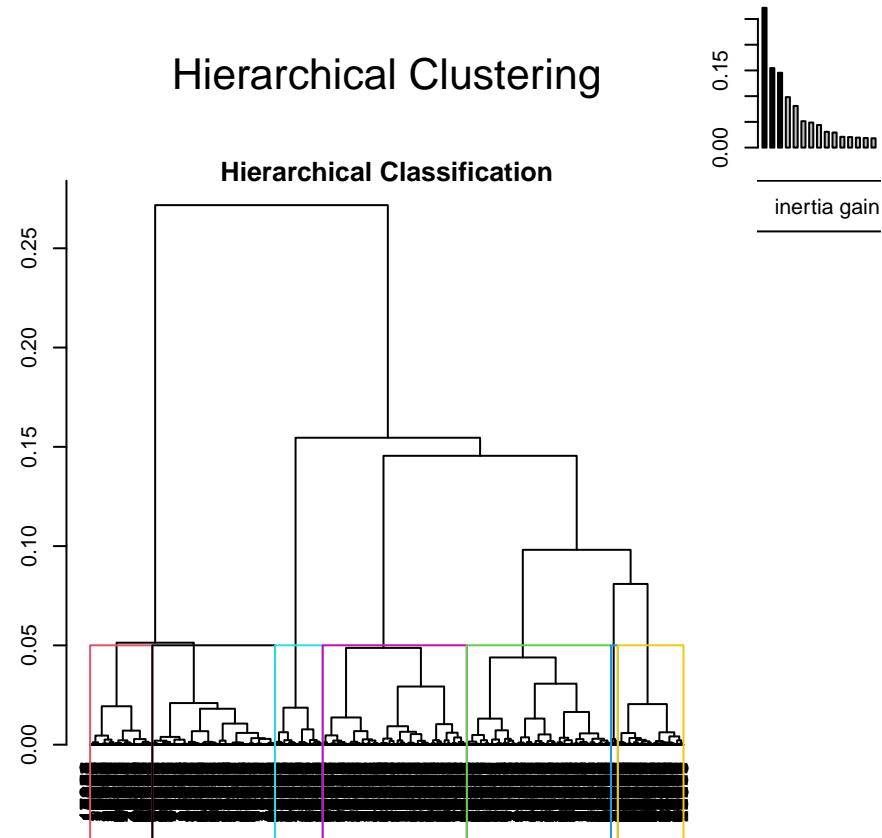
En el siguiente plot podemos ver la ganancia de inercia que se produce en nuestro modelo de clusterización jerárquica. Si aplicamos la regla de elbow, podemos determinar que el número óptimo de clusters sería alrededor de 6.

```
plot(res.hcmc$call$t$inert.gain[1:20])
```

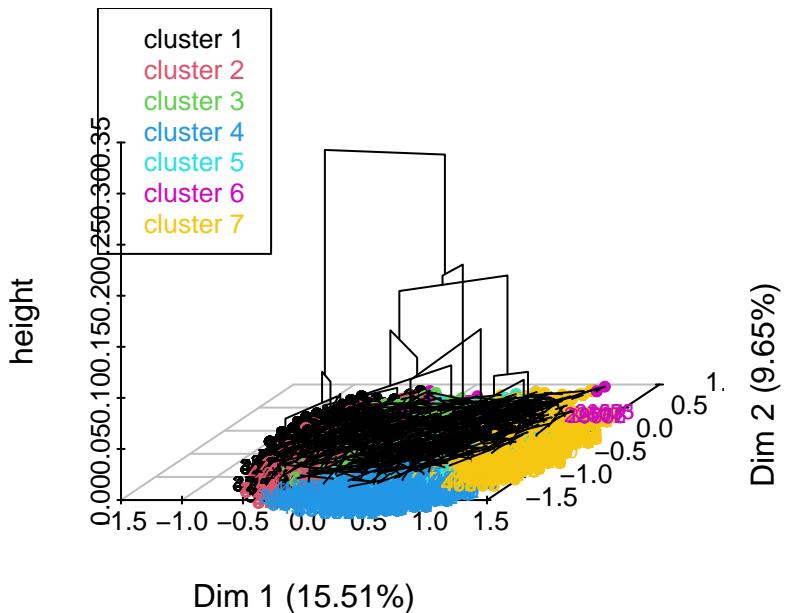


Por lo tanto, y resumiento, nb.clust nos da 4 componentes, elbow nos da 6 y Kaiser nos da 7. Nos quedaremos con el de Kaiser porque es el más específico.

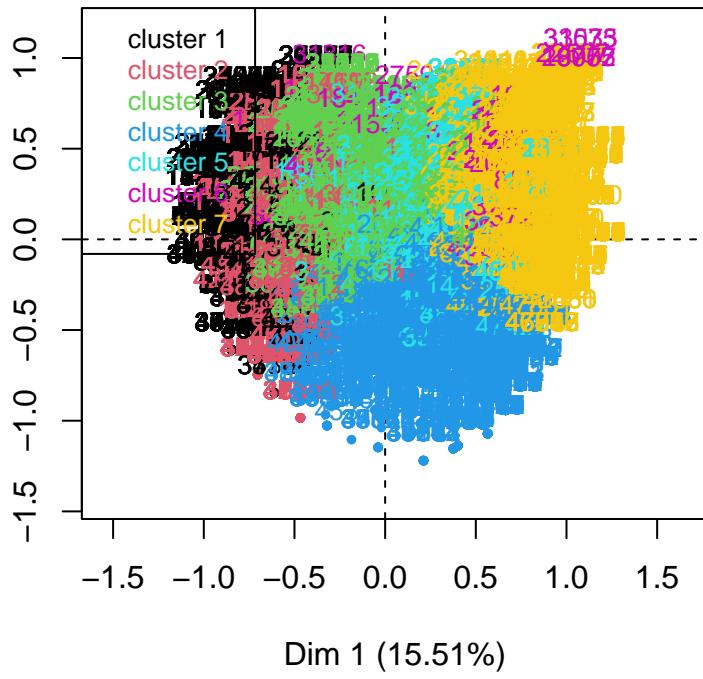
```
res.hcmc<-HCPC(res.mca, nb.clust=7, order=TRUE)
```



## Hierarchical clustering on the factor map



## Factor map



A continuación podemos ver inercia acumulada en las 7 primeras componentes.

```
(res.hcmc$call$t$within[1]-res.hcmc$call$t$within[1:7])/res.hcmc$call$t$within[1]
```

```
[1] 0.0000000 0.1778367 0.2790013 0.3742149 0.4384370 0.4914121 0.5250179
```

Crearemos una nueva variable en nuestro df para guardar en que cluster se han asignado los individuos.

```

df$claHCMC<-7
df[row.names(res.hcmc$data.clust),"claHCMC"]<-res.hcmc$data.clust$clust
df$claHCMC<-factor(df$claHCMC)
levels( df$claHCMC ) <- paste0( "f.claHCMC-",levels( df$claHCMC ) )
table(df$claHCMC)

```

```

f.claHCMC-1 f.claHCMC-2 f.claHCMC-3 f.claHCMC-4 f.claHCMC-5 f.claHCMC-6
 959        712       873       825       387       55
f.claHCMC-7
 1189

```

## 5.1 Análisis según las variables

### 5.1.1 Factores

En primer lugar, vamos a analizar el estadístico chi-squared para determinar que factores son las que más determinan las diferencias entre clusters:

```
res.hcmc$desc.var$test.chi2
```

	p.value	df
f.price	0.000000e+00	42
transmission	0.000000e+00	12
fuelType	0.000000e+00	12
manufacturer	0.000000e+00	18
f.miles	0.000000e+00	18
f.mpg	0.000000e+00	18
f.tax	0.000000e+00	18
f.age	0.000000e+00	12
Audi	4.185767e-22	6

Si profundizamos un poco más podemos ver caracterizar un poco los clusters según los valores de estos factores. <He omitido la salida porque ocupa mucho, pero podemos ver algunas de las conclusiones abajo>

```
#res.hcmc$desc.var$category
```

Podemos ver, por ejemplo, que el cluster 1 esta compuesto por coches muy nuevos (f.age=f.age-[1,2] -> Mod/Cla 96.6631908) y con poco kilometraje (f.miles=f.miles-[0.001,5.81] -> Mod/Cla 69.9687174). Por el contrario, prácticamente no hay coches viejos (f.age=f.age-(+4) -> Mod/Cla 0.1042753) o con precios bajos (f.price=f.price-[899,1.1e+04] -> Mod/Cla 0.0000000).

Contrariamente, en el cluster 7, se acumulan los vehículos con más de 4 años, Diesel y con mpg elevados.

### 5.1.2 Variables numéricas

En los que se refiere a las variables cuantitativas:

```
res.hcmc$desc.var$quanti.var
```

	Eta2	P-value
price	0.4897173	0
mileage	0.6023484	0
tax	0.5548954	0
mpg	0.5287686	0
age	0.6897952	0

Podemos ver que la que mejor explica la separación entre clusters es age, mientras que price parece ser la menos explicativa.

En este apartado, cabe destacar que todas las variables tienen factores derivados que se han usado para generar el clustering. Sin embargo, hay que tener en cuenta que en el caso de nuestro target price, su discretización solo

se ha añadido como variable suplementaria, de modo que no ha tenido influencia en la clusterización, hecho que explica que aparezca como la menos representativa.

Si analizamos más en profundidad, podemos ver las distribuciones que toman cada una de las variables cualitativas dentro de los distintos clusters.

```
res.hcmc$desc.var$quanti
```

	v.test	Mean in category	Overall mean	sd in category	Overall sd
price	38.190083	31064.974974	21027.36730	9144.2651126	9107.713091
tax	-2.006296	145.708878	146.39236	1.8304266	11.804945
mileage	-30.893568	4970.549531	21187.33402	4599.6345470	18189.736407
age	-33.974519	1.870699	3.68627	0.4810991	1.851781
mpg	-37.179059	41.742231	53.31998	8.1369598	10.790845
	p.value				
price	0.000000e+00				
tax	4.482465e-02				
mileage	1.457380e-209				
age	5.300403e-253				
mpg	1.487684e-302				
	\$'2'				
	v.test	Mean in category	Overall mean	sd in category	Overall sd
price	7.443810	23370.61938	21027.36730	6664.2265225	9107.713091
tax	-2.397051	145.41433	146.39236	1.8748685	11.804945
mpg	-5.093678	51.42021	53.31998	5.1800254	10.790845
mileage	-25.487369	5163.52388	21187.33402	5687.4179339	18189.736407
age	-27.860641	1.90309	3.68627	0.6429318	1.851781
	p.value				
price	9.782212e-14				
tax	1.652764e-02				
mpg	3.511833e-07				
mileage	2.721500e-143				
age	8.005358e-171				
	\$'3'				
	v.test	Mean in category	Overall mean	sd in category	Overall sd
mpg	9.594880	56.486827	53.31998	9.5781537	10.790845
price	6.037760	22709.333333	21027.36730	6252.6630256	9107.713091
mileage	-5.119504	18339.029782	21187.33402	9366.3957996	18189.736407
tax	-5.137804	144.537237	146.39236	4.9688693	11.804945
age	-5.442517	3.378007	3.68627	0.9604181	1.851781
	p.value				
mpg	8.401518e-22				
price	1.562686e-09				
mileage	3.063398e-07				
tax	2.779679e-07				
age	5.253288e-08				
	\$'4'				
	v.test	Mean in category	Overall mean	sd in category	Overall sd
age	13.487897	4.47697	3.68627	1.314516	1.851781
mileage	7.948345	25764.33615	21187.33402	12987.250326	18189.736407
mpg	6.789609	55.63939	53.31998	6.324558	10.790845
tax	-3.456239	145.10071	146.39236	7.560053	11.804945
price	-25.376843	13710.50424	21027.36730	4707.440048	9107.713091
	p.value				
age	1.842827e-41				
mileage	1.890192e-15				
mpg	1.124379e-11				
tax	5.477702e-04				
price	4.544033e-142				

```
$'5'
      v.test Mean in category Overall mean sd in category    Overall sd
tax     49.497910      174.864706     146.39236     19.906090     11.804945
age     23.593188       5.815138      3.68627      1.414365      1.851781
mileage 20.407788     39275.507860    21187.33402    16696.243937   18189.736407
price    -2.735532     19813.354005    21027.36730    7077.764822    9107.713091
mpg     -15.432028      45.205685      53.31998      6.070262      10.790845
```

p.value

```
tax     0.000000e+00
age     4.527472e-123
mileage 1.425924e-92
price    6.227962e-03
mpg     9.968823e-54
```

\$'6'

```
      v.test Mean in category Overall mean sd in category Overall sd
price   3.582564      25402.0182    21027.3673     8714.781321   9107.71309
tax     -3.334608      141.1146     146.3924      6.771841     11.80494
      p.value
price  0.0003402381
tax    0.0008541970
```

\$'7'

```
      v.test Mean in category Overall mean sd in category Overall sd
mileage 36.96279      40730.470194   21187.33402    15943.458413   18189.736407
mpg     36.30891      64.708616      53.31998      6.469240      10.790845
age     35.57747      5.601266      3.68627      1.209700      1.851781
tax     -20.57083      139.333760     146.39236     8.273959      11.804945
price   -25.88837     14173.793249    21027.36730    3794.958263   9107.713091
      p.value
mileage 4.538160e-299
mpg     1.170673e-288
age     3.125949e-277
tax     5.010468e-94
price   9.004372e-148
```

Como se ha comentado con anterioridad según la descripción a partir de los factores, podemos ver que en el primer cluster price es más alto, mientras que mileage, age o mpg son signitivamente más bajos que en el resto del df.

Asimismo, podemos ver como en el cluster 7, se acumulan vehiculos más viejos, con mayor consumo y más kilometraje, pero también más baratos.

## 5.2 Análisis según las componentes del MCA

Vamos a proceder a analizar la clusterización a apartir de los ejes generados a partir del MCA.

En primer lugar, podemos ver la relevancia que han tenido las distintas componentes del MCA para generar la clusterización:

```
res.hcmc$desc.axes$quanti.var
```

	Eta2	P-value
Dim.1	0.8404262	0.000000e+00
Dim.2	0.4949556	0.000000e+00
Dim.3	0.6440101	0.000000e+00
Dim.4	0.5653203	0.000000e+00
Dim.5	0.4988811	0.000000e+00
Dim.7	0.2819214	0.000000e+00
Dim.6	0.2351583	3.639829e-272

Podemos ver como la dimensión 1 ha tenido la mayor relevancia, mientras que la dimensión 6 ha sido la menos determinante.

Por último, podemos ver algunas estadísticas sobre como se distribuyen las coordenadas del MCA para cada individuo en los diferentes clusters:

```
res.hcmc$desc.axes$quanti
```

\$'1'

	v.test	Mean in category	Overall mean	sd in category	Overall sd
Dim.6	24.554509	0.27840462	1.191801e-16	0.3065580	0.3928936
Dim.3	9.897221	0.13508631	1.476254e-16	0.2331792	0.4729637
Dim.2	9.679917	0.13917280	1.605875e-16	0.3752721	0.4982100
Dim.5	9.584258	0.11308307	3.306217e-16	0.1933379	0.4088545
Dim.4	-7.568343	-0.09390327	-4.898780e-17	0.2581729	0.4299417
Dim.7	-17.852636	-0.20013574	6.413848e-17	0.2540431	0.3884651
Dim.1	-42.777253	-0.77957383	-9.014292e-17	0.1885507	0.6315011
	p.value				
Dim.6	3.871029e-133				
Dim.3	4.280045e-23				
Dim.2	3.670141e-22				
Dim.5	9.312458e-22				
Dim.4	3.780139e-14				
Dim.7	2.757099e-71				
Dim.1	0.000000e+00				

\$'2'

	v.test	Mean in category	Overall mean	sd in category	Overall sd
Dim.7	22.808153	0.30623620	6.413848e-17	0.2659344	0.3884651
Dim.3	2.885667	0.04717244	1.476254e-16	0.1804584	0.4729637
Dim.2	-5.500431	-0.09471605	1.605875e-16	0.4587562	0.4982100
Dim.6	-15.437222	-0.20963242	1.191801e-16	0.2961252	0.3928936
Dim.5	-16.738145	-0.23653234	3.306217e-16	0.2458388	0.4088545
Dim.4	-24.170549	-0.35917864	-4.898780e-17	0.2509115	0.4299417
Dim.1	-25.859909	-0.56443698	-9.014292e-17	0.1872810	0.6315011
	p.value				
Dim.7	3.805856e-115				
Dim.3	3.905857e-03				
Dim.2	3.788635e-08				
Dim.6	9.197741e-54				
Dim.5	6.910850e-63				
Dim.4	4.540929e-129				
Dim.1	1.882491e-147				

\$'3'

	v.test	Mean in category	Overall mean	sd in category	Overall sd
Dim.4	24.555338	0.32291452	-4.898780e-17	0.2792232	0.4299417
Dim.2	16.804880	0.25608251	1.605875e-16	0.3201478	0.4982100
Dim.1	-2.040786	-0.03941882	-9.014292e-17	0.3427683	0.6315011
Dim.7	-2.832450	-0.03365475	6.413848e-17	0.3637652	0.3884651
Dim.6	-15.368773	-0.18469119	1.191801e-16	0.3436848	0.3928936
Dim.3	-35.276306	-0.51032043	1.476254e-16	0.2420552	0.4729637
	p.value				
Dim.4	3.792892e-133				
Dim.2	2.247682e-63				
Dim.1	4.127207e-02				
Dim.7	4.619277e-03				
Dim.6	2.651420e-53				
Dim.3	1.355913e-272				

\$'4'

	v.test	Mean in category	Overall mean	sd in category	Overall sd
Dim.1	14.244974	0.28478279	-9.014292e-17	0.3044410	0.6315011
Dim.5	8.073963	0.10450415	3.306217e-16	0.3755884	0.4088545
Dim.4	6.824823	0.09289214	-4.898780e-17	0.3280819	0.4299417
Dim.6	-2.484464	-0.03090193	1.191801e-16	0.3501083	0.3928936

```

Dim.3 -10.473071      -0.15681220  1.476254e-16      0.3667009  0.4729637
Dim.2 -45.881955      -0.72365632  1.605875e-16      0.2487687  0.4982100
    p.value
Dim.1 4.817256e-46
Dim.5 6.805245e-16
Dim.4 8.803385e-12
Dim.6 1.297467e-02
Dim.3 1.148553e-25
Dim.2 0.000000e+00

$'5'
    v.test Mean in category Overall mean sd in category Overall sd
Dim.3 45.099629      1.03937712  1.476254e-16      0.3059038  0.4729637
Dim.4 35.080265      0.73492807  -4.898780e-17     0.3266240  0.4299417
Dim.1 11.507853      0.35411209  -9.014292e-17     0.2229436  0.6315011
Dim.2 7.713449       0.18725501  1.605875e-16      0.3201178  0.4982100
Dim.7 3.748776       0.07096004  6.413848e-17      0.4020680  0.3884651
Dim.5 -5.393149      -0.10744436  3.306217e-16      0.2318294  0.4088545
Dim.6 -10.379645     -0.19871460  1.191801e-16      0.4069518  0.3928936
    p.value
Dim.3 0.000000e+00
Dim.4 1.348051e-269
Dim.1 1.204399e-30
Dim.2 1.224625e-14
Dim.7 1.776999e-04
Dim.5 6.923335e-08
Dim.6 3.069148e-25

$'6'
    v.test Mean in category Overall mean sd in category Overall sd
Dim.5 44.051001      2.4147084   3.306217e-16      0.3420429  0.4088545
Dim.7 24.129447      1.2567228   6.413848e-17      0.4388973  0.3884651
Dim.2 6.145001       0.4104634   1.605875e-16      0.2900397  0.4982100
Dim.1 4.349714       0.3682772   -9.014292e-17     0.4986605  0.6315011
Dim.6 3.376278       0.1778497   1.191801e-16      0.3435683  0.3928936
Dim.4 -8.174260      -0.4711922  -4.898780e-17     0.2915627  0.4299417
    p.value
Dim.5 0.000000e+00
Dim.7 1.227355e-128
Dim.2 7.996319e-10
Dim.1 1.363154e-05
Dim.6 7.347353e-04
Dim.4 2.976880e-16

$'7'
    v.test Mean in category Overall mean sd in category Overall sd
Dim.1 45.493939      0.83508506  -9.014292e-17     0.1832566  0.6315011
Dim.2 15.470298      0.22403414  1.605875e-16      0.3658165  0.4982100
Dim.6 12.572771      0.14358498  1.191801e-16      0.3766458  0.3928936
Dim.7 -7.804744      -0.08812797  6.413848e-17      0.3536360  0.3884651
Dim.5 -8.718122      -0.10360837  3.306217e-16      0.3222143  0.4088545
Dim.4 -22.895727     -0.28613245  -4.898780e-17     0.2733127  0.4299417
    p.value
Dim.1 0.000000e+00
Dim.2 5.505198e-54
Dim.6 2.980890e-36
Dim.7 5.962236e-15
Dim.5 2.828532e-18
Dim.4 5.124762e-116

```

Podemos destacar que el componente más relevante del cluster 1 es el sentido negativo de la dimensión 1, mientras que para el cluster 5 adquiere más relevancia la dimensión 3.

### 5.3 Análisis según individuos

En las siguientes salidas podemos ver los parámetros del primer cluster.

```
res.hcmc$desc.ind$para
```

```
Cluster: 1
 40410    44910    45035    45038    45440
0.3120722 0.3120722 0.3120722 0.3120722 0.3120722
```

```
Cluster: 2
 34644    36032    36853    37857    38157
0.2135718 0.2135718 0.2135718 0.2135718 0.2135718
```

```
Cluster: 3
 3909     662     31881    33887    32842
0.2976628 0.3669386 0.4529215 0.4529215 0.4681449
```

```
Cluster: 4
 36534    37522    41873    42151    42468
0.3275151 0.3275151 0.3275151 0.3275151 0.3275151
```

```
Cluster: 5
 30937    3077     48361    48391    48404
0.3519279 0.4454410 0.4525254 0.4525254 0.4525254
```

```
Cluster: 6
 16294    16900    16987    16491    17777
0.4990666 0.4990666 0.4990666 0.5964068 0.5972435
```

```
Cluster: 7
 40062    22961    39454    41073    1434
0.3153163 0.4129064 0.4577924 0.4577924 0.4838603
```

Curiosamente, entre los parámetros del primer cluster, aparecen vehículos VW semi-automáticos Diesel. Todos tienen entre 1 y 2 años y consumos y kilometrajes muy bajos.

```
summary(df[c("40410", "44910", "45035", "45038", "45440")])
```

	model	year	price	transmission	
VW- Tiguan:4	2019 :5	Min. :25500	f.Trans-Manual :0		
VW- Passat:1	2001 :0	1st Qu.:27998	f.Trans-SemiAuto :5		
Audi- A1 :0	2002 :0	Median :28199	f.Trans-Automatic:0		
Audi- A3 :0	2003 :0	Mean :28339			
Audi- A4 :0	2004 :0	3rd Qu.:29999			
Audi- A5 :0	2005 :0	Max. :29999			
(Other) :0	(Other):0				
	mileage	fuelType	tax	mpg	engineSize
Min. : 1	f.Fuel-Diesel:5	Min. :145	Min. :39.8	2 :5	
1st Qu.: 669	f.Fuel-Petrol:0	1st Qu.:145	1st Qu.:40.4	1 :0	
Median :3000	f.Fuel-Hybrid:0	Median :145	Median :44.1	1.2 :0	
Mean :2737		Mean :145	Mean :43.1	1.3 :0	
3rd Qu.:4289		3rd Qu.:145	3rd Qu.:45.6	1.4 :0	
Max. :5726		Max. :145	Max. :45.6	1.5 :0	
				(Other):0	
	manufacturer	age	outs	f.miles	
Audi :0	Min. :2	Min. :0	f.miles-[0.001,5.81]:5		
BMW :0	1st Qu.:2	1st Qu.:0	f.miles-(5.81,17.7] :0		
Mercedes:0	Median :2	Median :0	f.miles-(17.7,34.1] :0		
VW :5	Mean :2	Mean :0	f.miles-(34.1,119] :0		
	3rd Qu.:2	3rd Qu.:0			
	Max. :2	Max. :0			

```

          f.tax           f.mpg           f.age           Audi
f.tax-[0,144] :0   f.mpg-muy bajo:5   f.age-[1,2]:5   Audi No :5
f.tax-(144,145]:5 f.mpg-bajo       :0   f.age-(2,4):0   Audi Yes:0
f.tax-(145,155]:0 f.mpg-medio     :0   f.age-(+4)  :0
f.tax-(155,205]:0 f.mpg-alto      :0

mout           aux           f.price        claKMPCA
NoMOut :5    (2.6e+04,3.15e+04]:4   f.price-(2.6e+04,3.15e+04]:4   Min.   :2
YesMOut:0    (2.2e+04,2.6e+04]  :1   f.price-(2.2e+04,2.6e+04]  :1   1st Qu.:2
              [899,1.1e+04]       :0   f.price-[899,1.1e+04]       :0   Median :2
              (1.1e+04,1.4e+04]  :0   f.price-(1.1e+04,1.4e+04]  :0   Mean    :2
              (1.4e+04,1.7e+04]  :0   f.price-(1.4e+04,1.7e+04]  :0   3rd Qu.:2
              (1.7e+04,1.95e+04] :0   f.price-(1.7e+04,1.95e+04]:0   Max.    :2
              (Other)           :0   (Other)           :0

claHCMC
f.claHCMC-1:5
f.claHCMC-2:0
f.claHCMC-3:0
f.claHCMC-4:0
f.claHCMC-5:0
f.claHCMC-6:0
f.claHCMC-7:0

```

En la siguiente salida podemos ver los individuos característicos de cada cluster:

```
res.hcmc$desc.ind$dist
```

```
Cluster: 1
  5388      7854     10322     10323     10482
1.494958 1.463050 1.463050 1.463050 1.463050
```

```
Cluster: 2
  15670     11950     12089     12871     13337
1.424690 1.335168 1.335168 1.335168 1.335168
```

```
Cluster: 3
  21818     21999     22408     22474     22485
1.593239 1.593239 1.593239 1.593239 1.593239
```

```
Cluster: 4
  42464     43555     35209     37488     37509
1.598215 1.598215 1.594687 1.594687 1.594687
```

```
Cluster: 5
  7910      9145      9872     10511     10617
2.020672 2.020672 2.020672 2.020672 2.020672
```

```
Cluster: 6
  18820     19986     20160     11341     14452
3.349056 3.349056 3.349056 3.347672 3.347672
```

```
Cluster: 7
  16650     17562     18460     18841     19174
1.717968 1.717968 1.717968 1.717968 1.717968
```

Si nos fijamos, en este caso, los infivi

```
summary(df[c("5388", "7854", "10322", "10323", "10482"),])
```

model	year	price	transmission	
Audi- A1:1	2019 :4	Min. :21500	f.Trans-Manual :0	
Audi- A3:1	2020 :1	1st Qu.:23500	f.Trans-SemiAuto :0	
Audi- A5:1	2001 :0	Median :30000	f.Trans-Automatic:5	
Audi- Q3:1	2002 :0	Mean :28804		
Audi- TT:1	2003 :0	3rd Qu.:32000		
Audi- A4:0	2004 :0	Max. :37020		
(Other) :0	(Other):0			
mileage	fuelType	tax	mpg	engineSize
Min. : 2500	f.Fuel-Diesel:0	Min. :145	Min. :31.70	2 :3
1st Qu.: 2796	f.Fuel-Petrol:5	1st Qu.:150	1st Qu.:38.70	1.5 :2
Median : 2893	f.Fuel-Hybrid:0	Median :150	Median :40.90	1 :0
Mean : 6299		Mean :149	Mean :39.66	1.2 :0
3rd Qu.: 3215		3rd Qu.:150	3rd Qu.:42.20	1.3 :0
Max. :20091		Max. :150	Max. :44.80	1.4 :0
				(Other):0
manufacturer	age	outs	f.miles	
Audi :5	Min. :1.0	Min. :0	f.miles-[0.001,5.81]:4	
BMW :0	1st Qu.:2.0	1st Qu.:0	f.miles-(5.81,17.7] :0	
Mercedes:0	Median :2.0	Median :0	f.miles-(17.7,34.1] :1	
VW :0	Mean :1.8	Mean :0	f.miles-(34.1,119] :0	
	3rd Qu.:2.0	3rd Qu.:0		
	Max. :2.0	Max. :0		
f.tax	f.mpg	f.age	Audi	
f.tax-[0,144] :0	f.mpg-muy bajo:5	f.age-[1,2]:5	Audi No :0	
f.tax-(144,145]:1	f.mpg-bajo :0	f.age-(2,4]:0	Audi Yes:5	
f.tax-(145,155]:4	f.mpg-medio :0	f.age-(+4) :0		
f.tax-(155,205]:0	f.mpg-alto :0			

mout	aux	f.price
NoMOut :5	(3.15e+04,1.09e+05]:2	f.price-(3.15e+04,1.09e+05]:2
YesMOut:0	(1.95e+04,2.2e+04] :1	f.price-(1.95e+04,2.2e+04] :1
	(2.2e+04,2.6e+04] :1	f.price-(2.2e+04,2.6e+04] :1
	(2.6e+04,3.15e+04] :1	f.price-(2.6e+04,3.15e+04] :1
	[899,1.1e+04] :0	f.price-[899,1.1e+04] :0
	(1.1e+04,1.4e+04] :0	f.price-(1.1e+04,1.4e+04] :0
	(Other) :0	(Other) :0
claKMPCA	claHCMC	
Min. :2	f.claHCMC-1:5	
1st Qu.:2	f.claHCMC-2:0	
Median :2	f.claHCMC-3:0	
Mean :2	f.claHCMC-4:0	
3rd Qu.:2	f.claHCMC-5:0	
Max. :2	f.claHCMC-6:0	
	f.claHCMC-7:0	

Si recogemos los datos de los parámetros del primer cluster, podemos ver como todos los vehículos son Audi, de cambio automático y gasolina. También tienen consumos muy bajos y tiene entre 1 y 2 años.

## 6 K-Means Clustering desde MCA

Por último, vamos a realizar un clustering con el algoritmo de K-Means a partir del MCA.

```
ppcc <- res.mca$ind$coord[,1:7];
kc<-kmeans(dist(ppcc),7)
```

En primer lugar, vamos a analizar el porcentaje de distancias que se acumulan con la suma de distancias entre clusters respecto al total.

```
kc$betweenss/kc$totss
```

```
[1] 0.611554
```

Podemos ver que es de un 61%, lo que a priori es peor que en la clusterización a partir de ACP.

En el caso de las distancias dentro de los clusters, estas suman un 39%.

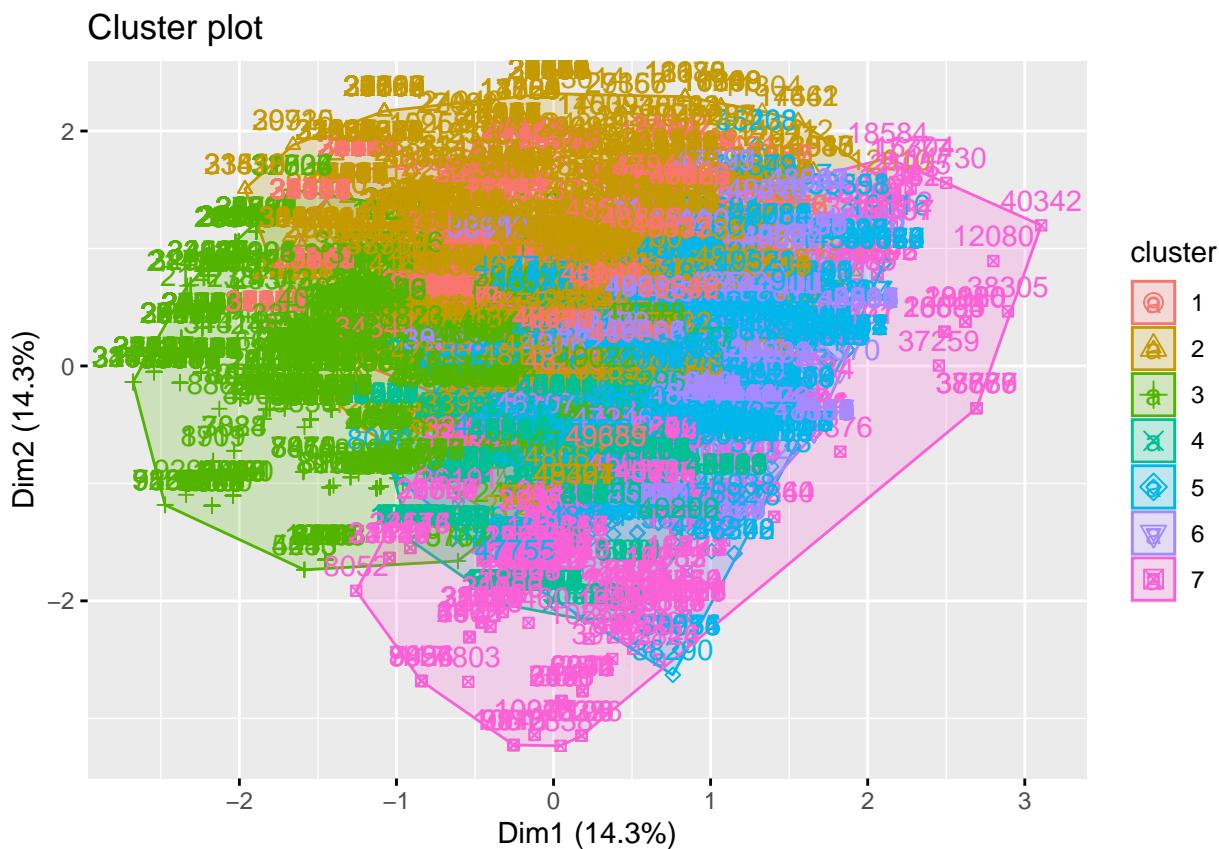
```
kc$tot.withinss/kc$totss
```

```
[1] 0.388446
```

Con estos dos resultados, se puede concluir que la clusterización desde el ACP es de mayor calidad que la generada a partir del ACM.

Este hecho es aún más evidente si lo representamos de manera gráfica:

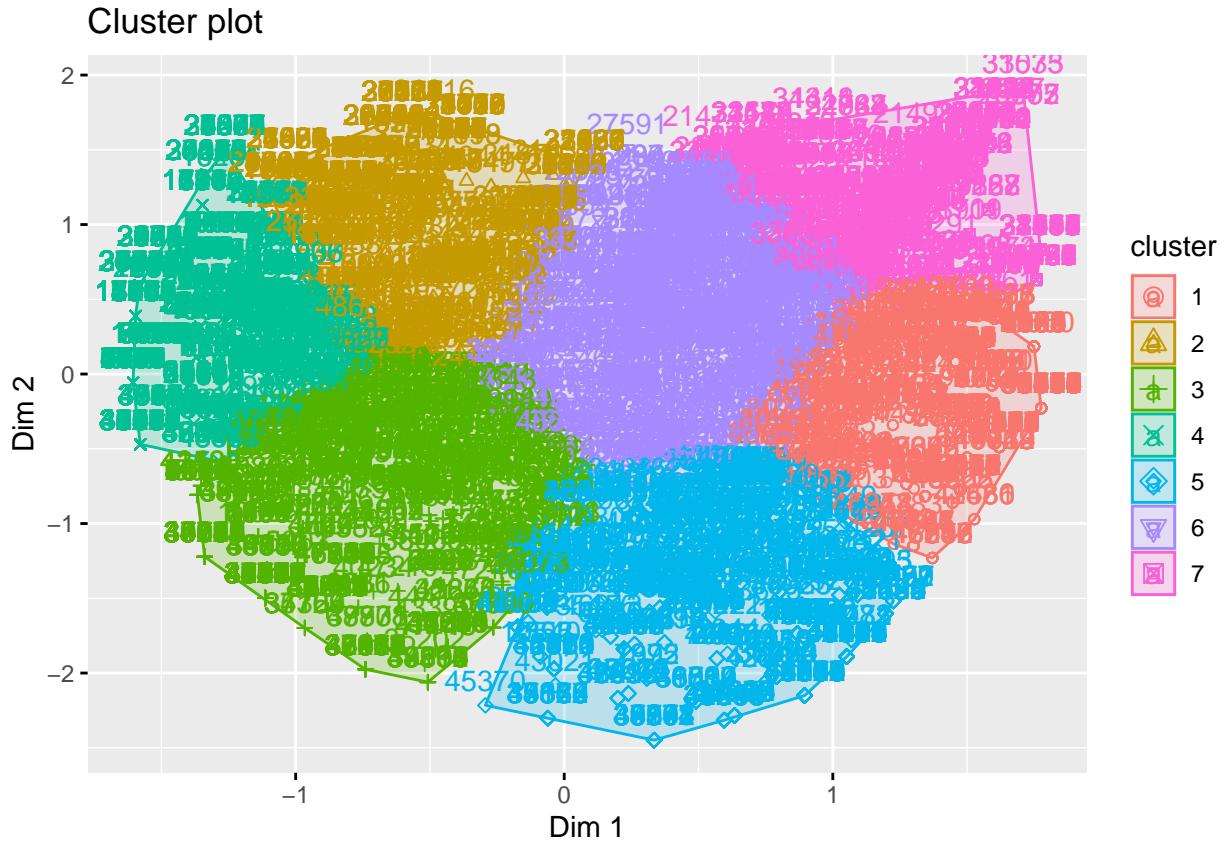
```
fviz_cluster(kc, data=ppcc)
```



Lamentablemente, al solo poder representar los clusters en 2 dimensiones, el gráfico resultante es prácticamente incomprendible, de modo que vamos a realizar el mismo proceso pero solo aportando las coordenadas pertenecientes a las 2 primeras componentes del MCA, que son las que más variabilidad acumulan (25%).

<Solo por motivos de visualización. Si pudieramos representar las 7 dimensiones veríamos como las clusterización SÍ que tiene sentido>

```
ppcc <- res.mca$ind$coord[,1:2];
kc<-kmeans(dist(ppcc),7)
fviz_cluster(kc, data=ppcc)
```



En

este caso si que podemos ver bien diferenciados los distintos clusters.

Si analizamos las distancias, podemos ver como en este caso, se crean clusters más diferenciados entre si, pero con individuos más similares. Esto es debido a que solo se están usando 2 dimensiones y la variabilidad que se acumula es muy baja.

`kc$betweenss/kc$totss;kc$tot.withinss/kc$totss`

[1] 0.8372228

[1] 0.1627772

Podemos ver que la distancia entre clusters acumula un 82% del total, mientras que las distancias dentro de los clusters acumulan solo un 17%. Sería un resultado bastante bueno si con las dos dimensiones que estamos representando acumuláramos mayor variabilidad.

Por último, volvemos a poner el gráfico generado a partir del clustering jerárquico con MCA para comparar las distintas clusterizaciones que se han llevado a cabo según clustering jerárquico y Kmeans.

```
plot.HCPC(res.hcmc, choice="map")
```

### Factor map

